МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ УКРАИНЫ

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ГОСУДАРСТВЕННОЕ ВЫСШЕЕ УЧЕБНОЕ ЗАВЕДЕНИЕ
«ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

ТАГАНРОГСКИЙ ГОСУДАРСТВЕННЫЙ РАДИОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

ШТУТТГАРТСКИЙ УНИВЕРСИТЕТ

ИНЖЕНЕРНАЯ АКАДЕМИЯ УКРАИНЫ

ООО „Технопарк ДонНТУ „УНИТЕХ”

МОДЕЛИРОВАНИЕ И КОМПЬЮТЕРНАЯ ГРАФИКА

Материалы
второй международной научно-технической конференции

10─12 октября 2007 года

г. Донецк –2007
ДонНТУ

ББК 33.31
УДК 004.925.8: 681.5: 658.5
М 74
М 74 **Моделирование и компьютерная графика :** Материалы 2-й международной научно -технической конференции , г Донецк , 10-12 октября 2007 г . — Донецк , ДонНТУ , Министерство образования и науки Украины , 2007. — 358 с .

Представлены доклады ученых и специалистов в области компьютерной графики, визуализации и моделирования сложных динамических систем. Приведены результаты исследований, связанных с различными аспектами разработки аппаратного, программного и алгоритмического обеспечения средств компьютерной графики, параллельного моделирования сложных динамических систем, моделирования и проектирования средств вычислительной техники. Рассматриваются вопросы практического применения компьютерной графики и моделирования. Материалы предназначены для ученых, преподавателей , аспирантов и студентов высших учебных заведений , научная деятельность которых связана с вопросами разработки и использования современных компьютерно – информационных систем.

Представлені доповіді вчених та спеціалістів в сфері комп’ютерної графіки, візуалізації, моделювання складних динамічних систем. Наведені результати досліджень, пов’язаних з різними аспектами розробки апаратного, програмного, алгоритмічного забезпечення засобів комп’ютерної графіки, паралельного моделювання складних динамічних систем, моделювання і проектування засобів обчислювальної техніки. Розглядаються питання практичного використання комп’ютерної графіки і моделювання. Матеріали призначені для вчених, викладачів, аспірантів та студентів вищих навчальних закладів, наукова діяльність яких пов’язана з питаннями розробкт і використання сучасних комп’ютерно–інформаційних систем.

Редакционная коллегия :

д -р техн . наук Минаев А .А .,
д -р техн . наук Башков Е .А .,
д -р техн . наук Зори А .А .,
д -р техн . наук Ли В .Г .,
д -р техн . наук Пьявченко О .Н .,
д -р техн . наук Святный В .А .,
д -р техн . наук Скидан И .А .,
д -р техн . наук Фельдман Л .П .

Рекомендовано к печати Ученым советом Донецкого национального технического университета Министерства образования и науки Украины
Протокол № **6** от **07 сентября 2007 р .**

© Донецкий национальный технический университет Министерства образования и науки Украины , 2007

СОДЕРЖАНИЕ

П Л Е Н А Р Н О Е З А С Е Д А Н И Е

ПРИВЕТСТВЕННОЕ СЛОВО 7

П.01 **Н.Н Куссуль., А.Ю. Шелестов, С.В. Скакун** 8
НЕЙРОСЕТЕВАЯ МОДЕЛЬ ОПРЕДЕЛЕНИЯ ЗАТОПЛЕНИЙ С ИСПОЛЬЗОВАНИЕМ РАДИОЛОКАЦИОННЫХ ДАННЫХ

П.02 **Ли В.Г.** 13
ИНФОРМАЦИОННАЯ И ФУНКЦИОНАЛЬНАЯ МОЩНОСТЬ СЦЕН СВР

П.03 **Васюхин М.И., Капштык О.И., Пономарев С.А.** 19
МЕТОДЫ РЕШЕНИЯ ЗАДАЧ АНАЛИЗА ВОЗДУШНОЙ ОБСТАНОВКИ ПРИ ЕЁ ПРЕДСТАВЛЕНИИ В РЕАЛЬНОМ ВРЕМЕНИ

П.04 **Фельдман Л.П.** 24
ПАРАЛЛЕЛЬНЫЕ БЛОЧНЫЕ МЕТОДЫ РЕШЕНИЯ ЗАДАЧИ КОШИ ДЛЯ ОБЫКНОВЕННЫХ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ

П.05 **Аноприенко А.Я.** АРХЕОМОДЕЛИРОВАНИЕ: ДОЦИФРОВАЯ ЭПОХА В ВЫЧИСЛИТЕЛЬНОМ МОДЕЛИРОВАНИИ И ЕЕ ЗНАЧЕНИЕ В КОНТЕКСТЕ ОБОБЩЕННОГО КОДО – ЛОГИЧЕСКОГО БАЗИСА 29

Секция 1. «Компьютерная графика и визуализация» 8 корпус, аудитория 704

С1. 1 **Пьявченко А.О.** 35
ИНТЕГРАЛЬНЫЙ МЕТОД ВОССТАНОВЛЕНИЯ РЕЛЬЕФА ДОННОЙ ПОВЕРХНОСТИ ПО КАРТОГРАФИЧЕСКОЙ ИНФОРМАЦИИ

С1. 2 **Mykola Yas'ko** 41
G95 FORTRAN BINDINGS FOR OPENGL

С1. 3 **Аббасов И. Б.** 43
МОДЕЛИРОВАНИЕ САМОЛЕТА-АМФИБИИ БЕ-200

С1. 4 **Антощук С. Г., Щербакова Г.Ю. , Н.Ю. Хитрук, О.Л. Дрегваль** 48
АЛГОРИТМ КОНТЕКСТНОГО ПОИСКА ОБЪЕКТОВ ПО ФОРМЕ

С1. 5 **Белоус Н.В., Кобзарь Г.А.** 52
ПОСТРОЕНИЕ ЛОКАЛЬНЫХ ИЗОПОВЕРХНОСТЕЙ СКАЛЯРНЫХ ПОЛЕЙ С ПОМОЩЬЮ АЛГОРИТМА НЕНАПРАВЛЕННОЙ МАРКИРОВКИ

С1. 6 **Васюхін М.І., Васильєв І.В., Лобанчикова Н.М.** 56
ІНТЕРАКТИВНА АВТОМАТИЗОВАНА ГЕОІНФОРМАЦІЙНА СИСТЕМА ВИСОКОГО РІВНЯ БЕЗПЕКИ ОСОБЛИВО ВАЖЛИВИХ ОБ’ЄКТІВ

С1. 7 **Вовк О.Л.** 61
ПРИМЕНЕНИЕ МУЛЬТИМНОЖЕСТВ ДЛЯ ВЫДЕЛЕНИЯ РЕГИОНОВ ИЗОБРАЖЕНИЙ

С1. 8 **Карабчевский В. В. , Грыбинныk А.В.** 66
ТРЕХМЕРНАЯ АНИМАЦИЯ И ДИЗАЙН В РЕКЛАМЕ

С1.9 **Шелестов А.Ю.** 71
МОДЕЛЬ УЗЛА GRID-СИСТЕМЫ НА ОСНОВЕ СЕТЕЙ ПЕТРИ.

С1.10 **Ильин Н.И., Кравченко А.Н**. 78
МОДЕЛИРОВАНИЕ И СОЗДАНИЕ РАСПРЕДЕЛЕННЫХ СИСТЕМ ВИЗУАЛИЗАЦИИ ГЕОПРОСТРАНСТВЕННЫХ РЕЗУЛЬТАТОВ ОБРАБОТКИ СПУТНИКОВЫХ ДАННЫХ

С1. 11 **Кузнецов С.В., Холод К. О.** 83
ПРИКЛАД ВИБОРУ АРХІТЕКТУРНИХ РІШЕНЬ ПІДСИСТЕМИ РОБОТИ З ВІЗУАЛЬНИМИ МОДЕЛЯМИ

С1. 12 **Башков Е. А.,Пауков Д. П.** 88
ПОСТРОЕНИЕ ПОВЕРХНОСТИ ОБЪЕКТОВ СЛОЖНОЙ ФОРМЫ ПО НАБОРУ ТОЧЕК ДЛЯ ВИЗУАЛИЗАЦИИ И МОДЕЛИРОВАНИЯ

С1. 13 **Романюк А. Н.** 95
МЕТОДЫ И СРЕДСТВА ЗАКРАСКИ ТРЕХМЕРНЫХ ОБЪЕКТОВ С ИСПОЛЬЗОВАНИЕМ ЛОКАЛЬНЫХ МОДЕЛЕЙ ОСВЕЩЕНИЯ

С1. 14	**Смолий В.В.** ИТЕРАЦИОННЫЕ МЕТОДЫ ПОВОРОТА РАСТРОВЫХ ИЗОБРАЖЕНИЙ	103
С1. 15	**Погорелов О.О.** АПРОКСИМАЦІЯ ЗВОРОТНОГО ПЕРЕТВОРЕННЯ ЛАПЛАСА АНАЛІТИЧНИМИ ВИРАЗАМИ У РАЗІ СКЛАДНИХ ЗОБРАЖЕНЬ	108

Секция 2. «Моделирование и проектирование вычислительной техники и сложных динамических систем»

С2.1	**Ледовской М.И.** МОДЕЛЬ ЦИФРО-АНАЛОГОВОГО ПРЕОБРАЗОВАНИЯ НА ОСНОВЕ ШИМ МИКРОКОНТРОЛЛЕРОВ AVR	113
С2. 2	**Максимов А.В.** ИСПОЛЬЗОВАНИЕ СТРУКТУРНО-ЛИНГВИСТИЧЕСКОГО МЕТОДА АНАЛИЗА ДЛЯ РАСПОЗНАВАНИЯ СТРУКТУРЫ ЭКГ-СИГНАЛОВ	118
С2. 3	**Пирский А.В.** НЕОБХОДИМОСТЬ В РАЗРАБОТКЕ МЕТОДИК ПРОВЕДЕНИЯ ИССЛЕДОВАНИЙ И ИСПЫТАНИЙ ДЛЯ ИНТЕЛЛЕКТУАЛЬНОГО ДАТЧИКА ИЗМЕРЕНИЯ АБСОЛЮТНОГО ДАВЛЕНИЯ	123
С2. 4	**Пьявченко О.Н.** ВЫСОКОПРОИЗВОДИТЕЛЬНЫЕ РАСПРЕДЕЛЕННЫЕ ИНТЕЛЛЕКТУАЛЬНЫЕ МИКРОКОМПЬЮТЕРНЫЕ СИСТЕМЫ МОНИТОРИНГА СЛОЖНЫХ ДИНАМИЧЕСКИХ ОБЪЕКТОВ	127
С2. 6	**Дмитриева О.А., Фельдман Л.П.** ОБЕСПЕЧЕНИЕ УСТОЙЧИВОСТИ ЧИСЛЕННОГО МОДЕЛИРОВАНИЯ ДИНАМИЧЕСКИХ СИСТЕМ	136
С2. 5	**Фельдман Л.П., Назарова И.А., Шматько А.Е.** ПАРАЛЛЕЛЬНЫЙ РЕКУРСИВНЫЙ АЛГОРИТМ БЫСТРОГО МАТРИЧНОГО УМНОЖЕНИЯ	142
С2. 7	**Ладыженский Ю.В., Попов Ю.В., Тесленко Г.А.** ИСПОЛЬЗОВАНИЕ ДИНАМИЧЕСКОЙ СИНХРОНИЗАЦИИ В СИСТЕМЕ РАСПРЕДЕЛЕННОГО ЛОГИЧЕСКОГО МОДЕЛИРОВАНИЯ	146
С2. 8	**Андрюхин А.И.** МОДЕЛИРОВАНИЕ И ТЕСТИРОВАНИЕ НЕЙРОМОРФНЫХ МИКРОЧИПОВ НА ПЕРЕКЛЮЧАТЕЛЬНОМ УРОВНЕ	153
С2. 9	**Белецкий В.Н., Резникова С.А., Чемерис А.А.** ПОСТРОЕНИЕ СТАТИСТИЧЕСКИХ МОДЕЛЕЙ ЭФФЕКТИВНОСТИ ПАРАЛЛЕЛЬНЫХ ПРОГРАММ	157
С2. 10	**Иванов Д.Е., Скобцов Ю.А.** МОДЕЛИРОВАНИЕ ЦИФРОВЫХ СХЕМ С НЕИСПРАВНОСТЯМИ НА РАСПРЕДЕЛЁННЫХ СИСТЕМАХ КЛАССА MIMD	162
С2. 11	**Грищенко В.И., Ладыженский Ю.В.,** МОДЕЛИРОВАНИЕ РАБОТЫ ПРИЛОЖЕНИЙ НА СЕТЕВЫХ ПРОЦЕССОРАХ	167
С2. 12	**Куркчи В. А., Ладыженский Ю.В.** МЕТОД ПОЛУЧЕНИЯ ОПТИМАЛЬНОГО БАЛАНСА НАГРУЗКИ ДЛЯ РАСПРЕДЕЛЕННОГО ЛОГИЧЕСКОГО МОДЕЛИРОВАНИЯ	174
С2. 13	**Ладыженский Ю.В., Мирецкая В.А.** МУРАВЬИНАЯ МАРШРУТИЗАЦИЯ В СЕТЯХ НА КРИСТАЛЛАХ	182
С2. 14	**Ладыженский Ю.В., Мирецкий А.В.** ОПТИМИСТИЧЕСКИЙ АЛГОРИТМ СИНХРОНИЗАЦИИ С ВНУТРИКОМПОНЕНТНЫМИ ОТКАТАМИ ДЛЯ РАСПРЕДЕЛЕННОГО МОДЕЛИРОВАНИЯ	187
С2. 15	**Ладыженский Ю.В., Родригес Залепинос Рамон Антонио** МУРАВЬИНЫЙ АЛГОРИТМ ДЛЯ РАЗБИЕНИЯ ГРАФОВ	193
С2. 16	**Ладыженский Ю.В., Старосоцкая О.А.** ПРОГРАММНАЯ СИСТЕМА ДЛЯ РЕШЕНИЯ ЗАДАЧ ОПТИМИЗАЦИИ НА ГРАФАХ С ИСПОЛЬЗОВАНИЕМ ГЕНЕТИЧЕСКИХ АЛГОРИТМОВ	198

С2. 17	**Назарова И.А.** АЛГОРИТМИЧЕСКИЕ МЕТОДЫ ПОВЫШЕНИЯ ЭФФЕКТИВНОСТИ ПАРАЛЛЕЛЬНИХ ВС ПРИ ЧИСЛЕННОМ РЕШЕНИИ СОДУ С КОНТРОЛЕМ ПОГРЕШНОСТИ НА ШАГЕ	202
С2. 18	**Саух С.Е.** МЕТОД CR-ФАКТОРИЗАЦИИ МАТРИЦ	205
С2. 19	**Фельдман Л.П., Назарова И.А., Кожухов А.Е.** ПАРАЛЛЕЛЬНЫЕ АЛГОРИТМЫ ЧИСЛЕННОГО РЕШЕНИЯ СИСТЕМ ЛИНЕЙНЫХ ОБЫКНОВЕННЫХ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ В СИСТЕМАХ С РАСПРЕДЕЛЁННОЙ ПАМЯТЬЮ С ТОПОЛОГИЕЙ РЕШЁТКА-ТОР	214
С2. 20	**Афонов И.В., Ладыженский Ю.В.** МОДЕЛЬ ПРОИЗВОДИТЕЛЬНОСТИ РЕПЛИЦИРОВАННОЙ БАЗЫ ДАННЫХ 2RC	219
С2. 21	**Михайлова Т.В.** ОЦЕНКА ЭФФЕКТИВНОСТИ ВЫСОКОПРОИЗВОДИТЕЛЬНЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ С ИСПОЛЬЗОВАНИЕМ АНАЛИТИЧЕСКИХ МЕТОДОВ	222

Секция 3. «Практическое применение средств моделирования и компьютерной графики»

С3. 1	**Куссуль Н.Н., Кравченко А.Н.** ПОСТРОЕНИЕ КАСКАДА МОДЕЛЕЙ ДЛЯ ПРОГНОЗИРОВАНИЯ ПАВОДКОВ	224
С3. 2	**Каргін А. О., Доценко І. О., П'ятикоп О. Є.** МОДЕЛИРОВАНИЕ ФУНКЦИОНИРОВАНИЯ РЕЛЕЙНЫХ КЛЕТОК ЗРИТЕЛЬНОЙ СИСТЕМЫ В ЗАДАЧАХ ИНТЕРПРЕТАЦИИ ИЗОБРАЖЕНИЙ	229
С3. 3	**Каира В. В.** ИССЛЕДОВАНИЕ И ВЫБОР МЕТОДА ВЫДЕЛЕНИЯ ПРИЗНАКОВ ИЗОБРАЖЕНИЯ КЕРАТИНОЦИТОВ ПРИ ОЦЕНКЕ ЖИЗНЕСПОСОБНОСТИ И КОЛИЧЕСТВЕННОГО РОСТА КЛЕТОК	234
С3. 4	**Клевцова А.Б.** ОСОБЕННОСТИ ИННОВАЦИОННОЙ ПОДГОТОВКИ СПЕЦИАЛИСТОВ В ОБЛАСТИ МИКРОКОМПЬЮТЕРНЫХ СИСТЕМ НА ПЕРВЫХ КУРСАХ ОБУЧЕНИЯ	240
С3. 5	**Мартыненко Т. В.** ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА ОБРАБОТКИ И РАСПОЗНАВАНИЯ ИЗОБРАЖЕНИЙ ГИСТОЛОГИЧЕСКИХ СРЕЗОВ НА ОСНОВЕ ЭВОЛЮЦИОННЫХ МОДЕЛЕЙ	246
С3. 6	**Слесарев В.В., Коваленко А.Н., Алексеев А.М.** ЛОГІКО – ЛІНГВІСТИЧНЕ МОДЕЛЮВАННЯ АВАРІЙНИХ ПРОЦЕСІВ В ВЕНТИЛЯЦІЙНИХ СИСТЕМАХ ШАХТ	253
С3. 7	**Скобцов Ю. А., Родин Ю.А., Оверко В.С.** ИССЛЕДОВАНИЕ ТЕЧЕНИЯ КРОВИ В РАЗВЕТВЛЕНИЯХ КРОВЕНОСНЫХ СОСУДОВ СО СТЕНОЗОМ	259
С3. 8	**Коломойцева И.А.** ОБЪЕКТНАЯ МОДЕЛЬ ЕСТЕСТВЕННО-ЯЗЫКОВОГО МЕДИЦИНСКОГО ТЕКСТА	265
С3. 9	**Тихонова О. А.** НОВЫЙ КЛАСС СИНТЕТИЧЕСКИХ ИНДИКАТОРОВ	271
С3.10	**Коломойцева И.А., Юдина Ю.А.** МОДЕЛЬ СЕМАНТИЧЕСКОГО АНАЛИЗА МЕДИЦИНСКОГО ЕСТЕСТВЕННО-ЯЗЫКОВОГО ТЕКСТА	278
С3. 11	**Балюба І.Г., Поліщук В.І., Горягін Б.Ф., Старченко Ж.В.** ПОЧАТКОВІ ЕТАПИ ВИКОРИСТАННЯ КОМП'ЮТЕРА ЯК КРЕСЛЯРСЬКОГО ІНСТРУМЕНТА	286
С3. 12	**Щербакова М. Е.** ИСПОЛЬЗОВАНИЕ МОДЕЛИРОВАНИЯ ПРИ БАЛАНСИРОВКЕ НАГРУЗКИ В КОМПЬЮТЕРНОЙ СЕТИ УПРАВЛЕНИЯ	291
С3. 13	**Чернов Е.И.** РАСПРЕДЕЛЕННАЯ СИСТЕМА АВТОМАТИЧЕСКОГО УПРАВЛЕНИЯ ТРАНСПОРТНЫМ СРЕДСТВОМ	295

С3. 14	**Григорьев А.В.** ОЦЕНКА ЭФФЕКТИВНОСТИ ИНСТРУМЕНТАЛЬНОЙ СИСТЕМЫ ДЛЯ СОЗДАНИЯ ИНТЕЛЛЕКТУАЛЬНЫХ САПР КАК СИСТЕМЫ С БАЗОЙ ЗНАНИЙ.	300
С3. 15	**Григорьев А.В.** МОДУЛЬ ЗНАНИЙ КАК СРЕДСТВО ОГРАНИЧЕНИЯ СТРУКТУРНЫХ ПАРАМЕТРОВ БАЗЫ ЗНАНИЙ ИНСТРУМЕНТАЛЬНОЙ ОБОЛОЧКИ ДЛЯ ПОСТРОЕНИЯ ИНТЕЛЛЕКТУАЛЬНЫХ САПР	311
С3. 16	**Ладыженский Ю.В., Середа А.А.** СПЕЦИАЛИЗИРОВАННЫЕ ПАРАЛЛЕЛЬНЫЕ ПРОЦЕССОРЫ ДЛЯ ОТСЛЕЖИВАНИЯ ПЕРЕМЕЩЕНИЙ ОБЪЕКТОВ В ВИДЕОПОТОКЕ	322
С3. 17	**Рычка С.В., А. Я. Аноприенко, Хасан Аль Абабнех** ПОВЫШЕНИЕ ПРОИЗВОДИТЕЛЬНОСТИ ИНТЕРНЕТ-ОРИЕНТИРОВАННОЙ СЕТЕВОЙ ИНФРАСТРУКТУРЫ: МОДЕЛИ, АЛГОРИТМЫ И МЕТОДЫ	326
С3.18	**Зорі А.А., Тарасюк В.П.** ВИКОРИСТАННЯ ІСНУЮЧИХ ЗАСОБІВ МОДЕЛЮВАННЯ ЕЛЕКТРОННИХ СХЕМ ПРИ ЗАПРОВАДЖЕННІ ДО УЧБОВОГО ПРОЦЕСУ ЕЛЕКТРОННИХ ПІДРУЧНИКІВ З НАПРЯМКУ “ЕЛЕКТРОНІКА”	331
С3.19	**Вовна О.В., Хламов М.Г.** МОДЕЛЮВАННЯ ВИМІРЮВАНЬ КОНЦЕНТРАЦІЇ МЕТАНУ ДЛЯ УМОВ ВУГІЛЬНИХ ШАХТ ДОНБАСУ	336
С3.20	**Клевцов С.И.** ПОГРЕШНОСТИ ПРОСТРАНСТВЕННЫХ МОДЕЛЕЙ ФУНКЦИЙ ПРЕОБРАЗОВАНИЯ ЧУВСТВИТЕЛЬНЫХ ЭЛЕМЕНТОВ ДАТЧИКОВ ДАВЛЕНИЯ	342
С3.21	**Пинчук О.Г.** МОДЕЛИРОВАНИЕ ТОКОВ И ТЕМПЕРАТУР АСИНХРОННЫХ ДВИГАТЕЛЕЙ ПРИ РАЗЛИЧНЫХ ХАРАКТЕРИСТИКАХ НЕ-СИММЕТРИИ СЕТИ: МЕТОДИЧЕСКИЕ ПОДХОДЫ	347
С3.22	**Чепцов А.А.** UML-ПОДХОД К РАЗРАБОТКЕ ДИСКРЕТНЫХ СИМУЛЯТОРОВ (НА ПРИМЕРЕ ОБЪЕКТОВ УГОЛЬНОЙ ПРОМЫШЛЕННОСТИ)	354

П Р И В Е Т С Т В Е Н Н О Е С Л О В О

Донецкий национальный технический университет приветствует гостей – участников второй Международной научно – технической конференции «Моделирование и компьютерная графика».

В настоящее время Донецкий национальный технический университет, основанный в 1921 году, представляет собой крупнейший научно -технический и учебный центр , где обучается более 27 тысяч студентов и работает свыше ста двадцати профессоров, докторов наук и 650 доцентов , кандидатов наук. Научные школы ДонНТУ известны своими многочисленными фундаментальными разработками, которые на протяжении многих десятилетий ведутся в различных областях. Широко известны в научных кругах Украины и за рубежом имена профессоров, возглавлявших эти научные школы .

Создание компьютерно – информационных систем различного назначения и их внедрение в различные сферы производства в настоящее время является необходимым условием инновационного развития Украины. В университете на различных кафедрах и в исследовательских лабораториях сформировались научные направления, связанные с разработкой, исследованием и внедрением аппаратного, алгоритмического и программного обеспечения систем, ориентированных на моделирования сложных динамических процессов, построением САПР средств вычислительной техники и компьютерной визуализацией процессов различной природы. Исследования в области компьютерной графики и обработки изображений также находятся в центре внимания многих ученых. Результаты, полученные учеными нашего университета в указанных областях, выносятся на обсуждение научной общественности в рамках работы Второй международной научно – технической конференции “Моделирование и компьютерная графика». Доклады ученых из других организаций, результаты их исследований и разработок будут приняты с вниманием и благодарностью.

В работе Первой международной конференции «Моделирование и компьютерная графика», проводившейся два года назад, были заслушаны около пятидесяти докладов ученых из ДонНТУ, а также из Национального авиационного университета, Института проблем моделирования в энергетике им. Г.Е.Пухова, Таганрогского радиотехнического университета, института кибернетики им.В.М.Глушкова, Харьковского национального университета электроники и других ведущих научных организаций Украины и России. Приятно осознавать, что спустя два года конференция проводится снова.

Хотелось бы выразить надежду, что настоящая конференция будет полезной для всех ее участников. Приветствую Вас в стенах Донецкого национального технического университета и желаю Вам плодотворной работы и интересных встреч.

Председатель оргкомитета конференции
Минаев А .А .

УДК 681.32

НЕЙРОСЕТЕВАЯ МОДЕЛЬ ОПРЕДЕЛЕНИЯ ЗАТОПЛЕНИЙ С ИСПОЛЬЗОВАНИЕМ РАДИОЛОКАЦИОННЫХ ДАННЫХ

Н.Н Куссуль., А.Ю. Шелестов, С.В. Скакун
Институт космических исследований НАНУ-НКАУ

В работе предложен новый подход для выявления затопленных территорий на основе радиолокационных данных. Для сегментации и классификации спутниковых изображений используются нейронные сети – самоорганизующиеся карты Кохонена. Предложенный подход верифицирован на данных спутника ERS-2 во время наводнения на р. Тиса в марте 2001 г.

На сегодняшний день наводнения относятся к наиболее масштабным природным стихийным бедствиям, которые приводят к значительному ущербу имущества и потерям человеческих жизней [1]. В последние годы страны Центральной и Восточной Европы достаточно сильно страдают от затоплений, вызванных ливневыми дождями. Эффективный мониторинг и прогнозирование наводнений, а также управление рисками, связанными с наводнениями, невозможен без привлечения данных наблюдения Земли из космоса. Одной из актуальных задач, которая возникает при мониторинге наводнений, является задача оценки площадей затопленных территорий. Эта информация является очень важной при оценке ущерба, причиненного наводнениями, может использоваться как входная информация в гидрологических моделях для моделирования и прогнозирования наводнений.

Обычно паводковые ситуации сопровождаются значительным облачным покровом, который ограничивает использование данных, полученных в оптическом диапазоне. Поэтому более эффективным является использование радиолокационных данных (в сантиметровом и дециметровом диапазонах), получение которых не зависит от погодных условий.

На сегодняшний день радиолокационные данные широко применяются для мониторинга наводнений [2]. При этом для обработки радиолокационных изображений, как правило, используют методы пороговой сегментации [3] или анализа разновременных снимков [4]. Несмотря на простоту и высокую скорость этих методов, они не лишены следующих недостатков: необходимость ручной настройки пороговых значений и параметров для сегментации изображений, отсутствие пространственной взаимосвязи между пикселями изображения, привлечение оператора к обработке

изображений, необходимость наличия опыта у оператора для анализе радиолокационных изображений.

В данной работе предлагается новый подход для определения затопленных территорий, который основан на автоматической нейросетевой сегментации изображения с использованием пространственных взаимосвязей между пикселями изображения. Сегментация и классификация изображений выполняется с использованием нейронных сетей Кохонена [5]. К преимуществам нейронных сетей данного типа следует отнести автоматическое выявление статистически значимых признаков в обучающей выборке данных и отсутствие необходимости в сложных моделях фильтрации для удаления спекла.

Описание спутниковых данных. Для апробации предлагаемого метода определения затопленных наводнениями территорий были использованы радиолокационные изображения, полученные прибором SAR со спутника ERS-2 во время наводнения на речке Тиса в марте 2001 г. и без наводнения в апреле того же года. Данные были предоставлены Европейским космическим агентством (ЕКА) в рамках гранта ЕКА Category-1 “Wide Area Grid Testbed for Flood Monitoring using Spaceborne SAR and Optical Data” (№4181).

Для более точной геопривязки радиолокационных данных ERS-2 и валидации полученных результатов были использованные дополнительные данные для исследуемой территории: данные спутника Landsat-7/ETM+, дата съемки: 30.09.2000 и данные Европейского проекта Corine Land Cover [6] по классификации земного покрова.

С учетом указанной дополнительной информации для обучения нейросетевой модели (настройки весовых коэффициентов) были выбраны тестовые пиксели, которые отвечают как территориям с водой (условно отнесем их к классу "Вода"), так и без воды (класс "Суша"). Всего для обучающей выборки было отобрано 42955 пикселей, из которых 12939 относятся к классу "Вода" и 30016 к классу «Суша».

Метод определения затопленных наводнениями территорий. Предложенный метод определения затопленных наводнениями территорий включает предварительную и тематическую обработку. Предварительная обработка радиолокационных изображений состоит из следующих этапов:

преобразование исходных данных в географическую проекцию, калибровка изображения, географическая привязка на основе данных спутника Landsat-7/ETM+.

Тематическая обработка сводилась к сегментации изображения с использованием нейронных сетей Кохонена и их классификации на указанные выше типы («Вода» и «Суша»). Самоорганизующиеся карты Кохонена – это специальный класс искусственных нейронных сетей, работа которых основана на конкурентном принципе обучения [5]. Нейронная сеть представляет собой двумерную решетку, в узлах которой расположены нейроны (элементарные обрабатывающие элементы). Весовые коэффициенты ω_j всех нейронов из окрестности нейрона-победителя i модифицируются следующим образом при переходе от момента времени n к $n+1$ [5]:

$$i(x) = \arg\min_{j=\overline{1,l}} \left\| x - \omega_j \right\|,$$

$$\omega_j(n+1) = \omega_j(n) + \eta(n) h_{j,i(x)}(n)(x - \omega_j(n)), \quad j = \overline{1, l},$$

где η – коэффициент скорости обучения, $h(n)$ – функция окрестности нейрона-победителя, x – вектор обучающего множества; $\|\cdot\|$ — Евклидова норма.

Нейронные сети Кохонена широко применяются для сегментации изображений в разных предметных областях, возможно с последующей классификацией [5]. Важным аспектом при обработке изображений является выбор информативных признаков, которые будут использованы как входные значения для нейронной сети. Обработку изображений обычно выполняют не попиксельно, а используют данные некоторой окрестности пикселя. В качестве информативных признаков можно выбирать значения интенсивности пикселей, значения производных, разнообразные фильтры, преобразования (например, преобразование Фурье) и т.д. [**7**]. В данной работе на вход нейронной сети в качестве информативных признаков подаются значения интенсивностей пикселей некоторой окрестности. В частности, была проведена серия экспериментов, в которой размер окрестности составлял: 3x3, 5x5, 7x7, 9x9, 11x11.

Для оценки качества обучения нейронной сети Кохонена использовались следующие величины:

– погрешность дискретизации или квантования (quantization error) обученной сети Кохонена. Данная величина определяет качество сегментации изображения и вычисляется согласно следующей формуле:

$$QE = \frac{1}{N}\sum_{t=1}^{N}\left\| x_t - \omega_{i(x_t)} \right\|, \quad i(x_t) = \arg\min_{j=\overline{1,l}} \left\| x_t - \omega_j \right\|,$$

где N — размер обучающей выборки.

– процент верно классифицированных образов, т.е. результаты классификации нейронной сети на тестовых данных.

Экспериментальные результаты. Для определения оптимальной архитектуры и параметров нейронной сети Кохонена была проведена серия экспериментов. При этом варьировались следующие параметры:

– размер окрестности каждого пикселя радиолокационного изображения, который определял количество нейронов во входном слое нейронной сети;

– количество нейронов выходного слоя, который отвечает количеству кластеров, на которые будут сегментированы изображения.

Другие параметры нейронной сети Кохонена, которая использовалась для сегментации радиолокационных изображений, имели следующие значения: структура соседства – шестиугольная; – функция окрестности – гауссова; начальное значение коэффициента обучения: 0,1; – количество эпох обучения: 20.

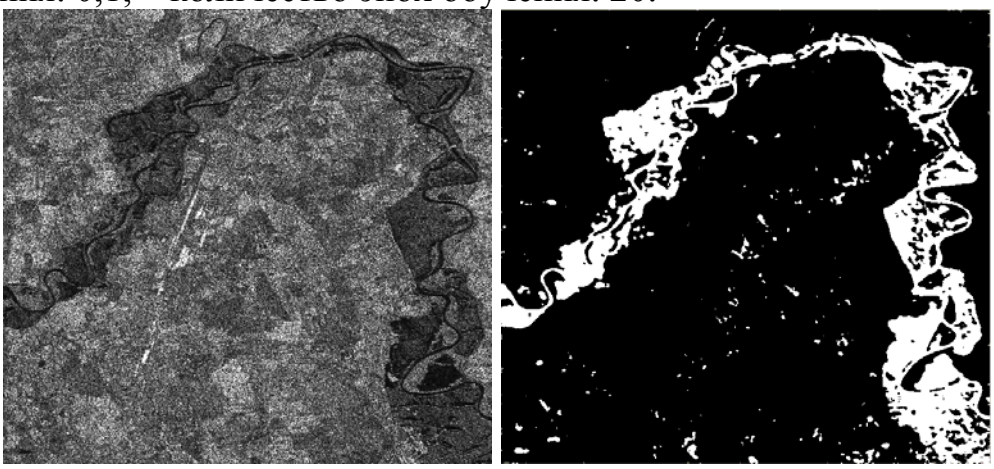


Рис. 1. Исходное радиолокационное изображение (слева) и результирующее классифицированное изображение (справа, белым цветом обозначены пиксели, относящиеся к классу «Вода»)

Уже для окрестности 7x7 было достигнуто значение 99,90% верно классифицированных пикселей на всех тестовых данных. Именно это значение окрестности было выбрано для сегментации изображений. Результаты классификации изображения ERS-2 на два класса («Вода» и «Суша») представлены на рис. 1.

Выводы. В данной работе описан новый нейросетевой подход для определения затопленных территорий на основе спутниковых данных. Преимуществом предложенного подхода по сравнению с известными методами определения площадей затопленных территорий является не попиксельная обработка, а использование контекстной (текстурной) информации, определяемой значениями соседних пикселей из некоторой окрестности. Кроме того, предложенный метод предполагает автоматическую настройку коэффициентов модели на основе принципов самоорганизации или так называемое обучение без учителя, что дает возможность использовать данный метод в автоматических сервисах экологического мониторинга. Точность классификации построенной нейронной сети с размерностью входа 49 и решеткой нейронов 7x7 составила 99,90% на тестовых данных, что существенно превышает известные в литературе результаты классификации другими методами. Применение предложенного подхода на данных спутника ERS-2 во время наводнений на р. Тиса 2001 г. показало высокую точность определения затопленных территорий

Литература

1. Committee on Earth Observation Satellites Disaster Management Support Group: The Use of Earth Observing Satellites for Hazard Support: Assessments & Scenarios // Final Report; National Oceanic & Atmospheric Administration, Department of Commerce, USA. — 2001.

2. Solheim I., Solbo S., Indregard M., Lauknes I. User Requirements and SAR-Solutions for Flood Mapping // 4th International Symposium on Retrieval of Bio-and Geophysical Parameters from SAR Data for Land Applications, Innsbruck, Austria, 2001.

3. Yang C., Zhou C., Wan Q. Deciding the Flood Extent with RADARSAT SAR Data and Image Fusion // 20th Asian Conference on Remote Sensing, 1999.

4. ESA Earth Watch. — http://earth.esa.int/ew/floods/.

5. Haykin S. Neural Networks: A Comprehensive Foundation. – Upper Saddle River, New Jersey: Prentice Hall, 1999.

6. European Topic Centre on Terrestrial Environment. – http://terrestrial.eionet.eu.int/CLC2000.

7. Gonzalez R.C., Woods R.E. Digital Image Processing. – Prentice Hall, Upper Saddle River: New Jersey, 2002. – 793 p.

Получено 01.06.07

УДК.744 (075.8)

ИНФОРМАЦИОННАЯ И ФУНКЦИОНАЛЬНАЯ МОЩНОСТЬ СЦЕН СВР

В.Г. Ли
Таганрогский технологический институт
Южного федерального университета

Работа посвящена проблеме оценки производительности визуализаторов графических систем при условии обеспечения достаточной статической и динамической реалистичности виртуальных сцен.

Информационное состояние визуализатора любой графической системы определяется, в первую очередь тремя параметрами (рис. 1):

- объемом информации статической модели макета – ось А;
- динамикой поведения (движения) всех подвижных объектов макета – ось В;
- динамикой поведения (движения) наблюдателя – ось С.

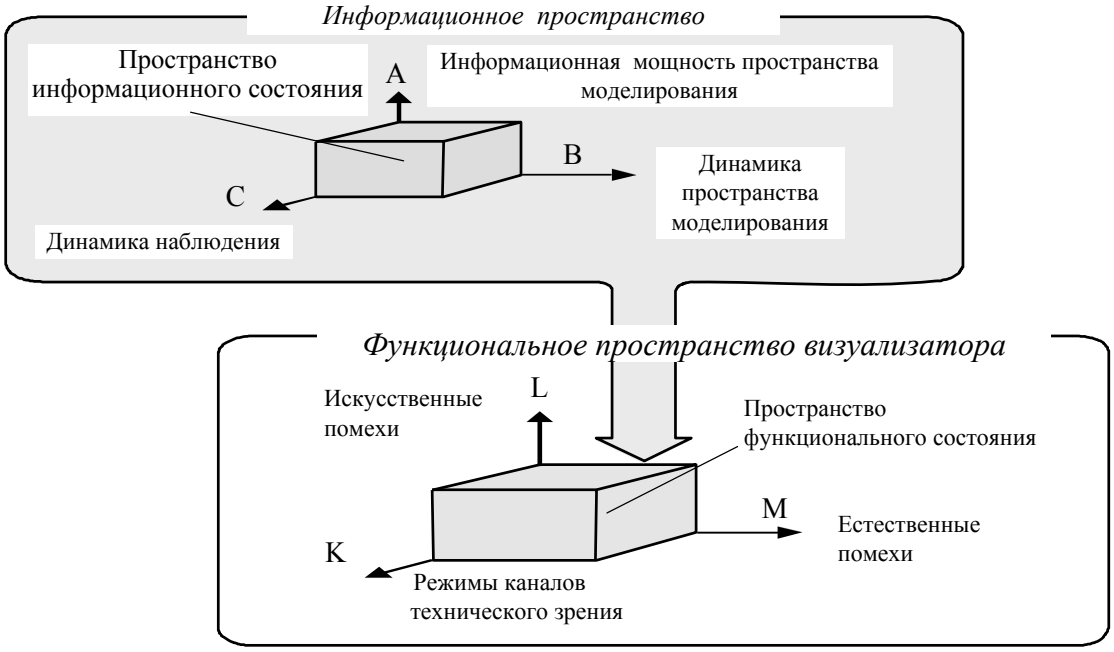


Рис. 1

По вертикальной оси А отображается количество геометрической информации, содержащейся в статическом макете пространства моделирования.

Объем информации, обрабатываемой визуализатором, зависит

также и от динамических характеристик объектов и наблюдателя. Причем эта зависимость будет возрастать пропорционально повышению скорости и сложности движений наблюдаемых объектов.

Значение функции A – суммарное количество геометрической информации всех элементов пространства моделирования. Информационное состояние визуализатора может быть представлено значением функции вида

$$I = f_1\{A, B, C\},$$

причем, произведение A×B×C можно использовать как количест-венную характеристику – интегрированную информационную мощ-ность пространства моделирования.

Функциональное пространство визуализатора с точки зрения задач визуализации определяется, в основном, тремя характеристиками:

- объемом и сложностью моделируемых искусственных помех и эффектов: текстуры поверхностей, трансформации объектов, специальные эффекты искусственного освещения, вносимые средствами технического зрения и пр. – L;
- объемом и сложностью моделируемых естественных помех и эффектов: имитация погодных условий, результаты естественного освещения и пр. – ось M;
- видом наблюдения, режимами работы оптической системы средства технического зрения: разрешающая способность, фокусное расстояние, кратность масштабирования и т.д. – ось K.

По аналогии с информационной оценкой функциональная мощность визуализатора может быть оценена функционалом вида

$$\Omega = f_2\{K, \Lambda, M\}.$$

Процесс моделирования можно представить как процесс "погружения" пространства I в пространство Ω, что находит отражение в стратегии избранной схемы прохождения геометрической информации и формирования кадров синтезированных изображений.

Изложенная схема формального представления информационного и функционального пространств визуализатора позволяет, с одной стороны, количественно оценить его производительность, а с другой стороны – отразить его конкретное состояние, с точки зрения полноты загрузки его вычислительных и графических ресурсов, а именно в виде функционала

$$V = f_4\{I, \Omega\}.$$

Количественно функционал V оценивается произведением I×Ω.

Необходимо отметить, что с точки зрения организации вычислительных процессов визуализатора и его компиляторов, функционалы I×Ω практически независимы.

При работе с пространством моделирования, а также с компилятором сценариев, используется одна глобальная мировая система 3-мерных координат и локальные системы координат объектов и наблюдателя. Кроме двух перечисленных систем координат, в соответствие со схемой прохождения геометрической информации используются также:

- локальная 3-мерная система зоны обзора;
- локальная 2-мерная система сцены;
- единая 2-мерная локальная система координат экрана и регенерирующих его кадров.

На всех этапах работы визуализатора фиксируется внутреннее время визуализатора, измеряемое в секундах. Быстродействие визуализатора оценивается величиной – N кадров/секунду (целое число).

Систематизация видов движений объектов СВР

Объекты, модели которых подлежат визуализации, должны быть реалистичными. В понятие реалистичности изображений входит и реалистичность траекторий их движений и форм движений. Как известно, объект 3-мерного пространства обладает в общем случае шестью степенями свободы, а именно:

- перемещение с различными линейными составляющими по трем координатным осям;
- вращение вокруг тех же трех координатных осей с различными составляющими углов: курсовой угол, угол тангажа, угол крена.

В общем случае движение представляет собой совокупность трансляций и вращений, такое полное движение будем называть составным сложным движением. Такая систематизация наиболее алгоритмична с точки зрения матричной вычислительной реализации каждого из видов движения, а также с позиций распараллеливания вычислительного процесса.

Для формализации характера движений объектов целесообразно ввести специальную классификацию динамики их поведения, что позволит осуществлять оценочную информационную градацию по осям движений.

Введем условное информационное понятие количества движения объекта. Для оценки этого количества каждому виду

движения присваивается числовой код, структура которого имеет вид (рассмотрим на примере оси В схемы информационного пространства):

$$B = \langle B1, B2, B3 \rangle,$$

где: В1 – признак пространства и условий движения, который может принимать значения:

1 – движение осуществляется по горизонтально расположенной плоскости, то есть объект обладает только тремя степенями свободы;

2 – движение осуществляется в воздушном пространстве, то есть объект обладает шестью степенями свободы. Причем все характеристики положения объекта в пространстве однозначно зависят только от вида заданной ему траектории движения;

4 – движение осуществляется по сложной поверхности. В этом случае объект также имеет шесть степеней свободы, но динамически изменяющиеся составляющие его положения в пространстве зависят не только от заданной траектории, но и от геометрических характеристик поверхности, по которой осуществляется движение.

В2 – признак линейной составляющей движения, который может принимать значения:

0 – отсутствие движения;

1 – движение вдоль одной из координатных осей глобальной системы координат с постоянной скоростью;

2 – движение с составляющими по двум или трем координатным осям с постоянной скоростью;

3 – произвольное прямолинейное движение со сменой направлений с постоянной скоростью;

4 – произвольное прямолинейное движение со сменой направлений с переменной скоростью. Изменение скорости происходит посредством положительного или отрицательного ускорения;

5 – сложное движение, содержащее не менее двух элементов данной классификации, например - движение 4 сочетается с периодами движения с постоянной скоростью без смены направления движения;

7 – составное движение, завершающееся особой формой обработки изображения объекта, а именно - "останов" и полное исчезновение объекта;

8 – завершающееся частичным исчезновением объекта;

9 – завершающееся трансформацией объекта.

В3 – признак вращательной составляющей движения объекта, может принимать значения:

0 – отсутствие вращения;

1 – вращение вокруг одной локальной оси с постоянной угловой скоростью (в рассматриваемых условиях имеется в виду вращение по курсовому углу, то есть вокруг оси z);

2 – вращение вокруг двух или трех локальных осей с постоянной угловой скоростью;

3 – сложное вращение, то есть вращение вокруг произвольного количества осей, но с переменной угловой скоростью. Перемена скорости осуществляется с помощью положительного или отрицательного ускорения;

далее, – аналогично классификации признака B2.

Таким образом составное сложное движение описывается сочетанием ненулевых значений признаков B2 и B3.

Количество движения оценивается величиной B=B1×(B2+B3). Тогда по оси B схемы информационного пространства можно ввести градацию (в общем случае неравномерную) от нуля до 51.

Специфика алгоритмов решения задач взаимной видимости визуально конкурирующих объектов требует, чтобы общее движение объектов макета оценивалось операцией перемножения. Аналогично производится градация оси C – динамики наблюдателя.

Технология оценки движений наблюдателя

Информационно наблюдатель представляет собой подвижный объект, но не подлежащий реальной визуализации. Фактически он представлен в информационном пространстве лишь своей локальной системой координат. Все перечисленные выше виды движений (исключая движение в воздухе) могут им отрабатываться. Однако специфическим свойством этого объекта является то, что необходимо реализовать интерактивное управление наблюдателем и средством технического зрения.

Структурно наблюдатель представляет собой составной подвижный объект, состоящий из двух связных объектов, причем в общем случае их локальные системы координат не имеют общих элементов. Составляющие объекты могут осуществлять синхронные, но независимые движения. Условно первый объект назовем платформой, второй – телекамерой.

Значение функции K для телекамеры можно представить в виде:

$$K = f_3\{K1, K2, K3\},$$

где K1 – признак вращения оптической оси телекамеры, который может принимать следующие значения:

1 – ось неподвижна;

2 – ось может перемещаться вращением вокруг одной локальной оси, например, по углу визирования с постоянной угловой скоростью;

3 – то же, но с переменной угловой скоростью. Изменение скорости осуществляется с положительным или отрицательным ускорением;

4 – ось может вращаться одновременно вокруг двух локальных осей с постоянной угловой скоростью;

5 – то же, но с переменной скоростью;

K2 – признак изменения телесных углов оптической системы телекамеры, может принимать значения:

1 – узкое поле зрения;

2 – широкое поле зрения (зона обзора резко увеличивается);

K3 – признак масштабирования изображения, может принимать значения:

1 – операция масштабирования отсутствует;

2 – операция масштабирования используется.

При этом телекамера должна иметь синхронные с платформой движения, которые значительно увеличивают количество движения телекамеры. Предлагается в этом случае использовать следующую оценку количества движения наблюдателя:

$$K = B \times K1 \times (K2 + K3),$$

где B – количество движения платформы наблюдателя.

Учитывая изложенные виды движений, можно конкретизировать оценку информационной мощности визуализатора (функционал I), если ввести понятие относительной информационной мощности макета. Под этой характеристикой понимается количество геометрической информации, обрабатываемой в текущей зоне обзора пространства моделирования при формировании очередного кадра изображения. Эта величина более объективно и точно отражает характеристику быстродействия визуализатора и непосредственно влияет на частоту регенерации кадров экрана.

При этом появляется возможность оценки количества возникающих задач коллизий, вызванных взаимным и относительным движением как самих объектов наблюдения, так и движения наблюдателя относительно движущихся наблюдаемых объектов.

Наличие движения как объектов наблюдения, так и самого наблюдателя увеличивает информационную нагрузку на визуализатор в пределах формирования одного экрана изображения. Поэтому информационно пространство визуализатора может быть оценено объемом соответствующего параллелепипеда со сторонами – I, Ω,V.

Получено 10.05.07 ТТИ ЮФУ, г. Таганрог © В.Г. Ли, 2007

УДК 681.3.06:681.327.2

МЕТОДЫ РЕШЕНИЯ ЗАДАЧ АНАЛИЗА ВОЗДУШНОЙ ОБСТАНОВКИ ПРИ ЕЁ ПРЕДСТАВЛЕНИИ В РЕАЛЬНОМ ВРЕМЕНИ

М.И. Васюхин, О.И. Капштык, С.А. Пономарев
Институт кибернетики имени В.М. Глушкова НАН Украины.

В роботі висвітлюються підходи до вирішення статистичних задач, пов'язаних з аналізом повітряної обстановки в реальному часі. Запропонована математична формалізація окремих задач пошуку в геометричній області, що представлена найпростішими графічними примітивами.

Задачи анализа воздушной обстановки. Одной из классических задач анализа воздушной обстановки при её представлении оперативному персоналу в реальном времени есть задача разводки самолётов в районе крупного аэропорта. Эта задача требует применения поиска в геометрической области. Раньше такие задачи решались визуально операторами при считывании информации с экранов кругового обзора непосредственно на аппаратуре наблюдения или с большого рабочего планшета. При этом, естественно, возникали ошибки, связанные с человеческим фактором и со значительными погрешностями отображения обстановки. Особенно часто эти ошибки встречались в критические моменты работы служб наблюдения за воздушной ситуацией, когда экраны предельно загружены информацией. Закономерно, что в такие моменты информация должна быть как можно более точной и полной для принятия адекватных обстановке решений, особенно тогда, когда какие-либо ошибки вообще недопустимы.

Поэтому логично возложить на компьютер некоторые хорошо формализованные задачи анализа воздушной обстановки. Несомненно, увеличение размеров массивов информации, связанное с ростом количества объектов, находящихся в воздухе, влияет на работу автоматизированной системы управления полётами, ведёт к увеличению времени разовой обработки массивов, но при использовании «хороших» алгоритмов количество одновремённо сопровождаемых объектов может быть увеличено в несколько раз. Первой задачей, которая имеет непосредственное отношение к поиску в геометрической области, есть определения количества объектов некоторого типа, которые находятся в заданном регионе. Конкретным примером может быть определение общего количества объектов, что находятся в воздухе, например, в 50-километровой зоне вокруг

аэропорта или количества только пассажирских самолётов определённого типа. При этом оператор должен иметь возможность изменения радиуса зоны или её формы, например, в виде треугольника, вписанного в сектор, или многоугольника.

Обозначение постановки, находится ли объект определенного типа в заданной области, можно рассматривать как упрощённый вариант предыдущей задачи. Здесь достаточно просто подсчитать количество необходимых объектов в области и, сравнив его с нулем, выдать ответ. При этом из-за своей простоты эта задача допускает более простые и быстрые алгоритмы. Кроме этого, для области запроса можно использовать геометрическую область более сложного построения, чем представленную в виде примитива. Пример реальной задачи: как только летательный объект появляется в заданной области например, в зоне повышенного внимания, распознать его и сопровождать в реальном времени.

Для решения ещё одного типа характерных для этого случая задач рассмотрим такую ситуацию. Известно среднестатистическое количество технических средств, необходимых для сопровождения определённого типа летательных объектов. Необходимо определить способность операторов диспетчерского центра следить за всеми самолётами в определенной зоне. Общая формулировка такой задачи: каждой точке на плоскости присвоен определённый вес, найти сумму весов всех точек, которые попали в область запроса. Ниже будет показано, как может задач поиска в геометрической области за счет выбора веса и операции сложения быть сведены к обозначенной постановке, что определяет эту задачу как наиболее общую.

Последним типом возможных задач поиска в геометрической области есть задача поиска и выведения списка всех точек определённого типа, что попали в область запроса, например, объектов, которые не сопровождаются оператором.

Этим круг возможных практических задач, так или иначе связанных с геометрическим поиском, безусловно, не ограничивается. Но даже сказанное выше указывает на важность задач такого рода, что могут быть как самостоятельными задачами для анализа и исправления, так и процедурными подзадачами. При этом автоматизированная система, обеспечивающая через определённые промежутки времени запросы в геометрических областях, информировала бы оператора, например, о текущей обстановке в зоне аэропорта.

Возможности системы могут быть значительно расширены в том случае, когда оператор дополнительно может сам определить и применить простейший геометрический примитив как область запроса.

Такая необходимость может возникнуть, например, в случае срочной посадки поврежденного самолёта.

Очень жестким требованиям к автоматизированной системе есть требование реализации режима реального времени. Это означает, что система, которая выполняет отмеченные запросы, должна успевать находить ответ за «приемлемое» время. Теоретически эффективность «быстрых» алгоритмов измеряется почти исключительно асимптотическим порядком роста их сложности. Языком алгоритмов «приемлемое» значит, что время запроса имеет не более чем полиномиальную сложность зависимости от количества входных данных.

Целью данной работы есть нахождение алгоритмов поиска в определенной геометрической области с последующим выбором из них наиболее эффективного с целью проведения анализа воздушной обстановки в режиме реального времени.

Для упрощения воздушные объекты можно представить в виде точек. Под пространством, в котором производится поиск, подразумевается воздушное пространство в виде его двумерной проекции представленной картой местности с возможными дополнительными данными, которые, например, указывают тип самолёта (свой, чужой; грузовой, пассажирский и т.д.). Целью алгоритма есть построение такой структуры данных про объекты, чтоб для любой области пространства, что задается, время нахождения данных об этих объектах в этой области было оптимальным. Здесь под оптимальностью подразумевается наименьшее время поиска для любых выбранных областей и координат точек. С учетом того, что в реальной практике нам нужны не все области, а области некоторых простых классов, рассмотрим задачи поиска в областях этих классов.

Уточненная математическая постановка задачи. Перед тем, как формализовать задачу, связанную с поиском в геометрической области, рассмотрим понятие геометрической области. Для размерности 2, то есть плоского случая, это часть плоскости, ограниченная замкнутой кривой без самопересечений. Однако задать такую область в общем случае затруднительно. Естественный подход состоит в том, чтобы ограничить множество рассматриваемых областей и использовать только так называемые геометрические примитивы, то есть области простейшей формы, а также производные от них применением операций наложения, часто называемых оверлейными.

Пусть $\Re$ — система подмножеств d - мерного Евклидового пространства R^d следующего вида:

$\Re_{ортог}$ = параллельные осям параллелепипеды, т.е., все множества вида $\prod_{i=1}^{d}[a_i, b_i]$, $a_1, b_1, ..., a_d, b_d \in R$.

$\Re_{полупр}$ = множество всех (замкнутых) полупространств в R^d.

$\Re_{симлекс}$ = множество всех (замкнутых) симплексов в R^d.

$\Re_{шар}$ = множество всех (замкнутых) шаров в R^d.

Множества из $\Re$ называются областями.

В результате наложения областей образуются производные области. В алгоритмах, реализующих операции наложения, могут применяться логические операции типа NOT, AND, OR, XOR. На рисунке 1.б показаны варианты применений этих операций к пересекающимся двумерному шару A и двумерному симплексу B, которые изображены на рисунке 1.а.

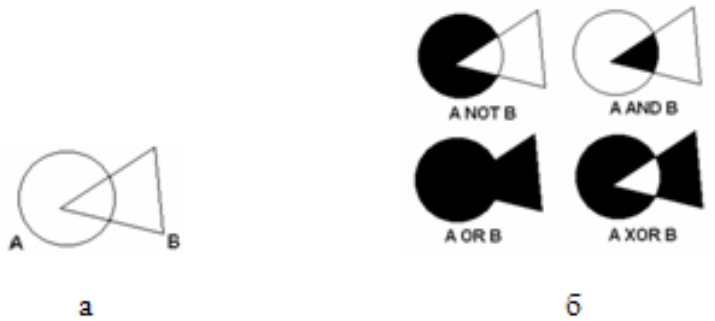


Рис. 1. Варианты оверлейных операций.

В дальнейшем, задается P — n-точечное множество в R^d. Предполагается, что такое множество находится в общей позиции, т.е. никакие три точки из этого множества не лежат на одной прямой. Если данное условие не выполняется, производится так называемое [1] «моделирование простоты»: входные точки смещаются на бесконечно малые величины для того, чтобы привести их к общей позиции, с условием, что никакие три точки из этого множества не лежат на одной прямой. Это не приводит к значительному искажению результата, хотя реальные входные данные как правило неточны по своей природе координаты точек обладают некоторой погрешностью измерений.

Сформулируем основные задачи поиска в геометрической области:

з.1) разработать эффективный алгоритм, который по заданной области $R \in \Re$ находит число точек P, лежащих в R;

з.2) разработать эффективный алгоритм, который определяет, содержит ли область запроса R хотя бы одну точку из P;

з.3) каждой точке назначен некоторый вес, разработать эффективный алгоритм, который по заданной области $R \in \Re$ определяет сумму весов всех точек из P, лежащих в R;

з.4) разработать эффективный алгоритм, который выдает список всех точек из P, лежащих в области запроса R.

Задача (з.3) является наиболее общей [1]. Задачи (з.1), (з.2) и (з.4) могут быть сведены к ней выбором подходящей операции суммы и весов точек. А именно: пусть каждой точке *p* из *P* назначен вес *w(p)* из полугруппы *S* с операцией сложения «+». Для задачи (з.1) достаточно все веса положить равными 1 и за полугруппу *(S, +)* взять полугруппу натуральных чисел относительно сложения. Для запросов о пустоте в качестве полугруппы подойдет множество булевых значений («истина» и «ложь») с операцией логической дизъюнкции (ИЛИ), веса всех точек следует взять равными «истине».

Отчетные запросы, которые выдаются при решении задачи (з4), также могут быть включены в полугрупповую модель задачи (з2). Здесь полугруппа *(S, +)* – это множество всех подмножеств *P* с операцией объедения множеств, а вес каждой точки *p* из *P* – одноточечное множество *{p}*. Однако в существующих моделях вычислений, такое подмножество не удается представить постоянным объемом памяти, а операцию объедения выполнить за постоянное время. Отметим, что для вывода ответа, который состоит из k точек, необходимо время порядка *k* только для вывода отчета, при этом сложность алгоритмов отчетного запроса в области выражается в форме *O(f(n)+k)*, где *f(n)* – некая функция общего количества точек *P*.

Выводы

В работе поданы подходы к решению статистических задач, связанных с анализом воздушной обстановки в реальном времени. Предложена математическая формализация отдельных задач поиска в геометрической области, представленной простейшими графическими примитивами.

Литература

1. Jirzi Matousek. Geometric Range Searching. ACM Computing Surveys, Vol/ 26-1994.- №4.- p. 421-461.
2. Aggarwal A, Hansen M, and Leighton T. Solving query-retrieval problems by compacting Voronoi diagrams. In Proceedings of the 22nd Annual ACM Symposium on Theory of Computing. – New York: ACM, 1990. – P. 331-340.

Получено 15.05.2007

УДК 681.5

ПАРАЛЛЕЛЬНЫЕ БЛОЧНЫЕ МЕТОДЫ РЕШЕНИЯ ЗАДАЧИ КОШИ ДЛЯ ОБЫКНОВЕННЫХ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ

Фельдман Л.П.
Кафедра ПМиИ ДонНТУ

Рассмотрены обобщенные блочные многошаговые многоточечные методы решения задачи Коши для системы обыкновенных дифференциальных уравнений. Получены расчетные формулы для общих m- шаговых k- точечных блочных методов, определен порядок их точности. Определены условия устойчивости по Далквисту и доказана сходимость устойчивых по начальным данным общих m- шаговых k- точечных блочных методов к точному решению. Получена априорная оценка погрешности приближенного решения. Полученные результаты представляют возможности построения новых более эффективных параллельных алгоритмов и их реализации на современных мультипроцессорных вычислительных системах

Введение

Доклад содержит обобщение результатов исследований, опубликованных в [1,2,3,4,5,6], посвященных параллельным методам численного решения задачи Коши для системы обыкновенных дифференциальных уравнений и является продолжением ранее опубликованных работ [7,8,9,10,11,12,13]. В нем рассматривается устойчивость решения по начальным данным, получена оценка точности решения, приводится доказательство сходимости приближенного решения для общих *m*- шаговых *k*- точечных блочных методов, что представляет обобщение ранее опубликованных результатов. Приведен пример решения блочным разностным методом и даны практические рекомендации их использования для более широкого набора параллельных разностных схем.

1. Погрешность аппроксимации блочных методов

Для численного решения задачи Коши

$$\frac{dx}{dt} = f(t,x), t > 0, x(0) = x_0 \qquad (1)$$

используется блочный m-шаговый k-точечный разностный метод вида

$$\frac{1}{\tau}\sum_{j=1-m}^{k} a_{i,j}u_{n,j} = \sum_{j=1-m}^{k} b_{i,j}F_{i,j}\ ,\ i=\overline{1,k}\,, \qquad (2)$$

Коэффициенты уравнений (2) определены с точностью до множителя. Чтобы устранить этот произвол, будем считать, что выполнено условие

$$\sum_{j=1-m}^{k} b_{i,j} = 1,\ i=\overline{1,k}. \qquad (3)$$

Показано, что погрешность аппроксимации имеет порядок p, если выполнены условия

$$\sum_{j=1-m}^{k} a_{i,j} = 0;\quad \sum_{\substack{j=1-m,\\ j\neq 0}}^{k} [\,ja_{i,j} - b_{i,j}\,] = 0\,,$$

$$\sum_{\substack{j=1-m,\\ j\neq 0}}^{k} [\frac{a_{i,j}j}{s} - b_{i,j}\,] * j^{s-1} = 0, s=\overline{2,p},\ i=\overline{1,k}. \qquad (4)$$

Каждое уравнение системы (4), (3), соответствующее фиксированному значению i, может содержать $2(k+m)$ неизвестных $a_{i,j}$, $b_{i,j}$. Для того чтобы система разностных уравнении, построенных на одном наборе узлов сетки, была линейно независима, шаблоны разностных схем для уравнений должны отличаться между собой хотя бы одним вхождением либо коэффициента $a_{i,j}$ либо $b_{i,j}$. Отсюда следует, что наивысший порядок аппроксимации m-шагового k-точечного блочного метода не может превосходить $p = 2(k + m)-3$. Его погрешность определяется формулой

$$r_{n,i} = \frac{\tau^{2(k+m)3}}{[\,2(k+m-1)-1]!}x^{2(k+m)-3}{}_{n,0}\sum_{j=1-k}^{m}[\frac{a_{i,j}}{2(k+m-1)} - \frac{b_{i,j}}{j}]\,j^{2(k+m)-3},\ i=\overline{1,k} \quad (5)$$

Приведем пример двухшаговой двухточечной разностной схемы наивысшего (пятого) порядка точности, полученной по формулам (3),(4).

$$-\tfrac{11}{30}u_{n-1} - \tfrac{4}{15}u_n + \tfrac{19}{30}u_{n+1} = \tau(\tfrac{1}{9}F_{n-1} + \tfrac{19}{30}F_n + \tfrac{4}{15}F_{n+1} - \tfrac{1}{90}F_{n+2}),$$

$$\tfrac{4}{15}u_{n-1} - \tfrac{9}{10}u_n + \tfrac{19}{30}u_{n+2} = \tau(-\tfrac{1}{10}F_{n-1} + \tfrac{9}{10}F_{n+1} + \tfrac{1}{5}F_{n+2})$$

2. Общая процедура интегрирования

Обозначим матрицы коэффициентов системы уравнений (2) через

$$\boldsymbol{A} = (\boldsymbol{a}_{i,j}),\ \ \boldsymbol{B} = (\boldsymbol{b}_{i,j}), \boldsymbol{i} = \overline{1,\boldsymbol{k}}, \boldsymbol{j} = -(\overline{\boldsymbol{m}-1}), \boldsymbol{k}\,.$$

Разобьем каждую из матриц на две части

$$A_1 = (\,a_{i,j}\,),\ \boldsymbol{B}_1 = (\boldsymbol{b}_{\boldsymbol{i},\boldsymbol{j}}), \boldsymbol{i} = \overline{1,\boldsymbol{k}}, \boldsymbol{j} = -(\overline{\boldsymbol{m}-1}),0\,,$$

$$\boldsymbol{A}_2 = (\boldsymbol{a}_{\boldsymbol{i},\boldsymbol{j}}),\ \boldsymbol{B}_2 = (\boldsymbol{b}_{\boldsymbol{i},\boldsymbol{j}}), \boldsymbol{i} = \overline{1,\boldsymbol{k}}, \boldsymbol{j} = \overline{1,\boldsymbol{k}}$$

Введем соответствующие им вектора

$$U_n = \left(u_{n,j}\right), j = \overline{1-m,0}, V_{n+1} = \left(u_{n,j}\right), n = 1,2,...,, j = \overline{1,k}.$$

$$\boldsymbol{F}_{\boldsymbol{n}} = \left(\boldsymbol{F}_{\boldsymbol{n},\boldsymbol{j}}\right), \boldsymbol{j} = \overline{1-\boldsymbol{m},0}, \boldsymbol{G}_{\boldsymbol{n}+1} = \left(\boldsymbol{F}_{\boldsymbol{n},\boldsymbol{j}}\right), \boldsymbol{j} = \overline{1,\boldsymbol{k}}.$$

В векторной форме уравнение (2) будет иметь вид

$$\boldsymbol{A}_1 \boldsymbol{U}_{\boldsymbol{n}} + \boldsymbol{A}_2 \boldsymbol{V}_{\boldsymbol{n}+1} = \tau\left(\boldsymbol{B}_1 \boldsymbol{F}_{\boldsymbol{n}} + \boldsymbol{B}_2 \boldsymbol{G}_{\boldsymbol{n}+1}\right)$$

Предполагая, что матрица $\boldsymbol{A}_2$ невырожденная, разрешим последнее уравнение относительно $\boldsymbol{V}_{\boldsymbol{n}+1}$

$$\boldsymbol{V}_{\boldsymbol{n}+1} = \boldsymbol{S}\,\boldsymbol{U}_{\boldsymbol{n}} + \tau\left(\Phi_1 \boldsymbol{F}_{\boldsymbol{n}} + \Phi_2 \boldsymbol{G}_{\boldsymbol{n}+1}\right), \tag{6}$$

где $S = -A_2^{-1} A_1, \boldsymbol{\Phi}_1 = A_2^{-1} B_1, \boldsymbol{\Phi}_2 = A_2^{-1} B_2$.

Если решение разностной задачи сходится к решению исходной задачи, то сначала вычислим стартовые значения, например, с помощью явных формул Рунге – Кутта

$$\boldsymbol{U}_0 = (\boldsymbol{u}_{0,\boldsymbol{j}}, \boldsymbol{j} = \overline{0,\boldsymbol{m}-1}),$$

затем на каждом последующем этапе необходимо решить нелинейное уравнение (6), определив последовательно вектора $\boldsymbol{V}_1, \boldsymbol{V}_2, \ldots$, для соответствующих значений векторов $U_0, U_1, U_2, \ldots$

3. Устойчивость по начальным данным

Задача Коши для однородного уравнения, соответствующего (6) состоит в отыскании сеточной функции $\boldsymbol{V}_{\boldsymbol{n}+1} = \{\boldsymbol{u}_{\boldsymbol{n}+\boldsymbol{j}}, \boldsymbol{j} = \overline{1,\boldsymbol{k}}\}$, удовлетворяющей при всех $\boldsymbol{n} \geq 0$ уравнению

$$\boldsymbol{V}_{\boldsymbol{n}+1} = \boldsymbol{S}\,\boldsymbol{U}_{\boldsymbol{n}} \tag{7}$$

при $\boldsymbol{U}_{\boldsymbol{n}} = \{\boldsymbol{u}_{\boldsymbol{n}-\boldsymbol{j}}, \boldsymbol{j} = \overline{0,\boldsymbol{m}-1}\}$ и принимающей при $\boldsymbol{n} = 0$ заданные начальные значения

$$\boldsymbol{U}_0 = \{\boldsymbol{u}_{\boldsymbol{j}}, \boldsymbol{j} = \overline{0,\boldsymbol{m}-1}\}$$

Также как и в случае многошаговых разностных методов, рассмотренных в [1,3,4,5], устойчивость или неустойчивость уравнения (7) по начальным данным определяется расположением корней характеристического уравнения матрицы $\widetilde{\boldsymbol{S}}$. Доказано. *Если все корни характеристического уравнения матрицы $\widetilde{S}$, лежат внутри или на границе единичного круга, причем на границе круга нет кратных корней, то блочный многошаговый многоточечный метод устойчив по начальным данным.*

Получены выражения для матриц перехода $\widetilde{S}$ для общих блочных многошаговых многоточечных методов для случаев $m>k$ и $m<k$.

4. Устойчивость по правой части

В предыдущем пункте была приведена оценка решения однородного уравнения (7) через начальные данные. Приведем здесь оценку решения неоднородного уравнения.

$$\boldsymbol{V}_{\boldsymbol{n}+1} = \boldsymbol{S}\,\boldsymbol{U}_{\boldsymbol{n}} + \tau\left(\Phi_1\,\boldsymbol{F}_{\boldsymbol{n}} + \Phi_2\,\boldsymbol{G}_{\boldsymbol{n}+1}\right) \qquad (6)$$

Представив уравнение (6) в виде

$$Y_{n+1} = \widetilde{S}\,Y_n + \boldsymbol{\tau}\,H_n$$

Получена оценка

$$\left\|Y_{k+1}\right\|_C \le M_1\left(\max_{0\le j\le m-1}\left|y_j\right| + \boldsymbol{\tau}\sum_{n=1}^{l}\left\|H_n\right\|_c\right), \qquad (8)$$

Выполнение оценки (8) означает по определению устойчивость уравнения (6) по правой части

5. Оценка погрешности разностного метода

Обозначив вектор погрешности решения разностного уравнения (24) через

$$\boldsymbol{\Sigma}_n = Y_n - X(t_n).$$

Справедлива теорема

Если выполнено условие корней, $\left\|\boldsymbol{\Sigma}_1\right\|_* \to 0$ *при* $\boldsymbol{\tau} \to 0$ *и разностное урав нение (2) аппроксимирует исходное уравнение (1), то решение разностной задачи (2) сходится при* $\boldsymbol{\tau} \to 0$ *к решению исходной задачи (1).*

Приводятся примеры решения задач Коши, рассмотренными блочными многошаговыми многоточечными методами

Заключение. Общие блочные многошаговые многоточечные методы, рассмотренные в докладе, представляют обобщение известных многошаговых методов и существенно расширяют класс разностных методов решения задач Коши для обыкновенных дифференциальных уравнений. Проведенный анализ этих методов позволил получить оценки их устойчивости по Далквисту, сходимости по правой части и оценки погрешности. Полученные результаты представляют возможности построения новых более эффективных параллельных алгоритмов и их реализации на современных мультипроцессорных вычислительных системах.

Литература

1. Дж. Холл, Дж. Уатт. Современные численные методы решения обыкновенных дифференциальных уравнений. М.: Мир,1979.-312с.
2. Системы параллельной обработки. Под ред. Ивенса Д. – М.:Мир,1985. – 416с.
3. Самарский А.А., Гулин А.В. Численные методы.- М.:Наука.1989.-432 с.

4. Хайрер Э., Нёрсет С., Ваннер. Г. Решение обыкновенных дифференциальных уравнений. Нежесткие задачи. - Мир,1990.-512 с .

5. Хайрер Э., Ваннер. Г. Решение обыкновенных дифференциальных уравнений. Жесткие задачи.-Мир,1999.-685 с.

6. Молчанов И.Н. Введение в алгоритмы параллельных вычислений. АН УССР, Инст. Кибернетики им. В.М. Глушкова. Киев: Наукова думка. 1990,-128с.

7. Фельдман Л.П. Параллельные интерполяционные алгоритмы численного решения задачи Коши для обыкновенных дифференциальных уравнений на SIMD компьютере. Научн. Тр. ДонГТУ. Серия: Проблемы моделирования и автоматизации проектирования динамических систем, выпуск 10:- Донецк: ДонГТУ,1999, с. 20-25.

8. Фельдман Л.П. Сходимость и оценка погрешности параллельных одношаговых блочных методов моделирования динамических систем с сосредоточенными параметрами. Научн. Тр. ДонГТУ. Серия: Інформатика, Кібернетика та обчислювальна техніка, выпуск 15:- Донецк: -ДонГТУ,2000, с. 34-39.

9. L.P. Feldmann Implementierung und Effizienzanalyse von parallelen blockartigen Simulati onsalgorithmen für dynamische Systeme mit konzentrierten Parametern. In: Möller, D.P.F. (Hrsg.): Tagungsband 14. ASIM-Symposium Simulationstechnik in Hamburg, September 2000, SCS-Europe BVBA, Ghent/Belgium 2000, S. 241-246.

10. Фельдман Л.П., Дмитриева. О.А. Разработка и обоснование параллельных блочных методов решения обыкновенных дифференциальных уравнений на SIMD– структурах. Научн. Тр. ДонГТУ. Серия: Проблемы моделирования и автоматизации проектирования динамических систем, выпуск 29:- Донецк: ДонГТУ,2001, с. 70-79.

11. L.P. Feldman, O.A. Dmitrieva, S. Gerber. Abbildung der blockartigen Algorithmen auf Parallelrechnerarchitekture. In: Tavangarian,D., Grützner,R. (Hrsg.): Tagungs-band 15. ASIM-Symposium Simulationstechnik in Rostock, September 2002, SCS-Europe BVBA, Ghent/Belgium 2002, S.359-364.

12. Фельдман Л.П., Параллельные алгоритмы моделирования динамических систем, описываемых обыкновенными дифференциальными уравнениями.//Электронное моделирование, том 26, № 1, 2004.- С.19-30.

13. Фельдман Л.П., Назарова И.А., Параллельные алгоритмы численного решения задачи Коши для систем обыкновенных дифференциальных уравнений Ж. Математическое моделирование. 2006. т.18, №6, с. 17-31.

Получено 01.06.07

УДК 004.383.4

АРХЕОМОДЕЛИРОВАНИЕ: ДОЦИФРОВАЯ ЭПОХА В ВЫЧИСЛИТЕЛЬНОМ МОДЕЛИРОВАНИИ И ЕЕ ЗНАЧЕНИЕ В КОНТЕКСТЕ ОБОБЩЕННОГО КОДО-ЛОГИЧЕСКОГО БАЗИСА

А. Я. Аноприенко

В доповіді розглядаються основні риси і визначення стосовно нового наукового напрямку, який запропоновано назвати археомоделюванням. Обґрунтовується актуальність цього напрямку і розглядаються його основні особливості в контексті концепції узагальненого кодо-логічного базису.

Археомоделирование – это новое научное направление, формирующееся на стыке технических, естественных и гуманитарных наук, и имеющее вследствие этого существенно междисциплинарный характер [1].

Основной акцент при этом делается на методы и средства моделирования, характерные для «доцифровой эпохи». Причем, речь идет преимущественно о вычислительном и/или когнитивном моделировании. Под «вычислительным моделированием» при этом понимается построение и использование таких моделей, которые связаны с разного рода вычислительными процессами, обусловленными необходимостью, например, календарных расчетов, масштабирования объектов, проведения измерений и т.п. Когнитивные свойства моделей связаны с их познавательной функцией, что предполагает использование их и как средства концентрации и систематизации определенных знаний, и как инструмент получения новых знаний.

Поэтому ***в узком смысле археомоделирование*** *можно определить как науку о методах и средствах вычислительного и когнитивного моделирования, характерных для древнего человека*. Первая часть слова «археo» (от греческого archaios - древний) как раз и призвана подчеркнуть ориентированность именно на историческое прошлое моделирования.

Подобно тому, как археология изучает прошлое по памятникам материальной культуры, находимым главным образом при раскопках,

археомоделирование также в значительной степени рассматривает в качестве объектов исследования различные артефакты древности. Но не только! Существенную роль в археомоделировании играют также и «интеллектуальные раскопки», позволяющие реконструировать реалии прошлого с помощью современных средств обработки информации, компьютерного анализа и моделирования. Поэтому ***в более широком смысле археомоделирование*** *можно определить как науку о методах и средствах моделирования, характерных для исторического прошлого, и об использовании современного компьютерного моделирования для исследования и реконструкции когнитивных реалий и интеллектуалоемких артефактов, дошедших до нас из глубин истории*.

«Доцифровая эпоха» также может толковаться двояко. С одной стороны как период до начало массового перехода на современные позиционные системы счисления, что в качестве начального рубежа предполагает позднее средневековье, а в качестве конечного – завершение научной революции в конце XVII века. В России, в частности, использование десятичной позиционной системы счисления («цифирной арифметики») стало массовым лишь в начале XVIII века [2, с. 52]. С другой стороны, началом «цифровой эпохи» многие склонны считать массовый переход на электронные цифровые вычислительные машины (ЭЦВМ) во второй половине XX века. И если под понятием «компьютер» иметь ввиду только ЭЦВМ, то ***в самом широком смысле под археомоделированием*** *можно понимать все методы и средства докомпьютерного моделирования, а также методы и средства их исследования с помощью современных компьютерных технологий.*

Сам термин «археомоделирование» предложен автором в середине 90-х годов как аналог понятия «археоастрономия», но применительно к методам средствам вычислительного моделирования. Следует отметить, что оба направления существенно взаимосвязаны. Пионером археоастрономии традиционно считают английского астронома Нормана Локьера, известного также открытием гелия на Солнце, написавшего на рубеже XIX-XX вв. фундаментальную книгу «Заря астрономии». Однако необходимость определения понятия «археоастрономия» (значительно реже используются также термины «палеоастрономия» и «астроархеология») возникла со времени становления этого научного направления, связанного с появлением в середине 60-х годов работ британского астронома, профессора Бостонского университета (США) Дж. Хокинса,

посвященных астрономическим аспектам уникального археологического памятника Стоунхендж. Его первая статья на эту тему была опубликована в 1964 году в солидном и широко читаемом журнале Nature, основанном в свое время Локьером. Эта публикация, в которой он, основываясь на полученных с помощью современного компьютера данных, доказывает, что Стоунхендж был своеобразной вычислительной машиной каменного века, почти всеми археологами была встречена весьма скептически. Но она породила многочисленные споры, кульминацией которых стало издание в 1966 году научно-популярной книги «Разгадка тайны Стоунхенджа» (на русском языке она вышла в 1973 году [3], а в 1978 году была переведена еще одна его работа [4]). К концу 60-х годов археоастрономия была признана практически всем научным сообществом как специфическое научное направление.

Современные исследователи склонны утверждать, что «археоастрономия как научное направление имеет исключительную важность для выявления не только истоков и этапов развития астрономических знаний, но и для реконструкции древних мировоззренческих комплексов, характера и содержания древних производственных циклов, этапности палеоклиматических колебаний, корректировки периодизации и хронологии археологических культур, разгадки тайн древнего мифотворчества, исследования путей и форм развития календарных систем и древнего знания в целом» [5, с.10].

Как в археоастрономии, так и в археомоделировании важнейшую роль играет возможность использования современных компьютерных технологий для реконструкции и моделирования исследуемых артефактов и феноменов.

Основатель археоастрономии Дж. Хокинс был одним из первых, кто начал использовать электронные вычислительные машины (ЭВМ) для подобных расчетов в начале 60-х годов XX века. Но ему в то время требовалась помощь двух программистов и одного оператора чтобы день программирования и минута вычислений позволяли получить такие же результаты, на которые у вычислителей, вооруженных калькуляторами уходили месяцы работы [3, с. 143]. «Революция персональных ЭВМ» в 80-х годах, плавно перешедшая в 90-х годах в «революцию компьютерных сетей», а в новом тысячелетии – в «революцию Интернет», позволила практически для всего научного мира снять проблему скорости и точности вычислений. Любой специалист отныне получил практически неограниченные (во всяком

случае, стремительно возрастающие с течением времени) возможности для достаточно быстрого выполнения сколь угодно сложных вычислений. Более того, «теперь, не выходя из-за стола, можно делать такие вычисления, которые еще недавно вообще нельзя было выполнить за разумное время» [2, с. 14].

К началу 90-х годов уже имелся достаточно большой набор общедоступных программ, не только существенно упростивших соответствующие расчеты, сделав их вполне доступными для непрофессионалов, но и обеспечивших качественно новый уровень астрономических вычислений. Суть качественного скачка заключалась в том, что новый инструментарий являлся уже не только средством для вычислений, но и включал в себя колоссальный массив справочной и экспертной информации, которая даже профессионалам ранее не всегда была доступна в таком объеме. Но самое важное состояло в том, что впервые стало возможным использование всего этого с невиданной ранее эффективностью не только благодаря максимальной автоматизации расчетов на базе обширных справочно-энциклопедических информационных массивов, но и благодаря недостижимой ранее наглядности результатов. Широкое использование возможностей компьютерной графики позволяло теперь отвлечься, наконец, от вычислительных сложностей и сосредоточиться на том, чтобы «просто увидеть» решение интересующих проблем, максимально мобилизуя образное мышление и интуитивные способности. Компьютер в таких условиях превращается уже не просто в мощное и эффективное средство моделирования, но и в своего рода «машину времени», способную почти мгновенно перенести исследователя в любую эпоху и любое место.

Новые возможности не заставили долго ждать результатов. Весьма красноречив следующий пример. Уже в 1994 году в списках мировых бестселлеров появляется книга Роберта Г. Бьювэла и Эдриана Г. Джилберта «Тайна Ориона», которая в немецком варианте имела сенсационный подзаголовок: «Через более чем 4000 лет наконец-то решена загадка пирамид» [7] (см. также русский перевод [8]). Авторы, инженер и журналист, основываясь на большом, собранном за десятилетия работы, фактическом материале, пришли к ошеломляющим выводам, «полностью переворачивающим наши представления об истории человечества и египетской цивилизации», как написано в одном из отзывов. Одним из основных «действующих лиц» книги явилась компьютерная программа Skyglobe 3.5,

позволившая достаточно аргументировано обосновать самые смелые из выдвинутых гипотез. Фактически, этот случай можно считать одним из первых прецедентов, когда благодаря добросовестному использованию компьютерного моделирования непрофессионалы смогли настолько основательно обосновать свои довольно радикальные гипотезы, что специалисты (астрономы и египтологи) фактически ответили молчаливым согласием и не нашли сколь-нибудь весомых аргументов против.

При этом следует обратить внимание на то, что упомянутая выше программа Skyglobe, как и многие другие современные программные продукты для выполнения астрономических вычислений, с полным основанием могут быть названы именно **моделирующими**, т.к. в основе их функционирования лежит целостная модель имеющихся на сегодня астрономических знаний, а продуманный графический интерфейс превращает работу с программами в серию модельных экспериментов, за каждым из которых стоит целый комплекс соответствующих вычислений.

Возвращаясь к археомоделированию следует отметить, что роль и место этого направления в системе компьютерных наук во многом определяется концепцией расширенного (обобщенного) кодологического базиса [9]. В рамках этой концепции модели и инструменты, являющиеся предметом изучения археомоделирования, соответствуют монокодовому этапу. Исследование особенностей этого этапа представляется на сегодня актуальным в связи с тем, что особый интерес в настоящее время представляют закономерности перехода от монокодового этапа к современному бинарному (дикодовому) этапу. Понимание этих закономерностей позволит более эффективно реализовать переход к следующему (гиперкодовому) этапу в развитии компьютерных технологий, первые элементы которого начали формироваться уже на протяжении XX века.

Можно констатировать, что первое десятилетие исследований в области археомоделирования явилось достаточно плодотворным. При этом можно достаточно уверенно констатировать, что моделирование в его различных проявлениях играло одну из важнейших ролей в истории человеческого познания. В современную компьютерную эпоху эта роль значительно возросла и, несомненно, будет расти в дальнейшем. На этом фоне вполне закономерным представляется и рост интереса к фактам и историческим закономерностям эволюции средств и методов вычислительного моделирования. Именно этим

определяется в настоящее время интерес к археомоделированию, позволяющему несколько по-новому взглянуть на когнитивную эволюцию человеческого общества и его перспективы.

Современные средства компьютерного моделирования позволяют комплексно реконструировать наиболее интересные объекты археомоделирования. Некоторые действующие модели такого рода располагаются на украинском портале моделирования (simulation.in.ua) [10]. Первоначальной основой данного портала явился раздел археомоделирования, на котором, в частности, представлены модели функциониования Мальтинской пластины и Фестского диска, реализованные магистром Донецкого национального технического университета Татьяной Самойловой (simulation.in.ua/asim1). В дальнейшем планируется расширять состав такого рода моделей по мере углубления и расширения исследований в области археомоделирования.

Литература:

1. Аноприенко А. Я. Археомоделирование. – Донецк: УНИТЕХ, 2007. – 318 с.
2. Апокин И.А., Майстров Е.М. Развитие вычислительных машин. – М.: Наука, 1974. – 399 с.
3. Хокинс Дж., Уайт Дж. Разгадка тайны Стоунхенджа: Пер. с англ. – М.: Мир, 1973. – 256 с.
4. Хокинс Дж. Кроме Стоунхенджа: Пер. с англ. – М.: Мир, 1977. – 268 с.
5. Астрономия древних обществ // Материалы конференции «Астрономия древних цивилизаций» Европейского общества астрономии в культуре (SEAC) в рамках Объединенного Европейского и Национального астрономического съезда. Москва, 23-27 мая 2000 г. – М.: Наука, 2002. – 334 с.
6. Монтенбрук О., Пфлегер Т. Астрономия на персональном компьютере. – СПб.: Питер, 2002. 320 с.
7. Bauval R., Gilbert A. Das Geheimnis des Orion. – Műnchen, List Verlag, 1994. – 384 с.
8. Бьювэл Р., Джилберт Э. Секреты пирамид. Созвездие Ориона и фараоны Египта. – М.: Вече, 1997. – 368 с.
9. Аноприенко А.Я. Расширенный кодо-логический базис компьютерного моделирования / В кн. «Информатика, кибернетика и вычислительная техника (ИКВТ-97). Сборник научных трудов ДонГТУ». Выпуск 1. Донецк, ДонГТУ, 1997, с. 59-64.
10. Аноприенко А.Я., Башков Е.А., Самойлова Т.А. Портал компьютерного моделирования: цели, задачи и особенности организации // Материалы первой международной научно-технической конференции «Моделирование и компьютерная графика», г. Донецк, 04-07 октября 2005 г., Донецк, 2005. С. 16-20.

Получено 12.06.07

УДК 681.323

ИНТЕГРАЛЬНЫЙ МЕТОД ВОССТАНОВЛЕНИЯ РЕЛЬЕФА ДОННОЙ ПОВЕРХНОСТИ ПО КАРТОГРАФИЧЕСКОЙ ИНФОРМАЦИИ

А.О.Пьявченко
Технологический институт Южного федерального университета,
г.Таганрог

В статье рассматривается разработанный автором интегральный метод восстановления рельефа донной поверхности в трехмерный вид по неполному набору значений глубин заданного района плавания.

Предлагаемый к рассмотрению метод восстановления рельефа донной поверхности в трехмерный вид по картографической информации относится к классу волновых алгоритмов и основывается на анализе корреляционных связей между соседними точками растра. При этом метод базируется на известном рекурсивном методе пространственной фильтрации – методе пространственной свертки.

Метод позволяет проводить обработку двумерных карт поверхности морского дна, представленных значениями промерных глубин или линиями изобат. При этом анализируемая поверхность должна быть представлена с привязкой к узлам растра, причем исходные данные о поверхности (опорные точки) могут быть распределены по растру случайно, без всякой закономерности. Остальные точки обозначаются как неопределенные, то есть как раз такие, значения в которых необходимо вычислить в конце работы алгоритма реализации метода.

Шагу растровой сетки соответствует некоторое заранее определенное значение расстояния, отмеренное на плоскости, например, промерной карты по долготе или по широте между двумя точками донной поверхности. Следовательно, величиной шага можно регулировать исходный масштаб представления донной поверхности.

В ходе проводимых вычислений опорные точки не меняют до конца работы алгоритма своих координат и своего определенного заранее значения. Они также являются источниками волны, распространяющейся в четырех направлениях от анализируемой в данный момент времени опорной точки и инициируют вычислительный процесс по следующей схеме: вниз, вверх, влево и

вправо, что составляет одну итерацию в распространении волны. Состояние каждой точки на каждом направлении распространения волны просчитывается независимо от расчета состояния остальных точек. Причем, за одну четверть волны выполняется расчет всех точек растра, не являющихся опорными. Это происходит каждый раз по мере формирования итерации волны и ее распространения во всех четырех направлениях. В итоге при растре 512x512 точек для худшего варианта представления исходных данных, когда определены только две опорные точки в противоположных углах сетки, получим искомый результат не более чем за 724 полные итерации распространения волны. Значение времени вычислений прямо пропорционально размерности самого растра и обратно пропорционально числу изначально известных узловых опорных точек сетки.

Очевидно, что восстановленная таким образом донная поверхность имеет дискретное представление. При этом, чем больше шаг растра, тем хуже точность восстановления поверхности из-за увеличения вероятности непопадания исходных координат в узлы сетки.

Рассмотрим предлагаемый метод более подробно на примере распространения четверть волны вниз (см. рисунок 1).

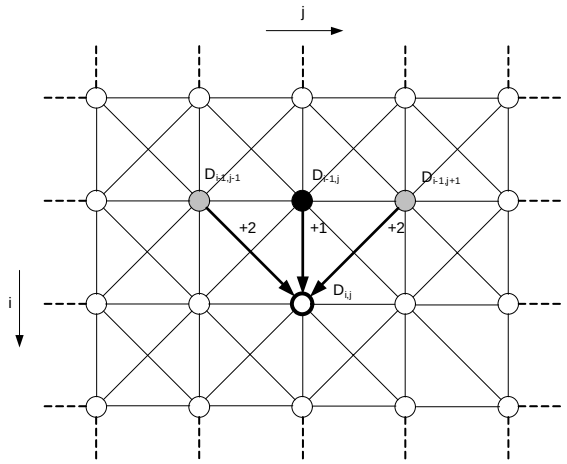


Рис. 1 - Один шаг распространения волны

Каждая узловая вершина сетки характеризуется двумя основными параметрами - расстоянием до ближайшей опорной точки $L_{i,j}$ и текущей глубиной $D_{i,j}$ в точке. На начальном этапе все точки растра, кроме опорных, имеют нулевой код глубины, который принимается за код неопределенности. Код дистанции также принимается равным нулю для всех точек растра.

В процессе расчетов дистанции до ближайшей опорной точки расстояние по вертикали и горизонтали между соседними узлами

сетки условно принято равным 1, а расстояние по диагоналям сетки - равным 2. Сделано это для сокращения используемой при расчетах разрядной сетки целочисленного вычислителя. Так, существуют модификации алгоритма, где расстояние между соседними узлами сетки равно 100 и 141, 1000 и 1414, соответственно, что, в свою очередь, позволяет увеличить точность производимых расчетов при соответствующем увеличении разрядной сетки вычислителя.

Анализируемая вершина (i,j) принимает “волну” значений глубин от соседних сверху вершин ($i-1,j$), (i,j), ($i+1,j$).

Пусть хотя бы одна из указанных вершин имеет ненулевое значение глубины, а текущая расчетная вершина не является опорной. Напомним, что опорные точки в основной части алгоритма не пересчитываются.

Тогда текущее «кажущееся» значение глубины определяется по следующей формуле (1)

$$\tilde{D}_{i,j} = \frac{\sum C_i^p D_{i\,\text{пред}}}{\sum C_{i\,\text{пред}}^p} = \frac{C_{i,j} \cdot D_{i-1,j} + C_{i,j-1} \cdot D_{i-1,j-1} + C_{i,j+1} \cdot D_{i-1,j+1}}{C_{i,j} + C_{i,j-1} + C_{i,j+1}}, \qquad (1)$$

где $C_{i,j-1}$,$C_{i,j}$, $C_{i,j+1}$ - весовые коэффициенты, $D_{i-1,j-1}$,$D_{i-1,j}$, $D_{i-1,j+1}$- значения глубин в соседних точках растра на текущий момент времени, *не равные 0*.

Как уже отмечалось, данная процедура является разновидностью дискретной свертки, используемой в методах цифровой пространственной фильтрации. Основой алгоритмов фильтрации данного класса является одномерная или двумерная последовательность коэффициентов, называемая ядром свертки. Применяемый оператор строится на основании значений соседних точек, а также на последовательности особого вида констант или коэффициентов. В случае двумерных сверток рассматриваемый сигнал проходит обработку окном конечного размера и определенной формы (например, весь растр или меньше). Результатом работы является взвешенная сумма значений окрестных точек внутри рассматриваемого окна, где весом как раз и являются отдельные коэффициенты $C_{i,j}$.

В нашем методе каждый коэффициент $C_{i,j}$ рассчитывается как функция от значения кратчайшего расстояния L до ближайшей опорной точки от анализируемой с учетом принятых выше значений расстояний dL между соседними узлами сетки. Следует отметить, что в общем случае на выбор коэффициентов также оказывают влияние предельные значения величин перепадов глубин, изначально представленных в растре, а также требуемые точность и скорость

вычислений, однако этот вопрос не является темой обсуждения настоящей статьи.

В упрощенном варианте коэффициенты $C_{i,j}$ могут быть рассчитаны как функция, обратная дистанции по каждой дуге, то есть

$C_{i,j+1} = LL_i - L^*_{i-1,j+1}+1$, $C_{i,j} = LL_i - L^*_{i-1,j}+1$, $C_{i,j-1} = LL_i - L^*_{i-1,j-1}+1$

где $L^*_{i-1,j-1}$, $L^*_{i-1,j}$, $L^*_{i-1,j+1}$ – значение кратчайшего расстояния до ближайшей опорной точки от точки *(i,j)*, LL_i – сумма значений кратчайших расстояний, рассчитанная по формуле:

$$LL_i = \sum_{k=-1}^{1} L^*_{i-1,j+k},$$

где $L^*_{i-1,j-1} = L_{i-1,j-1}+2$, $L^*_{i-1,j} = L_{i-1,j}+1$, $L^*_{i-1,j+1} = L_{i-1,j+1}+2$.

«Кажущееся» значение кода дистанции в точке *(i,j)* определяется как $\tilde{L}_{i,j} = min(L^*_{i-1,j-1}, L^*_{i-1,j}, L^*_{i-1,j+1})$.

После вычисления по формуле (1) «кажущегося» значения текущей глубины $\tilde{D}_{i,j}$ производится расчет истинной глубины обрабатываемой точки с округлением к ближайшему:

$$D_{i,j} = \frac{D_{i,j} + \tilde{D}_{i,j}}{2},$$

где $D_{i,j}$ – ненулевое значение глубины, полученное в узле на предыдущем шаге итерации (для правой части формулы).

Истинное значение кода дистанции определяется как результат нахождения минимума между его «кажущимся» и старым ненулевыми значениями для текущей точки растра.

Если же первоначальное значение глубины $D_{i,j}$ все же равно 0, то его новое значение принимается равным $\tilde{D}_{i,j}$, а код дистанции его кажущемуся значению.

В соответствии с представленным выше методом можно выделить следующий алгоритм определения глубины в отдельно взятой точке растра при одном проходе четверти волны сверху вниз.

1. Установка начальных значений $D_{i,j}$, $L_{i,j}$ для текущей точки растра. При нулевом первоначальном значении глубины точка неопределенна, дистанция равна нулю, в противном случае это опорная точка с нулевым кодом дистанции.

Замечание 1. *Все вычисления для опорной точки не производятся. Кроме того, рассматриваемая в статье версия алгоритма реализации метода не учитывает код береговой линии.*

2. Расчет кодов дистанции $L^*_{i-1,j-1}$, $L^*_{i-1,j}$, $L^*_{i-1,j+1}$, а также их суммарного значения LL_i.

3. Подсчет значений весовых коэффициентов $C_{i,j-1}$, $C_{i,j}$, $C_{i,j+1}$.

4. Определение кажущихся значений глубины $\tilde{D}_{i,j}$ и дистанции $\tilde{L}_{i,j}$.
5. Расчет истинных значений глубины и дистанции анализируемой точки $D_{i,j}$ с учетом соответствующих предыдущих значений.
6. Повторение п.2-5 для следующего по схеме вычислений прохода (вверх, влево, вправо) с учетом замечания 1.
7. Повторение п.2-6 при распространении полной волны до границ растра.

На рисунках 2.а,..,2.г представлены результаты работы экспериментальной программы, реализующей вычисления согласно рассмотренному выше методу. Здесь показаны этапы исходного, начального представления поверхности в 2D-виде (рис. 2.а) для растра 64x64 и соответствующий ему трехмерный вид (рис. 2.б), затем результат вычисления первой волны (рис. 2.в) и окончательный вид полученной трехмерной поверхности (рис. 2.г).

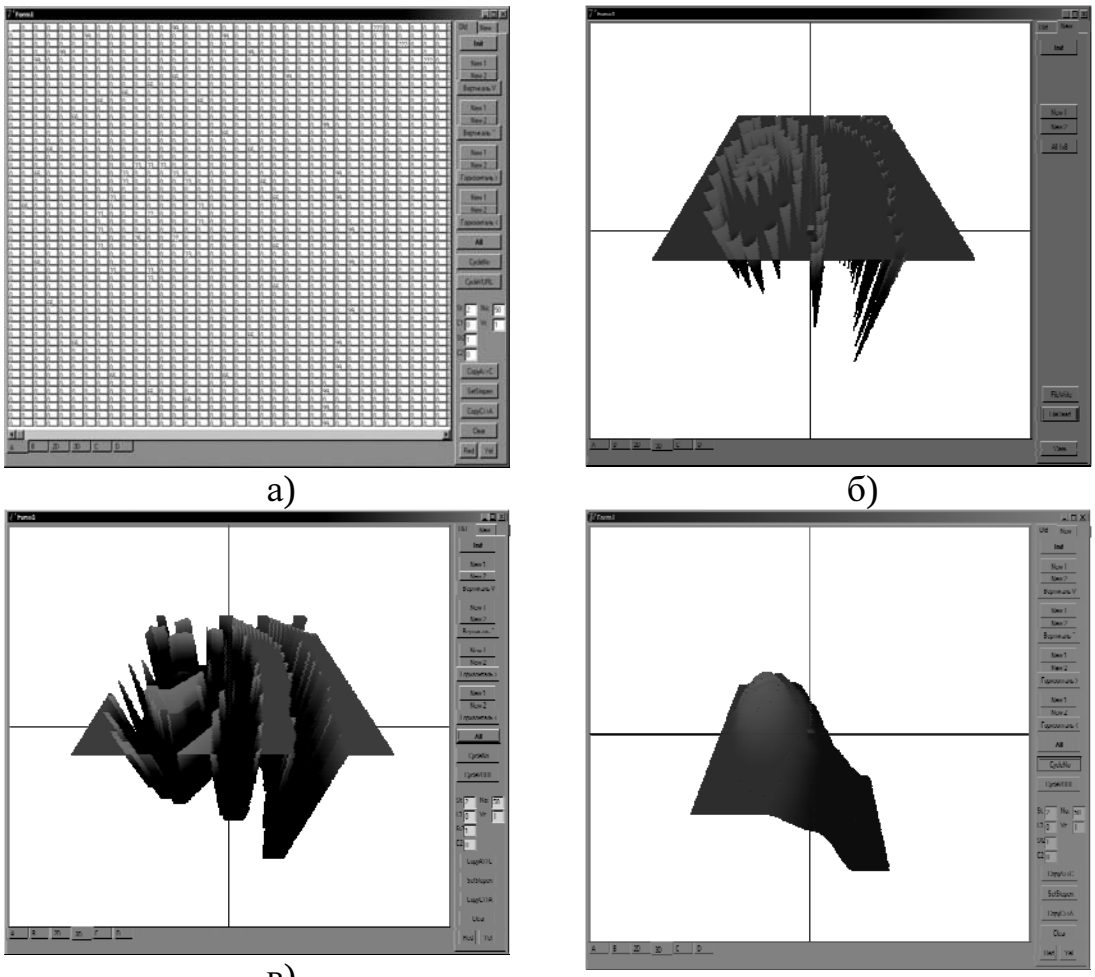


Рис. 2 - Результаты работы программы при расчете растра 64x64

В заключении необходимо отметить, что предложенный метод восстановления рельефа донной поверхности характеризуется естественным параллелизмом, векторизуемостью и масштабируемостью основных операций: алгоритм вычислений в каждом узле растра одинаков как для каждой четверти итерации, так и для всех итераций в целом. Это позволяет с высокой эффективностью выполнять решение задачи восстановления трехмерного рельефа на параллельных процессорах с SIMD архитектурой, например, на мультипроцессорах с конфигурацией связи типа n-куб, систолическая решетка и т.п.

В настоящее время на хоздоговорной основе коллективом кафедры вычислительной техники ТТИ ЮФУ проводятся работы по созданию макета параллельного вычислителя в базисе ПЛИС. Полученные в ходе выполнения работ результаты не только доказывают аппаратную реализуемость и эффективность метода, но и позволяют сделать необходимые оценки его характеристик, а также предложить различные его модификации.

Получено 01.06.07

UDK 004.4

G95 FORTRAN BINDINGS FOR OPENGL

Mykola Yas'ko
Dnipropetrovs'k National University
Computer Technology Department

Most mathematical software for scientific computing is written in Fortran, and most scientific computing applications require 2D/3D graphics for visualization. It is therefore important to provide a good Fortran interface to OpenGL [1] and related libraries [2]. Now there is a perfect G95 Fortran the compiler for many Unix, Windows and Mac operation systems. This compiler is developed according to Unix traditions and exists everywhere where is gcc.

The official bindings are defined by William Michall in [3], and were approved by the OpenGL Architecture Review Board in February 1998. The bindings are equally applicable to Fortran 95 Standard. By using the new features of Fortran 2003 is possible to define bindings for OpenGL that do not depend on any extensions to the standard and provide access to the full functionality of OpenGL. The G95 compiler gives new remarkable opportunities for direct linkage of dll/so libraries. They can also increase the capability of robustness and portability in the user application code, and increase the similarity between the Fortran and C interfaces.

This report describes the Fortran g95 bidings for OpenGL. The major differences between the Fortran 90 and Fortran 2003 bindings are:

- the new approach does not require wrapper functions and additional libraries;
- the standard types of a Fortran 2003 are used only;
- procedure names are the same as in C language.

The Fortran 90 interface to OpenGL is accessed through module OpenGL. The module provides access to kind type parameters, named constants and procedure interfaces. Some details of the interfaces are given below.

Character strings should come to an end in zero byte as in C. In particually **glutInit** command you should use in the next way:

```
call glutinit(1,loc('thePgm'//char(0)))
```

For functions, which return a character string, you should use pointers. See example:

```
character(1200), pointer :: str
..............................
str=>glGetString(GL_VERSION)
```

```
        print *, 'OpenGL version is ',str(1:
min(len(str),index(str,char(0))))
```

In some commands you have to pass pointers of the different types. Use G95 fortran function **loc** to get address, for example

```
    call glVertexPointer( 2, GL_FLOAT, 0, loc(P) )
```

where **P** is a real array of vertices with coordinates x,y.

In GLU some functions return the pointer on structure. I remade it in **integer(kind=C_INTPTR_T)**. For example

```
        integer(C_INTPTR_T) :: nurb
        nurb = gluNewNurbsRenderer()
```

If the callback subroutine has arguments, you have to pass it by value (as in C). Example:

```
    subroutine motion(x, y) bind(c)
    integer(GLint), intent(in), value :: x, y
```

This approach works good for Unix 32/64 platforms, but Windows platform has some problems. First there are some GLUT versions. The first version of GLUT was ported to Win32 by Nate Robins in 2001. At present time we have a OpenGLUT project, which differs from the old version. In addition to these circumstances the OpenGL procedures are described as stdcall, while GLUT CALLBACKS have to be cdecl. The G95 fortran compiler uses cdecl procedure calls by default. It can be changed by option –mrtd. Some updating of a file opengl.mod in this case is required which is carried out specially written script. In this case described interfaces of functions suit all versions of GLUT dll libraries.

References

1. Mark Segal and Kurt Akeley, The OpenGL Graphics System: A Specification (Version 1.1), http://www.sgi.com/Technology/openGL/glspec1.1/glspec.html
2. Smith K. P., Frazier C., The OpenGL Graphics System Utility Library, http://reality.sgi.com/mjkasd/GLUspec.ps.gz
3. Mitchell, W. F., The Fortran 90 Bindings for OpenGL, ACM Fortran Forum, **18** (1999), pp.5-13.

25.05.2007

УДК.744 (075.8)

МОДЕЛИРОВАНИЕ САМОЛЕТА-АМФИБИИ Бе-200

И.Б. Аббасов
Таганрогский технологический институт
Южного федерального университета

Работа посвящена моделированию самолета-амфибии с использованием графической системы AutoCAD. Рассматриваются вопросы моделирования конструктивных частей самолета-амфибии. Представлены варианты тонирования и визуализации разрабатываемой модели.

В данной работе рассматриваются вопросы трехмерного моделирования самолета-амфибии Бе-200, разработанной Таганрогским авиационным научно-техническим комплексом (ТАНТК) им. Бериева. Для моделирования используется графическая система AutoCAD.

В настоящее время существует достаточное количество графических систем трехмерного моделирования и САПР: Mechanical Desktop, CATIA, Maya, 3 ds max, Light Wave, Solid Works, T-Flex, КОМПАС и др. Графическая система AutoCAD позволяет создавать и редактировать чертежи, также в ней существует возможность трехмерного моделирования и создания реалистичных изображений.

Трехмерные модели самолета-амфибии можно создавать методами твердотельного моделирования. Для этого потребуется некоторый исходный материал в виде габаритных чертежей и фотографий моделируемого объекта (рис.1).

Плоские сечения можно строить на основе сплайна, и необходимо распределить их по длине лодки. Аналогичным способом строится крыло и хвостовая часть. Двигатели и стабилизаторы созданы методом вращения из полилинии. На рис.2 представлен результат моделирования конструктивных частей самолета-амфибии на основе исходных плоских форм.

Необходимо отметить, что проектирование самолета началось в 1990 году на основе самолета-амфибии А-40 "Альбатрос", от которого Бе-200 отличается уменьшенными размерами. Впервые проект был представлен в 1991 г. на авиационно-космической выставке в Париже. Для осуществления программы самолета Бе-200 было образовано совместное предприятие ЗАО "БЕТА ИР", в состав которого вошли Иркутское авиационно-производственное объединение, ОАО "ТАНТК

им. Г.М. Бериева" и швейцарская фирма "ИЛТА Трейд Файненс С.А.". Серийное производство Бе-200 начато в г. Иркутске.

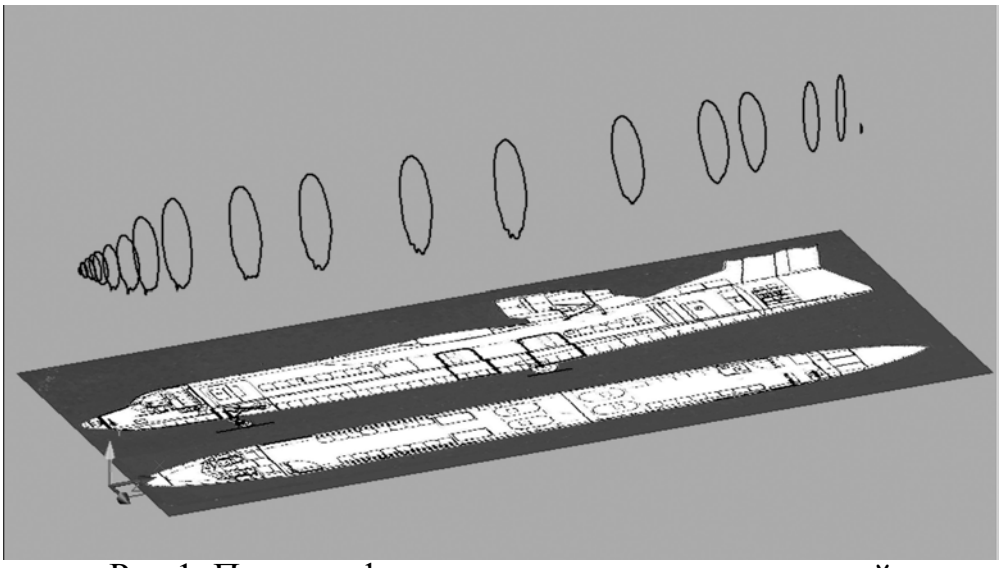


Рис.1. Плоские формы лодки на основе чертежей

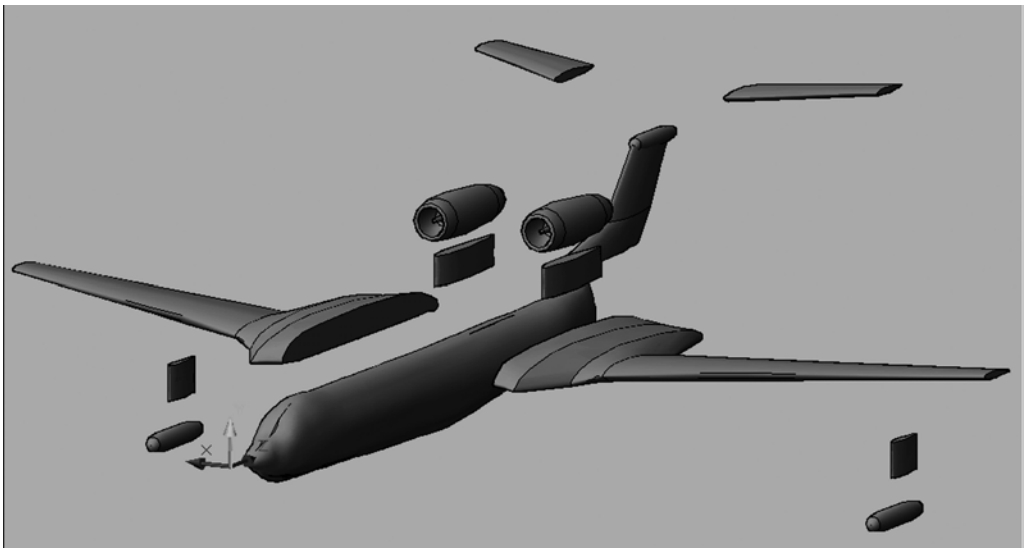


Рис.2. Конструктивные части самолета-амфибии

Разработаны следующие модификации самолета-амфибии Бе-200: пожарный, пассажирский, грузовой. Пожарный вариант оборудован водяными баками общей емкостью 12 м^3, заполнение которых производится в режиме глиссирования самолета над водной поверхностью с помощью водозаборников, расположенных за реданом лодки.

В нижней части баки имеют автоматически открываемые люки для сброса воды. За одну заправку топливом при взлетной массе 37,2 т

самолет способен доставить к очагу пожара до 270 т воды при удалении очага пожара от аэродрома 100 км, а от водоема 10 км.

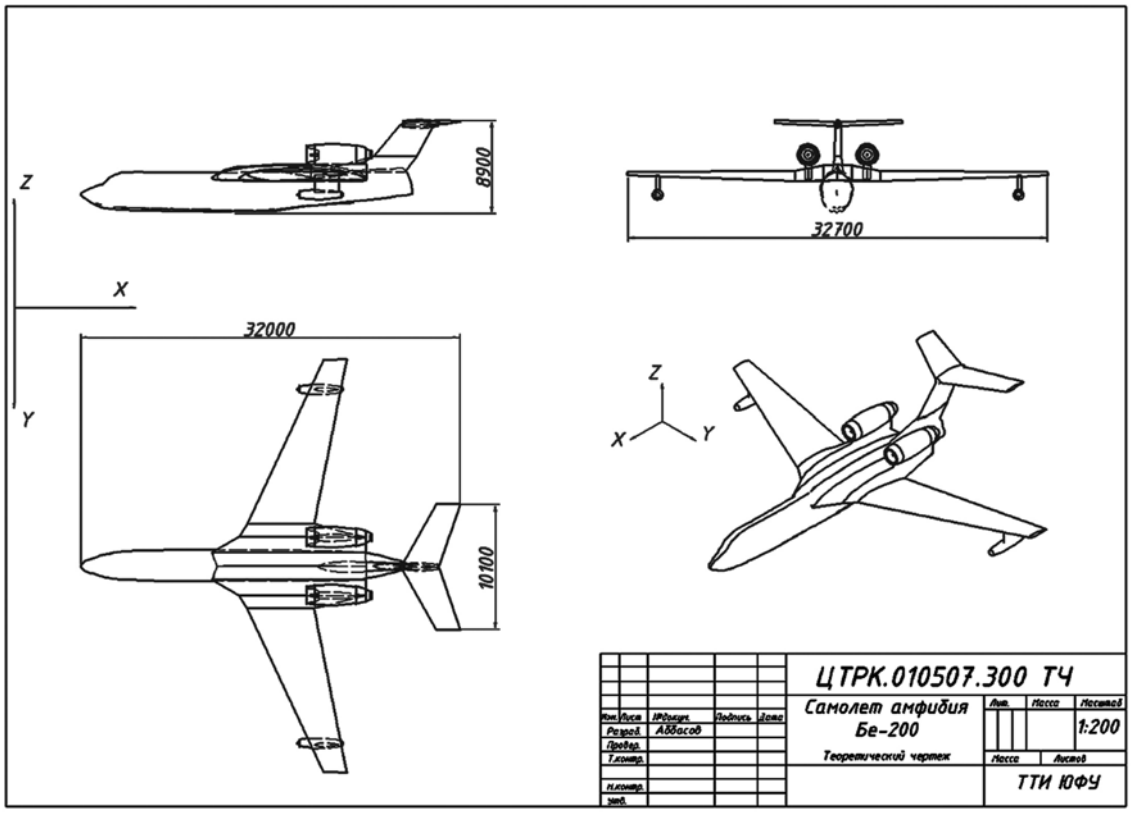


Рис.3. Теоретический чертеж на основе параметрической модели

Размах крыла самолета-амфибии Бе-200 составляет 32,78 м, длина 32,05 м, высота 8,9 м, максимальный диаметр фюзеляжа 2,86 м, экипаж 2 человека. Самолет-амфибия Бе-200 оснащен двигателями ТРДД Д-436ТП Запорожского МКБ "Прогресс" (существует модификация с двигателями BR-715 фирмы Rolls-Royce).

На рис.3. представлен теоретический чертеж самолета-амфибии, который был автоматически построен на основе созданной параметрической модели. Для разработки данного конструкторского документа использовалась графическая система Mechanical Desktop, являющейся трехмерной надстройкой системы AutoCAD.

Максимальная взлетная масса самолета-амфибии составляет 37200 кг, после забора воды в режиме глиссирования 43000 кг. Максимальная крейсерская скорость на высоте 8000 м составляет 710 км/ч, практический потолок 11000 м, дистанция взлета с воды 1000 м, дистанция посадки на воду 1300 м, дальность полета с нагрузкой 6500 кг - 1850 км.

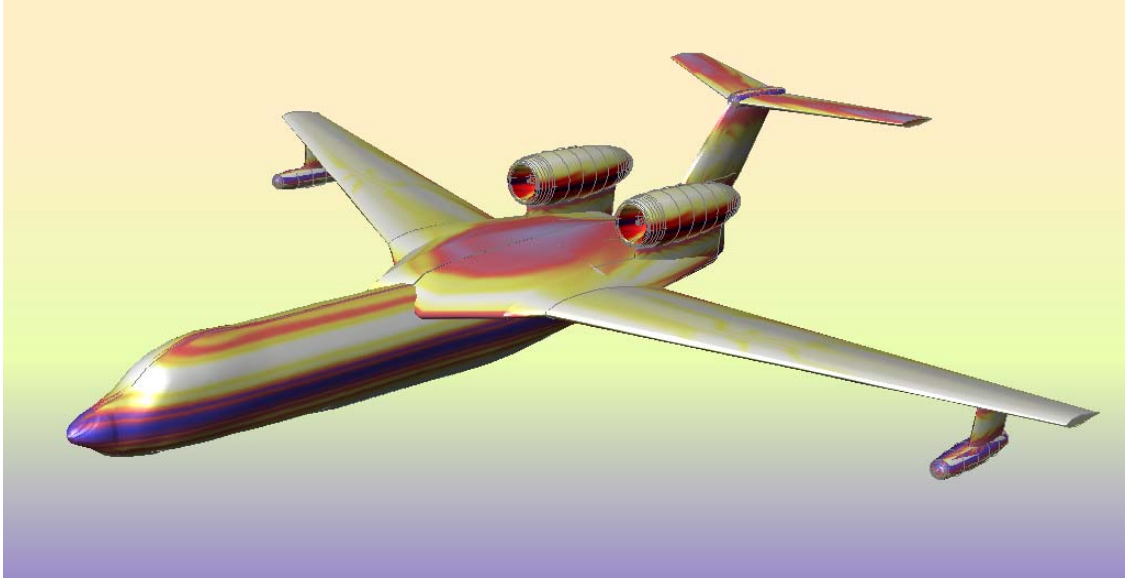


Рис.4. Твердотельная тонированная модель самолета-амфибии

В версии графической системы AutoCAD 2007 имеется удобный специальный интерфейс для трехмерного моделирования. В режиме 3D Modeling (3М моделирование) классическое рабочее окно программы меняется, появляется панель Dashboard (Инструментальная панель), которая состоит из панелей управления:

- 2D Make (2М построения), по умолчанию она скрыта;
- 3D Make (3М построения);
- 3D Navigate (3М навигация);
- Visual Styles (Стили визуализации);
- Light (Освещение;
- Materials (Материалы);
- Render (Тонирование).

Для изменения внешнего вида поверхностных и твердотельных моделей используется панель Visual Styles (Визуальные стили). Опции Visual Styles (Визуальные стили) позволяют выбрать режимы просмотра трехмерных объектов: двухмерный каркас, трехмерный каркас, реалистичный, концептуальный.

На рис.4 представлена твердотельная модель самолета-амфибии в режиме отображения Realistic (Реалистичный).

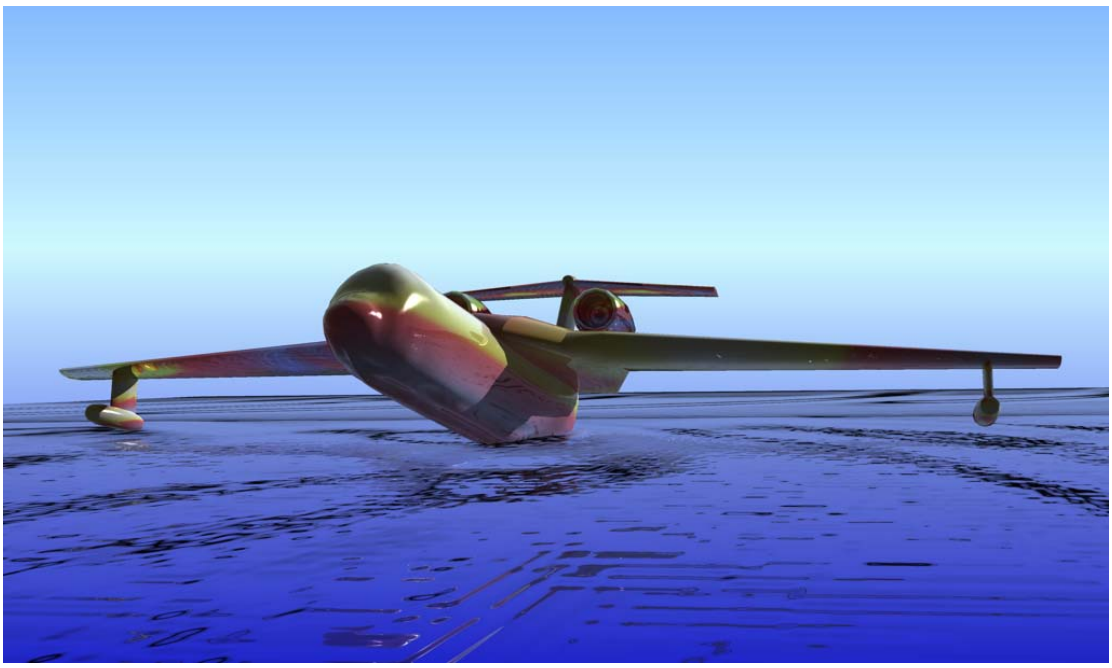


Рис.5. Визуализация сцены взлета с воды самолета-амфибии

На рис.5 представлена визуализация сцены взлета самолета-амфибии. Следует отметить, что самолет-амфибию можно смоделировать и в других трехмерных системах, однако, данная графическая система позволяет изготовить конструкторскую документацию моделируемого объекта.

Литература

1. Хейфец А. Инженерная компьютерная графика AutoCAD. Учебное пособие. СПб.: BHV, 2005. 336 с.
2. Аббасов И.Б. Чертежи в графической среде AutoCAD. Учебное пособие. Таганрог. Изд-во ТРТУ. 2002. 82 с.
3. Красильникова Г., Самсонов В., Тарелкин С. Автоматизация инженерно-графических работ. СПб. Изд-во «Питер», 2000. 256с.
4. Чекмарев А.А., Верховский А.В., Пузиков А.А. Начертательная геометрия. Инженерная и машинная графика /Под ред. А.А. Чекмарева. М.: Высш. шк., 1999. 154 с.
5. Ткачев Д.А. AutoCAD 2007: СПб.: Питер; Киев, BHV, 2007. 464 с.
6. Официальный сайт www.aviaport.ru

Получено 18.05.2007, ТТИ ЮФУ, г. Таганрог

© И.Б. Аббасов, 2007

УДК 004.932

АЛГОРИТМ КОНТЕКСТНОГО ПОИСКА ОБЪЕКТОВ ПО ФОРМЕ

С.Г. Антощук, Г.Ю. Щербакова, Н.Ю. Хитрук, О.Л. Дрегваль
Одесский национальный политехнический университет

З метою удосконалення методів та алгоритмів контекстного пошуку зображень за їхнім описом з використанням параметрів форми запропонована ієрархічна сигнально-семантична модель об'єктів розпізнавання у вигляді набору контурних препаратів.

Развитие информационных сетей, увеличение количества изображений в базах данных, и количества пользователей, которым необходимо проводить поиск изображений в соответствии со своими требованиями, привело к развитию методов и алгоритмов контекстного поиска изображений [1]. При проведении контекстного поиска используют следующие признаки определения сходства и описания изображений: параметры формы, цвет (корелограммы, цветовые моменты и множества цветов в различных цветовых пространствах), текстурные признаки. Для определения формы объекта необходимо иметь его контурное описание, для получения которого проводять следующие операции: выделение контура, скелетизацию и прослеживание контуров.

Один из современных подходов к обработке и распознаванию изображений – это иерархическое моделирование, при котором изображение представляется в виде последовательности матриц с различным уровнем детальности, что соответствует пирамидальной модели. Существующие методы пирамидального представления информации используют низкочастотную фильтрацию, которая размывает перепады интенсивности и затрудняет выделение контуров [2]. В статье разработана иерархическая модель представления объектов распознавания в виде набора (пирамиды) контурных препаратов, полученных с разным разрешением. Такое представление повысит эффективность работы широкого класса поисковых алгоритмов.

Большинство методов выделения контуров можно описать схемой, состоящей из двух этапов:

– на первом производится подчеркивание контуров;

– на втором выделяются контурные точки.

Методы выделения контуров можно разделить на две большие группы: дифференциальные и корреляционно-экстремальные [3]. При

выделении контуров дифференциальными методами перепады интенсивности усиливаются, и с помощью порогового устройства выделяется область контура. Дифференциальные методы просты и обладают высоким быстродействием, однако имеют низкую помехоустойчивость. В корреляционных методах реальный перепад интенсивности сравнивается по форме с набором моделей-эталонов. В качестве меры близости используется максимум корреляционной функции или минимум среднеквадратической ошибки. После этого в контурный препарат изображения вносится фрагмент, соответствующий принятому решению. Корреляционные методы обладают высокой помехоустойчивостью, однако имеют и существенные недостатки, связанные с выбором моделей перепадов интенсивности. При этом ни один из существующих методов выделения контуров не позволяет регулировать детальность полученных контурных препаратов.

Поэтому представляется целесообразным для иерархического представления изображений использовать вейвлет-преобразование (имеющее свойства пространственно-частотной локализации и регулируемой детальности) с такой базисной функцией, которая позволила бы эффективно подчеркивать контуры [4,5] — гиперболическое вейвлет-преобразование (ГВП).

ГВП определяется путем свертки

$$W(s,x_0)=\frac{1}{\sqrt{s}}\int_{-\infty}^{+\infty} f(x)\psi^*(\frac{x-x_0}{s})dx \ ,$$

где $f(x)$ – преобразуемая (анализируемая) функция; $\psi_{s,x_0}(x)$ – двухпараметрическая базисная функция, получающаяся из материнского вейвлета $\psi_0(x)$ при масштабировании с масштабным множителем $s \in R^+$ и сдвиге с параметром $x_0 \in R$, ψ^* – комплексно-сопряженная функция по отношению к ψ (s отвечает за ширину вейвлета, x_0– определяет положение вейвлета на оси x). Множитель $\frac{1}{\sqrt{s}}$ вводится для того, чтобы все вейвлетные функции ψ_{s,x_0} имели постоянную норму в пространстве $L^2(R)$.

При ГВП базисная функция определяется $\psi_0(x)=\frac{1}{\pi\alpha x}G(x)$, где $G(-x) = G(x)$, $G(x)=1(x-\varepsilon)-1(x-\gamma)$; 1(t) – единичная функция Хевисайда; $\alpha>0$ – масштабирующий коэффициент для получения базисных вейвлетных функций.

При обработке изображений в пространстве ГВП энергия изображений концентрируется вблизи наиболее информативной части – перепадов интенсивности, что повышает вероятность правильного

распознавания — семантическую меру информации. С другой стороны, вейвлет-преобразование представимо в виде свертки исходного изображения с вейвлет-функциями разного масштаба, что характерно для сигнального представления. Поэтому представленная в данной статье информационная модель обработки изображений названа иерархической сигнально-семантической (рис. 1). Обработка изображений при этом предусматривает следующие этапы:

— на вход модели поступает изображение в матричном виде;

— матрица интенсивностей подвергается ГВП с разным уровнем разрешения. Масштабы преобразования выбираются с помощью априорной информации об изображениях. Результатом является упорядоченная последовательность матриц с разным уровнем детальности;

— на выбранных уровнях в блоке выделения области контура (БВОК) производится выделение контуров — на выходе модели формируется пирамидальное представление контурных препаратов. Приведенные выше свойства ГВП [4,5] создают основу для синтеза методов иерархического выделения границ на полутоновых изображениях в пространстве ГВП. Целесообразно создавать как дифференциальные методы для обработки слабо зашумленных изображений, так и корреляционно-экстремальные для изображений с низким отношением сигнал/шум.

Для реализации сигнально-семантической модели представления изображений предлагается следующая методика:

— изображение подвергается свертке с гиперболической вейвлет-функцией разных масштабов от наибольшего к наименьшему;

— на контрастированном изображении выделяется область контура;

— при необходимости проводится процедура «скелетизации» (утончения) области контура;

— на изображениях наибольшего масштаба находится внешний контур объекта исследования;

— обработка следующего иерархического уровня проходит внутри контура изображения объекта;

— обработка проводится по строкам и по столбцам. Результат объединятся по схеме логического сложения ИЛИ.

На базе полученного подхода проводилось прослеживание контуров реперных знаков (РЗ) на фотошаблонах (ФШ) интегральных микросхем, рассчитывались признаки их формы (топологические дескрипторы, дескрипторы области, геометрические моменты-признаки с использованием информации о характерных точках).

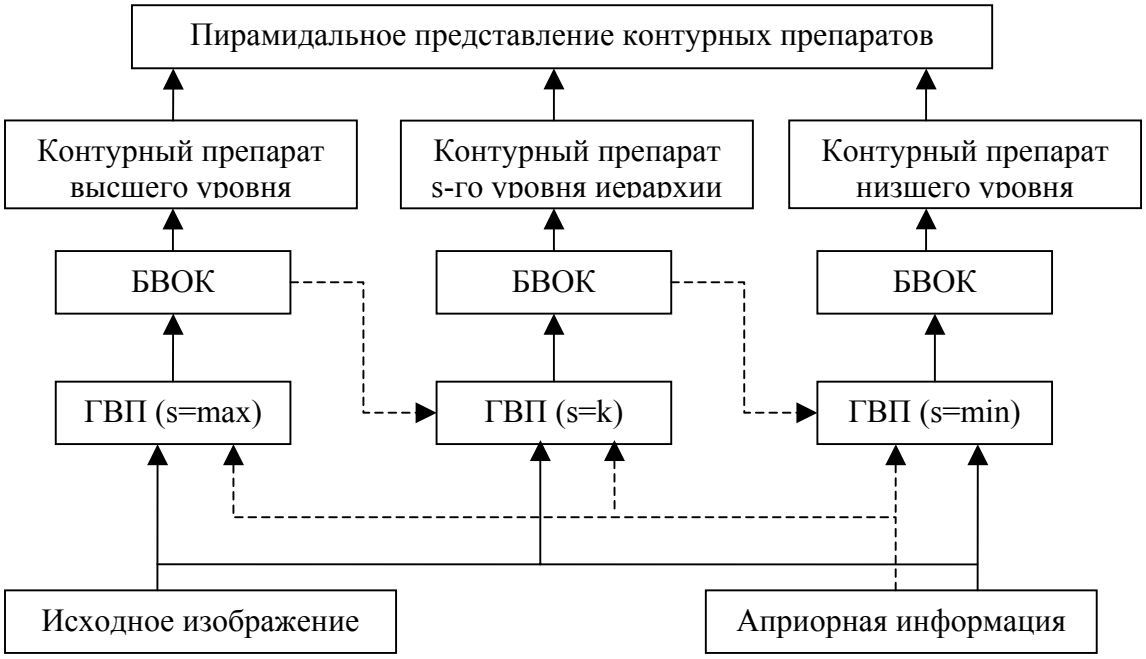


Рис.1. Сигнально-семантическая модель представления изображений

При формировании ансамбля признаков формы значения ГМП и компактности РЗ для ортогонализации подвергаются дискретному косинусному преобразованию. Для повышения производительности метода и сокращения размерности признакового пространства был проведен отбор признаков с помощью оценки их информативности по Шеннону. Далее после классификации принимались решения про наличие либо отсутствие объектов заданной для поиска формы. Компьютерное моделирование на тестовых изображениях показало хорошую производительность предложенного алгоритма для контекстного поиска объектов по форме.

Литература

1. Н.С. Байгарова, Ю.А. Бухштаб, А.А. Воробьев, А.А. Горный Организация управления базами визуальных данных Препринт Института прикладной математики им. М.В. Келдыша РАН, 2000, N 6

2. Методы компьютерной обработки изображений / под ред. Сойфера В.А. — 2-е изд., испр. — М.: ФИЗМАТЛИТ, 2003. — 784 с.

3. Крылов В.Н., Максимов М.В. Вторичные преобразователи сигналов изображений. - Одесса, Астропринт, 1997.

4. Antoshchuk S.G., Krylov V.N. Hyperbolic wavelet domain image processing — //Сучасні проблеми радіоелектроніки, телекомунікацій, комп'ютерної інженерії. Матеріали міжнародної конференції. Вид-во нац. ун-ту "Львівська політехніка". – Львів, 2004. — С.219-220.

5. Антощук С.Г. Реализация вейвлетного преобразования при структурном анализе изображений // Електромашинобудування та електрообладнання. - 2004 - Вип.62 - С. 153-157

Дата поступления в редакцию 25.05.07

УДК 004.92

ПОСТРОЕНИЕ ЛОКАЛЬНЫХ ИЗОПОВЕРХНОСТЕЙ СКАЛЯРНЫХ ПОЛЕЙ С ПОМОЩЬЮ АЛГОРИТМА НЕНАПРАВЛЕННОЙ МАРКИРОВКИ

Белоус Н.В., Кобзарь Г.А.

Харьковский национальный университет радиоэлектроники

Пропонується використання алгоритму ненаправленого маркірування для ефективного побудування локальних ізоповерхонь скалярних полів, що підвищує швидкість побудови відносно найбільш ефективного алгоритму на основі РОТ, та забезпечую часткову побудову ізоповерхонь та також облік внутрішнього об'єму.

Введение. Построение изоповерхностей - одно из важнейших средств визуализации многомерных скалярных полей. Процесс построения направлен на выделение контура постоянной скалярной величины. Таким образом, обеспечивается эффективное выявление внутренних структур поля, таких как границы различных тканей в томографии, ударной волны в экспериментах с жидкостями и изменения трехмерных контуров (поверхностей) в процессе физических симуляций.

Со времени публикации алгоритма бегущих кубов [1] было предложено множество способов построения изоповерхностей, поскольку наборы точек таких поверхностей являются входными данными для этого алгоритма. Результат работы алгоритма - трехмерные полигональные модели, т.е. представление изоповерхностей в форме, позволяющей наиболее эффективно визуализировать внутренние структуры поля.

Основной задачей подготовки к работе алгоритма является выделение набора активных ячеек (параллелепипеды, каждый из которых состоит из восьми ограничивающих вершин дискретного скалярного пространства), которые пересекаются изоповерхностью. Последние исследования направлены на уменьшения объема данных и вычислений, необходимых для построения изоповерхностей, поскольку объемы входных данных иногда не позволяют разместить все значения скалярных полей в памяти вычислительных машин.

Предложены методы, более чувствительные к размеру поверхности, чем к объему полного набора данных [2]. Более прогрессивные методы позволяют восстанавливать только необходимые участки изоповерхности [3,4]. Наиболее эффективный метод основан на применении Устойчивого октантного дерева (РОТ), которое позволяет очень эффективно выделять активные ячейки. В самом худшем случае вычислительные затраты такого алгоритма будут O(logN+K) (где N – количество всех ячеек, а K – количество активных ячеек), а объем необходимой памяти - O(N).

Все эти методы рассчитаны на выделение всех поверхностей, соответствующих заданному изоуровню и дальнейшее повышение эффективности ограничивается самой постановкой задачи. Однако в медицинских исследованиях чаще всего необходимо строить не все изоповерхности трехмерных данных томографии, а лишь поверхности, соответствующие исследуемому органу или части тела. Подобные задачи возникают и в других сферах, где необходимы локальные исследования изоповерхностей многомерных данных.

Предлагается использовать алгоритм ненаправленной маркировки [5] для построения локальных частей изоповерхности, поскольку применение этого алгоритма позволит свести вычислительные затраты к O(K), что является более эффективным решением задачи построения локальной части изоповерхности.

1 Алгоритм ненаправленной маркировки

Алгоритм ненаправленной маркировки представляет собой модификацию алгоритма быстрого марша, предназначенного для решения более узко специализированной задачи построения скалярного поля T, представляющее решение так называемого Эиконального уравнения:

$$|\nabla T|,$$

где T – скалярное поле, T= 0 в точках распространения.

Поле T является аппроксимацией расстояния до точек распространения. Алгоритм ненаправленной маркировки отбрасывает процесс вычисление поля T путем нахождения минимальных значений, оставляя процесс последовательного расширения последовательности за счет соседних точек.

Алгоритм ненаправленной маркировки основан на использовании последовательности точек (описанных лишь своими координатами в исходном объеме данных), представляющей линию фронта

распространения от исходных точек - очага распространения. Для простоты будем называть ее последовательностью линией распространения (ПЛР).

ПЛР может быть модифицирована только двумя способами: добавление новой точки в конце последовательности и удаление точки в ее начале. Тогда алгоритм можно описать в виде следующего набора правил.

а) В начальном состоянии ПЛР представляет собой последовательность начальных точек.

б) Каждый следующий шаг представляет собой добавление 4-х соседних точек в конец ПЛР.

г) После того как все 4 точек окружения проанализированы, рассматриваемая точка уделяется из ПЛР.

д) Шаги в) и г) выполняются до тех пор, пока ПЛР содержит хотя бы одну точку.

Пример работы алгоритма показан на рис. 1.

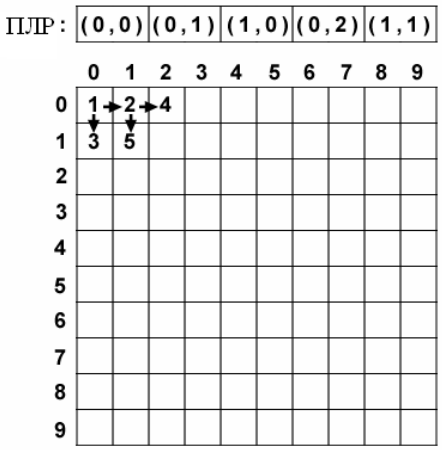


Рис. 1 –Пример построения ПЛР

2 Построение локальной части изоповерхности в трехмерном измерении

Предлагаемое применение алгоритма ненаправленной маркировки является интерактивным процессом, т.е. требует указания начальной ячейки для распространения со стороны пользователя. Далее алгоритм автоматически обходит все активные ячейки локальной части изоповерхности, проходящей через указанную ячейку, начиная с нее.

Очевидно, описанный для двухмерного случая алгоритм несложно модифицировать к работе в трехмерном измерении. Для этого изменяется лишь пункт б) добавления соседних точек. Кроме того, на этом же этапе соседние ячейки анализируются на предмет активности

и добавляются в ПЛР только в том случае, если действительно являются активными.

В результате алгоритм обходит только точки исследуемой локальной части изоповерхности, что сводит время работы алгоритма к O(K), где K – количество ячеек искомой части изоповерхности.

Специфика алгоритма также позволяет исследовать структуру внутреннего объема построенной изоповерхности, так как последующее расширение «внутрь» позволяет эффективно быстро измерить объем и восстановить скелет [5].

Заключение

Таким образом, предлагаемое применение алгоритма ненаправленной маркировки для построения локальных частей изоповерхности позволяет свести вычислительные затраты к O(K), что является более эффективным решением задачи построения изоповерхности по сравнению с алгоритмом POT, а также обеспечивает учет внутреннего объема построенных изоповерхностей для дополнительных исследований.

Перечень ссылок

1. W. E. Lorensen and H. E. Cline., Marching Cubes: A high resolution 3D surface construction algorithm // ACM SIGGRAPH Computer Graphics, 1987, vol. 21(4), pp.163-169.

2. A. Mascarenhas, M. Isenburg, V. Pascucci, and J. Snoeyink, Encoding volumetric grids for streaming isosurface extraction // 2nd International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT 2004), 2004, pp. 665-672.

3. S. Parker, P. Shirley, Y. Livnat, C. Hansen, and P.-P. Sloan, Interactive ray tracing for isosurface rendering // IEEE Visualization, 1998, pp. 233-238.

4. Q. Shi, J. JaJa, Isosurface extraction and spatial filtering using Persistent OcTree (POT) // IEEE Transactions on Visualization and Computer Graphics, vol. 12(5), 2006, pp. 1283-1290.

5. Белоус Н.В., Кобзарь Г.А., Маркировка компонентов бинарных изображений с параллельным выделением их контура и скелета // Компьютерное моделирование и интеллектуальные системы: сборник научных трудов / Под ред. Д.М. Пизы, С.О. Субботина. – Запорожье: ЗНТУ, 2007. - С. 17 – 22.

Поступила 12.05.2007 г.

УДК 004.932.75:519.76

ІНТЕРАКТИВНА АВТОМАТИЗОВАНА ГЕОІНФОРМАЦІЙНА СИСТЕМА ВИСОКОГО РІВНЯ БЕЗПЕКИ ОСОБЛИВО ВАЖЛИВИХ ОБ'ЄКТІВ

М.І.Васюхін, І.В. Васильєв, Н.М.Лобанчикова
Національний авіаційний університет (м. Київ)

В сучасних умовах підвищення рівня терористичної діяльності, досить актуальними стають проблеми побудови систем захисту особливо важливих об'єктів. В роботі запропоновано методи побудови інтерактивної автоматизованої геоінформаційної системи високого рівня безпеки особливо важливих об'єктів за допомогою інтегрованих модулів.

1. Вступ

З кожним роком збільшується кількість надзвичайних ситуацій, які виникають з різних причин. Серед джерел виникнення провідне місце посідають терористичні акти, тому сьогодні гостро постає питання захисту особливо важливих об'єктів від несанкціонованого втручання з метою запобігання виникненню надзвичайних ситуацій.

2. Основні завдання системи

Процес створення досконалою системи захисту ускладнюється багатьма чинниками, основними з яких є: велика територія об'єкту охорони, різнотипність рельєфу території, масовість об'єктів, що знаходяться на території, наявність агресивних середовищ та ін.

Основним завданням системи безпеки є своєчасне отримання інформації для прийняття адекватного рішення, яке повинно ґрунтуватися на достовірній, поточній і прогнозованій інформації, аналізі всіх факторів, що впливають на прийняття рішення з урахуванням можливих наслідків.

Досить актуальною в системах захисту є функція моніторингу об'єктів, в тому числі і людей, які перебувають на території об'єкту охорони в реальному часі та відображення результатів моніторингу на екранах колективного користування.

3. Структура та склад системи

В проектуванні сучасних автоматизованих систем захисту особливо важливих об'єктів застосовується системний підхід для підвищення якості та достовірності роботи. Єдині принципи побудови

та функціонування таких систем створюють передумови їх інтерграції з іншими життєзабезпечуючими системами об’єктів охорони, цілеспрямованої і узгодженої роботи численних складових.

Основною метою створення інтегрованої автоматизованої геоінформаційної системи високого рівня безпеки особливо важливих об'єктів є підвищення ефективності роботи всіх складових системи захисту за рахунок комплексного використання переваг її компонентів і сучасних методів та засобів захисту об'єктів. Поставлена нами мета досягається синтезом інтегрованих блоків, рис.1, а саме:

- інтерактивної геоінформаційної системи відображення інформації (ІГС ВІ) ;
- системи моніторингу території особливо важливих об’єктів.

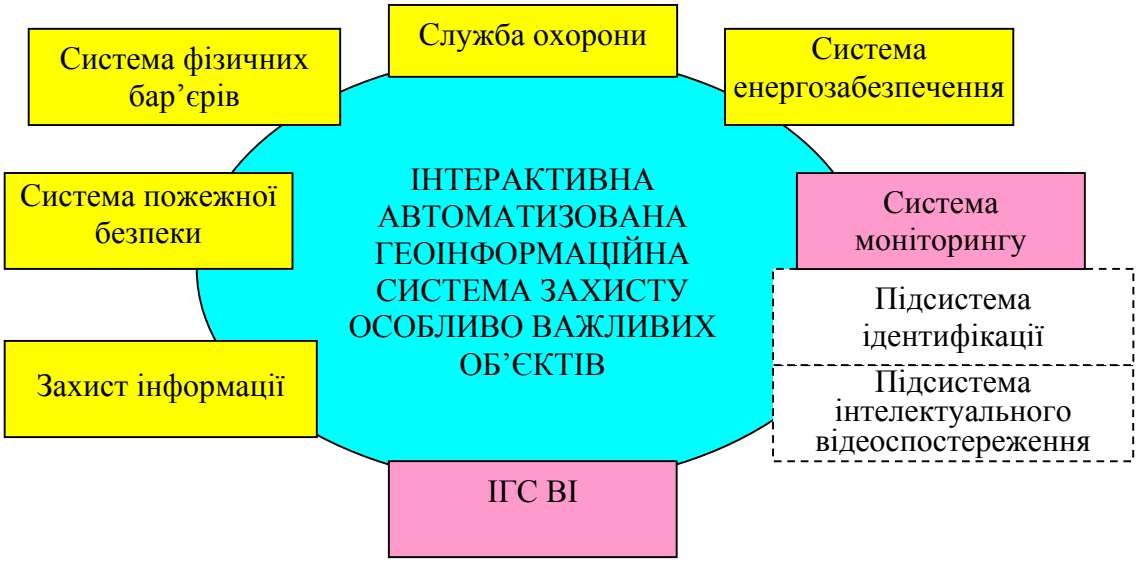


Рис.1. Структура інтерактивної автоматизованої геоінформаційної системи захисту особливо важливих об’єктів.

Система моніторингу в свою чергу включає наступні підсистеми: підсистему інтелектуального відеоспостереження та підсистему ідентифікації.

Підсистема відеоспостереження передбачає підключення інтелектуальних функцій на базі нейрокомп’ютерних мереж та спеціального програмного забезпечення. Дана підсистема забезпечує можливість швидкої передачі відеоінформації про надзвичайні події персоналу служби безпеки для оперативного прийняття адекватного рішення. Використання різних методів обробки відеоінформації перетворюють систему відеоспостереження в інструмент автоматизованого аналізу відеозображень, дозволяючи

відслідковувати об’єкти, що нас цікавлять, аналізувати траєкторії руху і поведінку людей, реєструвати потенційно небезпечні предмети, в автоматичному режимі проводити їх ідентифікацію, в тому числі транспортних засобів та людей.

Підсистема ідентифікації поділяється на систему ідентифікації транспортних засобів та систему ідентифікації особистості. Система ідентифікації транспортних засобів побудована з використанням методів та засобів супутникового позиціонування. Так, службові транспортні засоби повинні бути оснащенні супутниковим приймачем та пристроєм передачі інформації, який буде слугувати ідентифікатором того чи іншого транспортного засобу.

Підсистема ідентифікації особистості являє собою інтегровану автоматизовану систему з відповідним програмним забезпеченням. Проведений аналіз відомих методів біометричної ідентифікації особистості показав, що найбільш ефективною є комбінація наступних методів:

- метод ідентифікації по відбитку пальців,
- метод ідентифікації по голосу,
- метод ідентифікації особистості по зображенню її обличчя.

Використання сукупності цих методів, в значній мірі, підвищує точність проведення ідентифікації. В системі використовується розподілена система обрахунків та обробки параметрів. Для усунення затримки обробки даних пропонується використання кластерних систем обробки інформації в комбінації з RFID-чіпами з можливістю їх шифрування. Останні дають можливість проведення ідентифікації особистості на відстані до 30 метрів. За допомогою опрацювання сигналу з RFID-чіпів, ми можемо прискорити завантаження біометричних даних з баз даних для ідентифікації особистості. Цього досягаємо за рахунок часу, який потрібен людині для підходу до системи ідентифікації (щоб пройти шлях 30 метрів). Коли людина підходитиме до пристроїв сканування, база даних вже буде завантажена даними, які система отримала від шифрованого чіпа. Таким чином, система не перевіряє дані за всіма базами, а тільки завантажує їх в операційну систему та перевіряє їх з даними зчитувачів.

Крім того, виникає необхідність відображення інформації про моніторинг на моніторах колективного користування (МКК) в реальному часі.

Для зручності та об’єктивності роботи з масивами інформації, нами пропонується застосування геоінформаційної системи. Така система буде відображати реальну обстановку з прив’язкою до

території у вигляді динамічних сцен на картографічному фоні. Це спростить процедуру знаходження оптимального шляху до місця виникнення надзвичайних ситуацій. Користувач системи відразу зможе зорієнтуватися про залучення необхідних ресурсів для ліквідації НС та локалізувати або знищити джерело виникнення НС.

Диспетчерський пункт обладнаний великим екраном колективного користування, рис.2, істотно збільшує ефективність роботи персоналу разом із підвищенням зручності та комфорту сприйняття інформації.

Рис.2. Система відображення диспетчерської інформації на базі проекційної системи.

За допомогою системи комутації та розподілу сигналів організовується одночасне відображення на екрані інформації з комп'ютерів в локальній мережі диспетчерського пункту, а також можливе підключення джерел відеосигналів в повноекранному режимі або в режимі «зображення-в-зображенні»: відеокамер спостереження, камер відеоконференції, відеомагнітофонів, DVD-програвачів, TV-приймачів і т.п.

В процесі інтерактивної взаємодії користувач розпізнає повідомлення, що надходять у систему, перевіряє повноваження користувачів, виділяє необхідні обчислювальні ресурси, ініціює і координує спільну роботу засобів вводу-виводу, СУБД і ППК при виконанні запитів за відповідними циклограмами. По ходу надходження повідомлень і видачі розпоряджень за допомогою засобів вводу-виводу оперативно відображаються вхідні, проміжні і вихідні дані, які за необхідністю коригуються користувачем в режимі

діалогу типу меню. Залежно від рівня кваліфікації користувачів і виду діалогу варіанти меню-подібних запитань і відповідей можуть варіюватися по формам представлення і змістовному наповненню. Сервісні функції діалогу можуть розширюватися, зокрема, за рахунок багатовіконного інтерфейсу, що дозволяє паралельно відображати на відеотермінальному пристрої декілька фрагментів складної обстановки у зручному для сприйняття картографічному вигляді з необхідними кількісними та якісними характеристиками.

Ключові особливості системи: висока ефективність для організації моніторингу і управління окремими технологічними процесами, що вимагають уваги з боку диспетчерської служби, висока надійність і ремонтопридатність, тривалий термін експлуатації, підтримка цілодобового режиму роботи, можливість поступового нарощування і модернізації системи.

***Висновки**:*

В результаті проведеного аналізу автоматизованих систем захисту особливо важливих об'єктів було запропоновано методи побудови інтерактивної автоматизованої геоінформаційної системи високого рівня безпеки особливо важливих об'єктів при умові підключення до її складу інтегрованих модулів, а саме, інтерактивної геоінформаційної системи відображення інформації та системи моніторингу території особливо важливих об'єктів. Впровадження такого роду інтегрованих підсистем в автоматизовані системи захисту особливо важливих об'єктів, істотно зменшить обсяги інформації для опрацювання службою безпеки, полегшить процес прийняття рішень і підвищить швидкість реагування, що, в свою чергу, зведе до мінімуму «людський» фактор, та в значній мірі підвищить рівень захисту особливо важливих об'єктів.

Література

1.Васюхин М.И. Основы интерактивных навигационно-управляющих геоинформационных систем: Монография / М.И.Васюхин.- К.:Лира-К,2006.-536 с.

2. Пюшки Л. Методы и средства построения автоматизированных интегрированных систем защиты особо важных объектов: Дис…канд.техн.наук 05.13.06 /Институт кибернетики имени В.М. Глушкова.– Киев, 2005.–156 с.

3. Ряботягов А.В.Метод ідентифікації людини на основі індивідуального мовного коду: автореф. дис...канд.техн.н. 05.13.23/Харківський національний університет радіоелектроніки.– Харків,2006.–20 с.

4. Васюхин М.И., Лобанчикова Н.Н., Столяренко Р.А. Анализ методов обработки видеоинформации в системах видеонаблюдения// Вісник Херсонського Національного технічного університету.–2006.–№1(24).– С.271-275.

Отримано 15.05 2007 р.

УДК 681.3

ПРИМЕНЕНИЕ МУЛЬТИМНОЖЕСТВ ДЛЯ ВЫДЕЛЕНИЯ РЕГИОНОВ ИЗОБРАЖЕНИЙ

О.Л. Вовк
Донецкий национальный технический университет

У роботі розглядаються мультимножини для класифікації об'єктів, заданих суперечливими різнорідними ознаками (зокрема кількісними і якісними одночасно). Пропонується метод виділення регіонів зображень за допомогою теорії мультимножин.

При решении задач распознавания образов, классификации объектов, многокритериального принятия решений часто возникает проблема группировки, упорядочивания объектов, заданных разнородными признаками (которые могут быть как качественными, так и количественными). Кроме того, возможны ситуации, когда один объект может существовать в нескольких экземплярах с отличными значениями признаков, свертка которых невозможна или математически некорректна. В качестве примеров таких задач можно отметить: классификацию и ранжирование объектов, оцененных несколькими экспертами, распознавание изображений, обработку текстовых документов [1].

Среди актуальных задач обработки изображений можно выделить задачу выделения регионов изображений по цветовому подобию пикселей (кластеризация изображений) [2, 3]. При анализе методов разнесения пикселей по кластерам возможна противоречивая ситуация, так как согласно различным способам расчета близости пикселя к кластерам, один пиксель может входить в разные кластеры. В условиях автоматического выделения кластеров изображений выбор соответствующего пикселю кластера затруднен.

Мультимножества (множества с повторяющимися элементами) [1, 4] служит удобной математической моделью для задания объектов, которые характеризуются разнородными противоречивыми признаками.

Цель данной работы – дать общее определения мультимножества; рассмотреть применение теории мультимножеств для усовершенствования методов кластеризации изображений.

Перейдем к краткому обзору основных понятий и определений теории мультимножеств [1, 4].

Мультимножеством A, порожденным обычным множеством $U=\{x_1,x_2,\ldots\}$, все элементы которого различны, называется совокупность групп элементов вида $A=\{k_A(x)\bullet x|x\in U,\ k_A(x)\in Z_+\}$. Здесь $k_A:U\to Z_+=\{0,1,2,\ldots\}$ называется функцией числа экземпляров мультимножества, определяющей кратность вхождения элемента $x_i\in U$ в мультимножество A, что обозначено символом •.

Если $k_A(x)=\chi_A(x)$, где $\chi_A(x)=1$ при $x\in A$ и $\chi_A(x)=0$ при $x\notin A$, то мультимножество A становится обычным множеством.

Если все мультимножества семейства $A=\{A_1,A_2,\ldots\}$ образуются из элементов множества G, то G называется доменом для семейства A, а множество $\mathrm{Supp}A=\{x|x\in G,\ \chi_{\mathrm{Supp}A}(x)=\chi_A(x)\}$ – опорным множеством или носителем мультимножества A.

Мощность мультимножества $|A|=\Sigma_x k_A(x)$ определяется как общее число экземпляров всех его элементов; размерность мультимножества $/A/=\Sigma_x\chi_A(x)=|\mathrm{Supp}A|$ – как общее число различных элементов. Максимальное значение функции кратности $\mathrm{hgt}A=\max\limits_{x\in G} k_A(x)$ называется высотой, а элемент $x_{A*}=\arg\max\limits_{x\in G} k_A(x)$ – пиком мультимножества A. Мультимножество называется пустым $\varnothing$, если $k_\varnothing(x)=0$, и максимальным Z, если $k_Z(x)=\max\limits_{A\in\mathrm{A}} k_A(x)$, $\forall x\in U$.

Рассмотрим возможные способы сопоставления мультимножеств, обусловленные особенностями их различных характеристик. Мультимножества A и B называются равными ($A=B$), если $k_A(x)=k_B(x)$ для всех элементов $x\in G$, и неравными ($A\neq B$), если $k_A(x)\neq k_B(x)$ хотя бы для одного $x\in G$. Для равных мультимножеств имеем $|A|=|B|$, $/A/=/B/$, $\mathrm{hgt}A=\mathrm{hgt}B$, $x_{A*}=x_{B*}$, $\mathrm{Supp}A=\mathrm{Supp}B$. Мультимножества A и B будем называть равномощными, если $|A|=|B|$; равноразмерными, если $/A/=/B/$; равновеликими, если они равномощны и равноразмерны. Равные мультимножества равновелики, обратное утверждение, вообще говоря, неверно.

Будем говорить, что мультимножество B содержится или включено в мультимножество A ($B\subseteq A$), если $k_B(x)\leq k_A(x)$, для каждого элемента $x\in G$. Мультимножество B называется тогда подмультимножеством мультимножества A, а мультимножество A – надмультимножеством мультимножества B. В этом случае $|B|\leq|A|$, $/B/\leq/A/$, $\mathrm{hgt}B\leq\mathrm{hgt}A$, $\mathrm{Supp}B\subseteq\mathrm{Supp}A$, а $x_{A*}=x_{B*}$, либо $x_{A*}\neq x_{B*}$. Как и в случае обычных множеств, одновременное выполнение условий $B\subseteq A$ и $A\subseteq B$ влечет равенство мультимножеств $A=B$. Включение мультимножества обладает свойствами рефлексивности ($A\subseteq A$) и

транзитивности ($A\subseteq B$, $B\subseteq C \Rightarrow A\subseteq C$), а значит, является отношением предпорядка.

Мультимножества A и B будем называть одноименно или S-эквивалентными ($A\cong B$), если их носители совпадают (Supp$A$=Supp$B$) и существует взаимно однозначное соответствие f между одноименными компонентами: $k_B(x)=f(k_A(x))$, $\forall x \in G$; разноименно или D-эквива-лентными ($A\underset{\sim}{\approx}B$), если их носители эквивалентны (Supp$A$~Supp$B$) и существует взаимно однозначное соответствие f между разноименными компонентами: $k_B(x_i)=f(k_A(x_j))$, $x_i,x_j\in G$, где f – целочисленная функция с областью значений Z_+.

Введем следующие основные операции над мультимножествами:

- объединение $A\cup B = \{k_{A\cup B}(x)\bullet x \mid k_{A\cup B}(x)=\max(k_A(x), k_B(x))\}$;
- пересечение $A\cap B = \{k_{A\cap B}(x)\bullet x \mid k_{A\cap B}(x)=\min(k_A(x), k_B(x))\}$;
- арифметическое сложение $A+B = \{k_{A+B}(x)\bullet x \mid k_{A+B}(x)=k_A(x)+k_B(x)\}$;
- арифметическое вычитание $A-B = \{k_{A-B}(x)\bullet x \mid k_{A-B}(x)=k_A(x)-k_{A\cap B}(x)\}$;
- симметрическая разность $A\Delta B = \{k_{A\Delta B}(x)\bullet x \mid k_{A\Delta B}(x)=|k_A(x)-k_B(x)|\}$;
- дополнение $\overline{A} = Z-A = \{k_{\overline{A}}(x)\bullet x \mid k_{\overline{A}}(x)=k_Z(x)-k_A(x)\}$.

Другие операции, а также способы определения носителей операций над мультимножествами рассмотрены подробно в [4].

Действительная неотрицательная функция $m(A)$, определенная на алгебре L (Z) и удовлетворяющая условию коаддитивности: $m(A)+m(B)=m(A+B)$, называется мерой мультимножества. Мера мультимножества $m(A)$ обладает следующим свойствами: $m(\varnothing)=0$; монотонность $m(A)\le m(B)\Leftrightarrow A\subseteq B$; непрерывность $\lim\limits_{i\to\infty} m(A_i)=m(\lim\limits_{i\to\infty} A_i)$; симметричность $m(A)+m(\overline{A})=m(Z)$; эластичность $m(h\bullet A)=hm(A)$. Меру мультимножества можно определить различными способами, например, как линейную комбинацию функций кратности: $m(A)=\sum_j w_j k_A(x_j)$, $w_j>0$. Заметим, что мощность мультимножества $|A|$ также будет мерой мультимножества

Метрические пространства мультимножеств (A ,d) введены в [4], где определены следующие виды расстояний между мультимножествами:

$$d_1(A,B) = m(A\Delta B);\quad d_2(A,B) = m(A\Delta B)/m(Z);\qquad (1)$$
$$d_3(A,B) = m(A\Delta B)/m(A\cup B).$$

Функции $d_2(A,B)$ и $d_3(A,B)$ удовлетворяют условию нормировки $0\le d(A,B)\le 1$. По определению принимается $d_3(\varnothing,\varnothing)=0$. Основное расстояние $d_1(A,B)$ является метрикой типа Хемминга, традиционно используемым во многих приложениях. Полностью усредненное

расстояние $d_2(A,B)$ характеризует различие между двумя мультимножествами A и B, отнесенное к расстоянию, максимально возможному в исходном пространстве. Локально усредненное расстояние $d_3(A,B)$ задает различие, отнесенное к максимально возможной «общей части» только этих двух мультимножеств в исходном пространстве.

Рассмотрим решение задачи кластеризации изображений с помощью теории мультимножеств.

В качестве метода кластеризации для усовершенствования выберем наиболее распространенный и используемый метод k-средних [5,6].

В основе анализируемого метода – отнесение пикселя к кластеру с минимальным расстоянием до центра тяжести кластера, проверка удовлетворения критерия оптимальности разбиения на заданное количество кластеров и критерия окончания кластеризации. Рассмотрим усовершенствования этапа разнесения пикселей по кластерам. Существует множество методов расчета расстояний между отдельными парами числовых характеристик объектов (к примеру, Евклидово расстояние, расстояние Хемминга, корреляционное расстояние и другие), кроме того, могут использоваться различные модификации перечисленных методов при расчете расстояний между объектами, заданными несколькими признаками (метод суммирования ошибки, метод минимизации ошибки и т.д.) [5]. При выборе того или иного из перечисленных методов, результаты кластеризации в большинстве случаев получаются различными. Обозначим набор всех возможных числовых значений расстояний между пикселем и кластером – (где m – количество всех рассматриваемых расстояний). Пусть для группировки пикселей по кластерам мы имеем m оценок ($Q_1, Q_2,\ldots,Q_m$), каждая из которых рассчитывается s способами, т.е. $Q_i=\{ q_i^s \}$.

Предположим, что у нас имеется k пикселей – $A=\{A_1,\ldots,A_k\}$, каждый из которых мы относим к одному из p кластеров (r_t, $t=\overline{1,p}$).

Тогда критерий выбора кластера для отдельного j-го пикселя ($j=\overline{1,k}$) можно записать в виде:

$$A_j=\{(k_{Aj}(q_i^s)\bullet q_i^s), k_{Aj}(r_t) \bullet r_t\}, \qquad (2)$$

где $k_j(x_i)$ – число анализируемых методов расчета расстояний между пикселем j и кластером t, согласно которым j-ый пиксель изображения относится к t-ому кластеру.

Таким образом, решение задачи аппроксимации решающих правил для отнесения каждого пикселя изображения к тому или иному кластеру можно представить в виде решения m оптимизационных задач:

$$d(Q_{jr1},Q_{jr2},\ldots,Q_{jrp}) \rightarrow \max d(Q_{jr1},Q_{jr2},\ldots,Q_{jrp}) = d(Q_{jr1}^*,Q_{jr2}^*,\ldots,Q_{jrp}^*). \quad (3)$$

Результирующее для классификации правило можно записать в виде:

ЕСЛИ $((q_{ju} \in Q_{ut}^*)$ И $(q_{jv} \in Q_{vt}^*) \ldots$ И $(q_{jw} \in Q_{wt}^*))$
ТО (пиксель $A_j \in X_t$), (4)

где u, v, $\ldots w$ – некоторые анализируемые способы расчета расстояний.

В заключении хотелось бы отметить, что аппарат мультимножеств можно эффективно применять и при решении других задач обработки изображений. В качестве направлений дальнейших исследований хотелось бы выделить построение системы контекстного поиска изображений в базах данных, в основе механизма поиска которой упорядочивание изображений базы данных с использованием теории мультимножеств.

Литература

1. Петровский А.Б. Упорядочение и классификация объектов с противоречивыми признаками // Новости искусственного интеллекта. – 2003. – №4. – 17 с.
2. Башков Е.А., Вовк О.Л. Статистическая кластеризация для выделения регионов изображений // Збірник праць V Міжнародної наукової конференції “Інтелектуальний аналіз інформації”. – “Просвіта”, Київ. – 2005. – С. 50-59.
3. Li J., Wang J.Z., Wiederhold G. IRM: Integrated Region Matching For Image Retrieval // ACM Multimedia 2000. – 2000. – P. 147-156.
4. Петровский А.Б. Пространства множеств и мультимножесв. – Москва: Едиториал УРСС, 2003. – 248 с.
5. *Jain A.K., Murty M.N., Flynn P.J. Data Clustering: A Review // ACM Computing Surveys. – 1999. – vol. 31, №3. – P. 264-323.*
6. Kanungo T., Mount D., Netanyahu N., Piatko C., Silverman R., Wu A. An Efficient k-Means Clustering Algorithm: Analysis and Implementation // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 2002. – vol. 24, №7. – P. 881-892.

Поступление в редакцию 29.05.2007 г.

УДК 681

ТРЕХМЕРНАЯ АНИМАЦИЯ И ДИЗАЙН В РЕКЛАМЕ

В.В. Карабчевский, А.В. Грыбинныk
Донецкий национальный технический университет

В статье рассматриваются вопросы использования компьютерной 3D-графики и 3D-анимации для рекламирования товаров и услуг, а также изучение основ рекламного дела и методов трехмерного моделирования и дизайна в курсе «Реклама и компьютерная графика».

Реклама является важной областью деятельности значительной части предприятий, поэтому изучение основ рекламного дела, методов моделирования рекламной деятельности и прогнозирования ее экономической эффективности включено в программу подготовки студентов специальности «Экономическая кибернетика», обучающихся на кафедре ПМИ ДонНТУ. Наряду с этими вопросами в курсе «Реклама и компьютерная графика» изучаются методы разработки рекламных обращений с использованием средств компьютерной графики. Такой подход к определению содержания курса позволяет обеспечить выполнение курсовых работ, целью которых является планирование рекламной кампании для различных товаров и услуг и разработка соответствующих рекламных обращений.

На постсоветском пространстве компьютерное моделирование редко используется для рекламы в силу ряда причин, среди которых:

– высокая стоимость данного вида рекламы;

– неразвитость данного вида рекламы на нашем рынке рекламных продуктов;

– шаблонность работы большинства бренд-менеджеров и рекламных агентств, зачастую использующих и предлагающих традиционные виды рекламы.

Однако же, использование данного средства рекламы продолжает расти. О чем свидетельствует множество объявлений в Internet, предлагающих заказать рекламный 3D-ролик.

В условиях современной информационной перегруженности реклама зачастую выступает раздражителем, защищаясь от которого, люди стараются игнорировать рекламные обращения, не запоминая их и не вдаваясь в подробности. В этом случае приобретает острую актуальность вопрос поиска такого рекламного инструмента, который будет способен выделить конкретное рекламное обращение из

сплошного информационного потока. Таким инструментом может послужить 3D-анимация.

Анимационная реклама имеет ряд преимуществ:

1. Во-первых, анимационный ролик в силу своей оригинальности и образности не создает информационной и эмоциональной нагрузки на зрителя, напротив, развлекает;
2. Анимационные приемы динамичны, образно наполнены, а значит, и действенны с точки зрения воздействия на потребителя:
3. Возможности анимационных персонажей практически безграничны;
4. Свежесть: отсутствие на экранах большого количества анимационной рекламы дает преимущество тем фирмам, которые обращаются к ней в данный момент;
5. Анимационные ролики привлекают внимание, в первую очередь, детей, поэтому идеальны для рекламы детских товаров [1].

В качестве средства для разработки анимационных рекламных роликов выбран пакет 3D Studio MAX [2]. Изучить все его инструменты в односеместровом курсе не представляется возможным, однако имеющийся на кафедре ПМИ опыт обучения студентов применению методов и инструментов трехмерной компьютерной графики позволяет освоить основы работы в 3D Studio MAX и применить их в ходе курсового проектирования. Приведенное ниже краткое описание курсовой работы свидетельствует об успешности такого подхода.

Работа посвящена рекламе услуг туристического комплекса «Буковель», расположенного в Ивано-Франковской области. Основное внимание было уделено рекламной акции, в рамках которой, покупая абонемент на зимний сезон 2006/07 в предшествующий летний период, потребитель имел возможность приобрести его по сниженной цене. Что касается ролика, то реализован он довольно простыми средствами пакета, освоить которые можно в короткие сроки. Но при простоте реализации данная работа способна легко привлекать внимание зрителя, вызывать позитивные эмоции, а следовательно формировать у зрителя лояльное отношение к рекламируемому объекту.

Главный акцент при создании рекламы ТК «Буковель» был сделан на пользе здорового образа жизни и активного отдыха. Общая рекламная идея сформирована следующим образом: «Отдых в ТК «Буковель» - чудесная возможность весело и с пользой для здоровья провести время в кругу самых дорогих Вам людей, наслаждаясь природой Карпатских гор и покоряя горнолыжные трассы. При этом

Вы не отягощаете себя ненужной суетой, потому что огранизацию Вашего отдыха доверяете профессионалам.»

При планировании рекламной кампании были выполнены следующие основные шаги:

- определены общие цели фирмы и выполнен анализ текущего положения;
- была выбрана маркетинговая стратегия;
- исследовался рынок на наличие конкурентов, был выявлен потенциал рынка, анализировались модели поведения потребителей;
- поставлены цели рекламной кампании;
- на основе того анализа, который был проделан, осуществлялось уже непосредственно моделирование рекламной кампании: была выбрана основная рекламная идея, целевая аудитория, каналы коммуникации, определен бюджет рекламной кампании, виды используемых рекламных носителей, интенсивность, составлен календарный план использования СМИ, определены действия в случае изменения оперативной информации, выбран способ оценки эффективности рекламной кампании, составлена смета затрат.

Работа над рекламным роликом проводилась поэтапно, включая следующие шаги:

1. разработка сценария;
2. создание объектов;
3. организация движения объектов, освещения и движения камер;
4. визуализация.

При создании объектов анимационного ролика были использованы примитивные объекты (капсула, конус, бокс, плоскость) и сплайны, к которым применялись основные модификаторы и деформаторы (выдавливание, изгибание, выдавливание в несколько слоев, зеркальное отображение, модификатор Edit Mesh и другие) (рис.1).

Для большинства объектов были использованы текстуры. Их использование придает образности, позволяет сделать объекты похожими на реальные, создать пейзажи (рис. 2).

При задании анимации организовано движение объектов и камер. В одной из сцен смещение объектов и камер для каждого кадра задавалось вручную. В других сценах движение объекта происходит по указанной траектории (рис. 3).

Рисунок 1 – Кадр из готовой работы

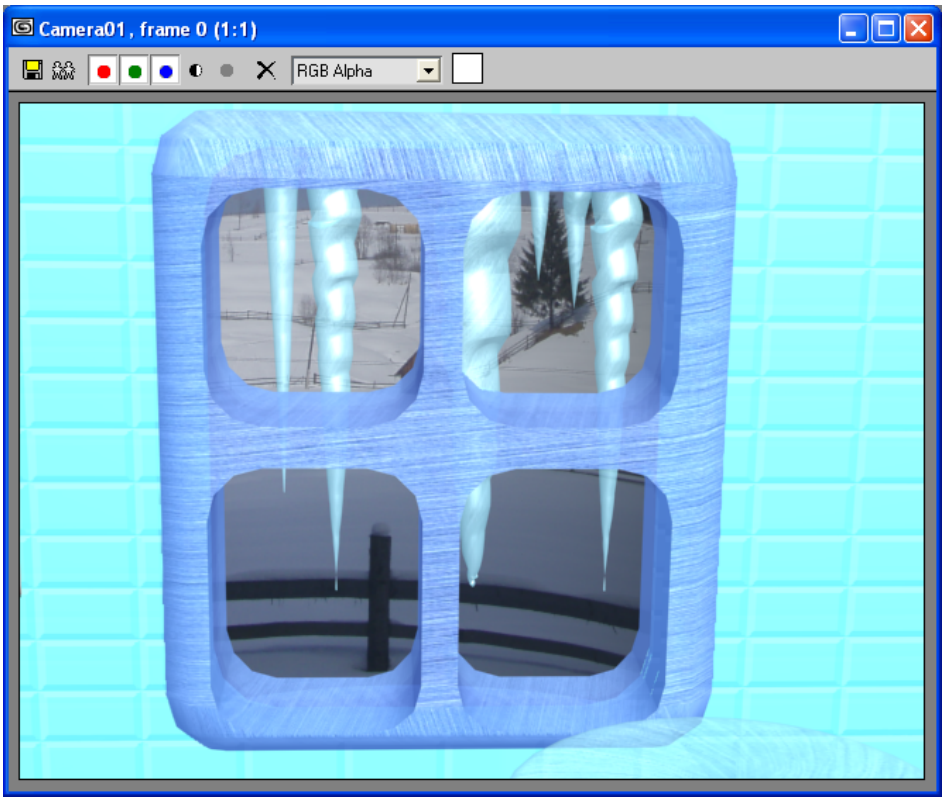


Рисунок 2 – Пример использования текстуры для создания пейзажа за окном

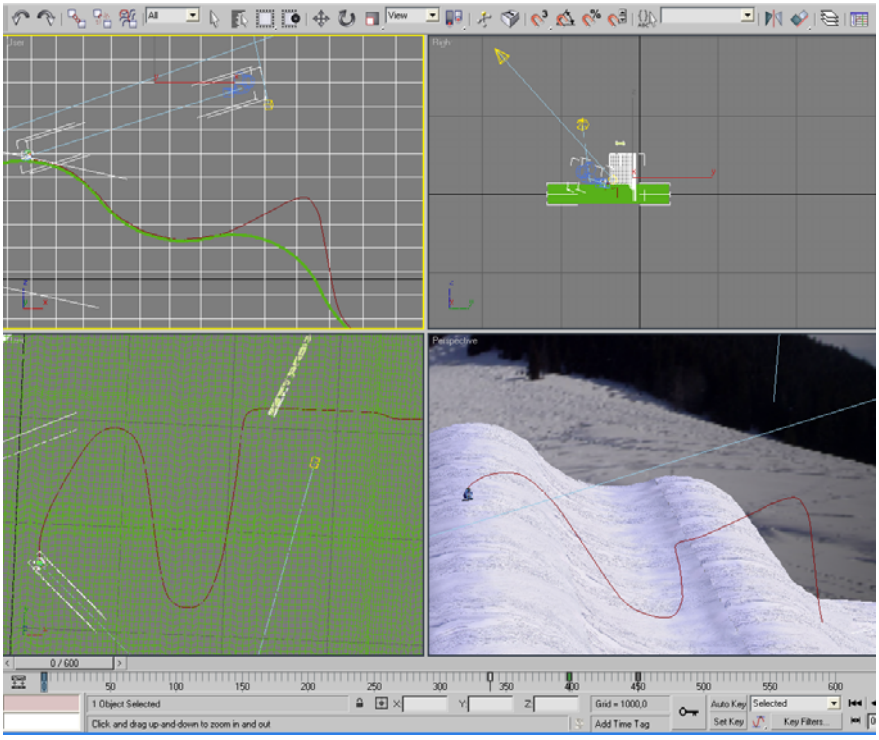


Рисунок 3 – Организация движения по траектории

Кроме того, для создания анимации использованы стандартные динамические объекты, например, снег.

При визуализации использован эффект Blur, что снизило резкость изображения, это позволило сделать изображение сглаженным и смягчить переходы между цветами и гранями объектов.

Опыт преподавания курса «Реклама и компьютерная графика» позволяет сделать следующие выводы.

Изучение средств разработки компьютерных анимационных роликов повышает интерес к предмету.

Минимальный набор компьютерных инструментов, необходимый для разработки рекламных роликов приемлемого качества, может быть изучен и практически применен в курсовом проектировании в течение одного семестра.

Многие курсовые работы носят реальный характер и служат основой для дипломного проектирования.

Литература

1. www.adme.ru
2. Мердок, Келли, Л. 3ds max 7. Библия пользователя. – М.: Издательский дом «Вильямс», 2006. – 1296 с.

Поступила 05.06.2007

УДК 61.681

МОДЕЛЬ УЗЛА GRID-СИСТЕМЫ НА ОСНОВЕ СЕТЕЙ ПЕТРИ

А.Ю. Шелестов
Институт космических исследований НАНУ-НКАУ

В доповіді запропоновано модель вузла Grid-системи на основі апарату мереж Петрі. Доведено основні властивості моделі: живучість, обмеженість, взаємне виключення та рівноправність.

Grid-система представляет собой распределенную систему доступа и обработки данных, состоящую из следующих основных компонентов: пользователи, формирующие запросы к системе, вычислительные узлы, выполняющие обработку данных, хранилища данных и управляющие узлы.

Задача состоит в построении Grid-системы, обеспечивающей прозрачную для пользователя обработку его запросов на поиск и обработку данных, в том числе, синхронизацию доступа к необходимым данным, обработку этих данных и предоставление пользователю результатов обработки. Такая Grid-система относится к Grid-системам смешанного типа (является одновременно и вычислительной и информационной [1]), поскольку она включает как высокопроизводительные вычислительные узлы так и обеспечивает доступ к данным распределенных хранилищ [**2**].

Синхронизация доступа к данным и их корректная обработка предполагает выполнение некоторых базовых свойств, а именно:

взаимное исключение (mutex) — синхронизация событий при обработке данных: два или более вычислительных узла не могут одновременно иметь доступ к общей области данных;

равноправие (fairness) — отсутствие дискриминации заданий. Если пользователь сформировал запрос (задание) и отправил его в систему, то обязательно наступит момент, когда это задание начнет выполняться и будет выполнено.

Работу Grid-системы смешанного типа без детализации можно проиллюстрировать рис. 1. Из рис. 1 видно, что управляющий узел обеспечивает синхронизацию при распараллеливании задания между ресурсами или при выполнении операций доступа к общей памяти.

Учитывая высокую стоимость компонентов Grid-систем, прежде чем приступать к построению системы, обладающей указанными свойствами, необходимо построить ее модель и исследовать свойства

этой модели. В докладе рассматривается модель функционирования вычислительного узла, работающего над общей памятью, и исследуются свойства этой модели.

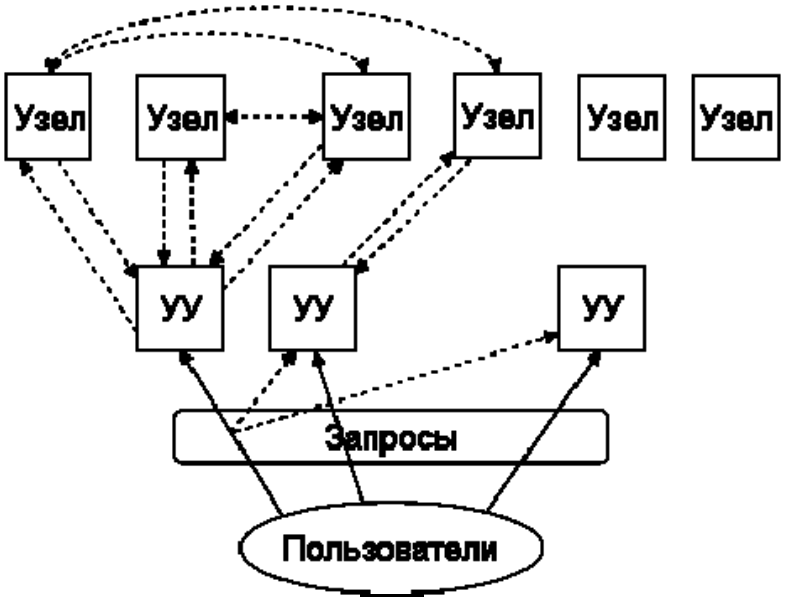


Рис. 1. Модель информационных потоков в Grid-системе

Учитывая высокую стоимость компонентов Grid-систем, прежде чем приступать к построению системы, обладающей указанными свойствами, необходимо построить ее модель и исследовать свойства этой модели. В докладе рассматривается модель функционирования вычислительного узла, работающего над общей памятью, и исследуются свойства этой модели.

Поскольку Grid-система представляет собой набор взаимодействующих между собой компонентов (вычислительных узлов и хранилищ данных), которые могут функционировать параллельно и в работе которых должна быть обеспечена синхронизация, наиболее подходящим математическим аппаратом для моделирования динамики таких систем являются сети Петри [3].

Модель работы узла в виде сети Петри

Модель работы вычислительного узла с 4 процессорами, обрабатывающего задания пользователя под управлением маршрутизатора в соответствии с общей схемой рис. 1 показана на рис. 2. Эта модель представляет собой одноцветную сеть Петри вида

$$C = (P, T, A), \qquad (1)$$

где P и T множества позиций и переходов сети Петри соответственно, A - матрица инцидентности графа сети, связывающая позиции с переходами сети. Начальное состояние сети описывается разметкой

M_0=(proc, want, query, block, free, busy, wait, work)=(4, 0, 0, 1, 1, 0, 0, 0).

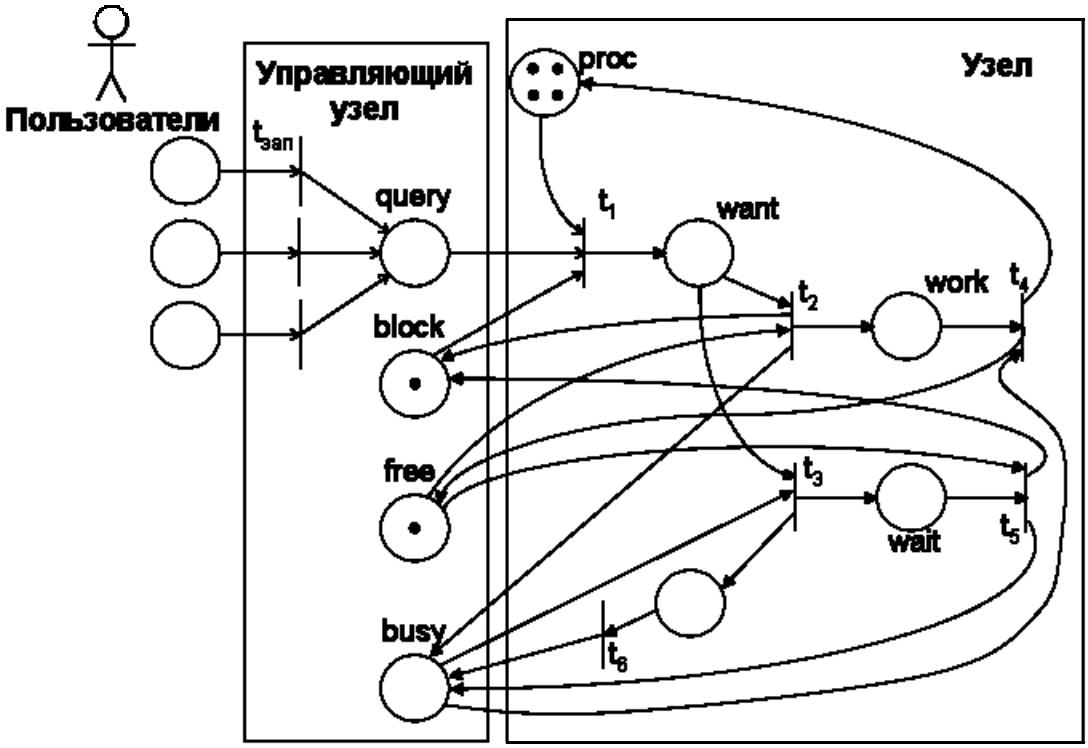


Рис. 2. Модель вычислительного узла, работающего под управлением планировщика над общей памятью с глубиной очереди 1

Позиции и переходы в данной сети Петри (СП) имеют следующую семантику.

proc — свободные в данный момент времени процессоры;

query — запрос на выполнение задания от пользователя;

want — процессор активен и желает начать работу;

work — процессор получил доступ к общей памяти и выполняет вычисления;

wait — активный процессор ожидает доступа к общей памяти;

block — блокируется доступ к общей памяти и очереди ожидания, если они заняты другими процессорами;

free — управляющий узел сигнализирует о снятии блокировки общей памяти;

busy — управляющий узел сигнализирует о том, что общая память заблокирована для доступа.

При начальной разметке M_0= (4, 0, 0, 1, 1, 0, 0, 0) система находится в состоянии покоя (который иногда называют состоянием «сна»). Из семантики позиций следует и смысл переходов:

$t_{зап}$ — наличие запроса на обработку данных;

t_1 — один из свободных процессоров хочет работать;

t_2 — процессор получает доступ к общей памяти;

t_3 — активный процессор переходит в состояние ожидания;

t_4 — процессор завершил работу с общей памятью и она разблокирована;

t_5 — активный процессор из состояния ожидания переходит в состояние работы с общей памятью, при этом одновременно разблокируется переход активного процессора в состояние ожидания.

При появлении запроса пользователя на выполнение задания, срабатывает сначала один из переходов $t_{зап}$, а затем переход t_1. В результате этого один из свободных процессоров переходит в состояние активности (готовности к работе), т.е. переходит в позицию want. Если одновременно поступает 2 запроса, то выполнение одного из них будет задержано отсутствием маркера в позиции block. Дальнейшая работа модели очевидна.

Исследование свойств модели

Сначала исследуем структурные свойства построенной на рис. 2 сети Петри, а именно проверим выполнение свойства ограниченности сети Петри, удостоверимся в отсутствии мертвых (лишних) переходов, и недостижимых позиций.

Для этого воспользуемся уравнением состояний

$$Ax=0, \qquad (2)$$

где A — целочисленная n×m матрица инцидентности сети Петри, *n* и m — мощности множеств *P* и *T* соответственно, а x — m-мерный вектор Париха [**4**]. С помощью уравнения состояния определим S- и T-инварианты сети Петри. Это и даст возможность выявить мертвые переходы, недостижимые позиции и проверить ограниченность сети.

S-инвариант — это линейное отношение на маркировке подмножества позиций, выражающееся в том, что сумма различных меток в позициях положительна. Если в S-инвариант входят все позиции сети Петри, то в построенной модели все позиции являются достижимыми. T-инвариант соответствует последовательности срабатывания переходов, переводящей сеть из маркировки M в ту же самую маркировку M [**5**]. Если T-инвариант включает все переходы сети, то она *жива*.

Размерность матрицы инцидентности A из (10) для данной сети Петри составляет 10×7 (10 уравнений, 7 неизвестных), а вектора Париха x — 7. В соответствии с табл. 1 матрица инцидентности для данной сети Петри будет иметь вид

$$A = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & -1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 & -1 & 1 & 1 \\ -1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & -1 \end{bmatrix}.$$

Система диофантовых уравнений (2) имеет 2 решения:

$x_1 = \{1, 1, 1, 0, 1, 0, 0\}^T$

$x_2 = \{1, 1, 0, 1, 1, 1, 1\}^T$

Эти решения задают Т-инварианты системы. Поскольку в перечне решений нет ни одного столбца, содержащего только нулевые элементы, то все переходы в сети Петри являются живыми при данной начальной разметке, т.е. каждый переход в сети срабатывает хотя бы один раз.

Сгенерируем S-инварианты данной сети Петри. Для этого найдем решения системы уравнений

$$A^T y = 0, \qquad (3)$$

где A — целочисленная n×m матрица инцидентности сети Петри, т — символ транспонирования матрицы, y — n-мерный вектор.

Система диофантовых уравнений (3), содержащая 7 уравнений и 10 неизвестных имеет следующие решения.

$y_1 = \{0, 0, 0, 0, 0, 1, 1, 0, 0, 0\}^T$

$y_2 = \{0, 0, 1, 1, 1, 0, 0, 0, 0, 0\}^T$

$y_3 = \{0, 1, 0, 1, 1, 1, 0, 0, 0, 0\}^T$

$y_4 = \{1, 0, 0, 1, 1, 1, 0, 0, 1, 0\}^T$

$y_5 = \{0, 0, 0, 0, 0, 0, 1, 1, 0, 1\}^T$

$y_6 = \{0, 1, 0, 1, 1, 0, 0, 1, 0, 1\}^T$

$y_7 = \{1, 0, 0, 1, 1, 1, 0, 0, 1, 0\}^T$

Решения системы уравнений (3) определяют S-инварианты сети Петри. Из перечня решений следует, что сеть Петри является *ограниченной*, поскольку все позиции в ней соответствуют положительным инвариантам. Это означает, что ни в одной позиции сети не может скапливаться бесконечное число фишек.

Для получения множества S- и T-инвариантов сети Петри используется TSS-алгоритм [**6**, **7**], который позволяет построить минимальную порождающую систему решений однородной системы линейных диофантовых уравнений над множеством натуральных чисел N.

Исследование выполнимости свойств взаимного исключения и равноправия

Для исследования выполнимости свойств взаимного исключения (`mutex`) и равноправия (`fairness`) необходимо построить *транзиционную систему* [**8**] или *граф достижимых разметок* сети Петри .

Для начальной разметки, определяемой соотношением

M_0=(proc, want, query, block, free, busy, wait, work)=(4, 0, 3, 1, 1, 0, 0, 0),

Транзиционная система сети Петри, показанной на рис 2, имеет вид, представленный на рис. 3.

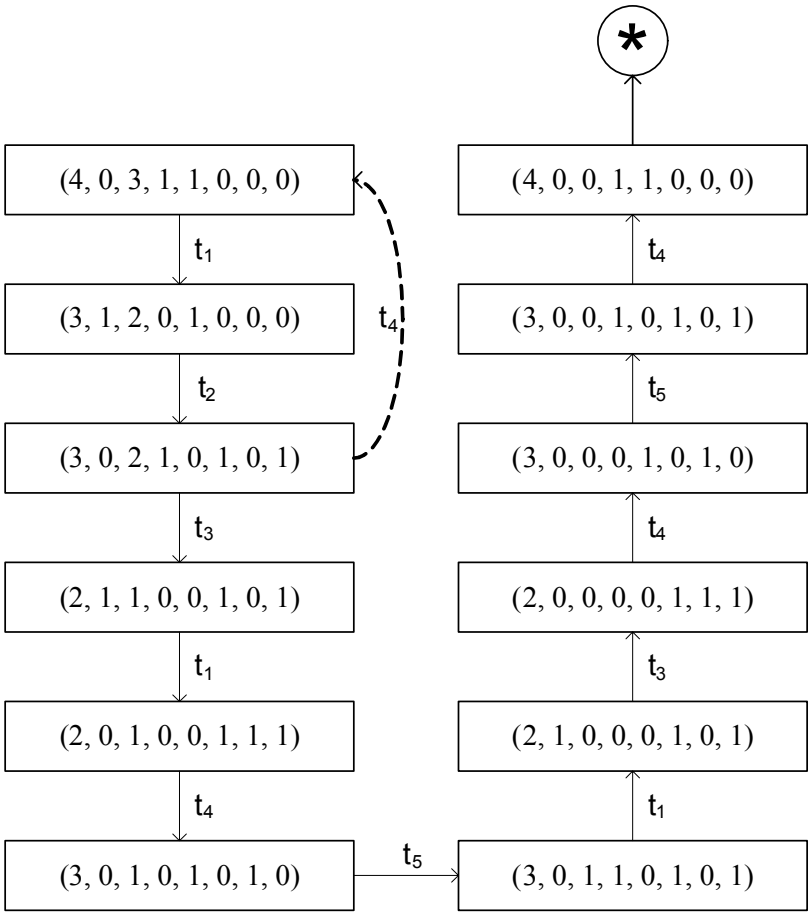


Рис. 3. Транзиционная система сети Петри, моделирующей работу узла Grid-системы

Простой анализ полученной транзиционной системы показывает, что свойство взаимного исключения (mutex) выполняется, поскольку в позиции work во время функционирования сети Петри появляется не более одной фишки. Это же справедливо и для позиции wait. Число фишек в этих позициях в транзиционной системе на рис. 3 задается двумя последними элементами векторов разметки.

Свойство равноправия (fairness) тоже, очевидно, выполняется, поскольку при появлении активного процессора (1 в позиции want) в разметке μ из этой разметки достижима разметка μ`, при которой в позиции want находится 0 фишек. А это означает, что процессор рано или поздно получит доступ к соответствующему ресурсу (памяти, данным и т.п.).

Литература

1. Krauter K., Buyya R., Maheswaran M. A Taxonomy and Survey of GRID Resource Management Systems and Distributed Computing // Software-Practice and Experience, John Wiley & Sons, Ltd. — 2001. — P. 1–10.
2. Куссуль Н.Н., Шелестов А.Ю., Лобунец А.Г. Применение методов операционного анализа для оценки производительности GRID-систем // Кибернетика и вычислительная техника. – 2004. – Выпуск 144. – С. 3-20.
3. Дж. Питерсон Теория сетей Петри и моделирование систем. — М.: Мир, 1984. — 264 с.
4. Parikh R. On Context-Free Languages// J. of the ACM, 13, # 4, 1966. — PP. 570-581.
5. Дубинин В.Н., Зинкин С.А. Языки логического программирования в проектировании вычислительных систем и сетей: Учеб. пособие. — Пенза: Изд-во Пенз. гос. техн. ун-та, 1997. — 100 с.
6. Кривой С.Л. Критерий совместности систем линейных диофантовых уравнений над множеством натуральных чисел// Доповіді НАНУ. — 1999. — № 5. — С. 107-112.
7. Krivoi S. A criteria of Compatibility Systems of Linear Diophantine Constraints // Lecture Notes in Comp. Science. — 2002. — № 2328. — С. 264-271.
8. Летичевський О.А. Сучасні проблеми кібернетики. Нормативний курс. Навчальна електронна бібліотека факультету кібернетики Київського національного університету ім. Тараса Шевченка// http://www.unicyb.kiev.ua/Library/

Получено 21.05.06

УДК 52(15).003

МОДЕЛИРОВАНИЕ И СОЗДАНИЕ РАСПРЕДЕЛЕННЫХ СИСТЕМ ВИЗУАЛИЗАЦИИ ГЕОПРОСТРАНСТВЕННЫХ РЕЗУЛЬТАТОВ ОБРАБОТКИ СПУТНИКОВЫХ ДАННЫХ

Н.И. Ильин, А.Н. Кравченко
Институт космических исследований НАН и НКА Украины

*В даній роботі запропоновано підхід до моделювання Web-орієнтованих розподілених систем візуалізації геопросторових даних [**1**], наведено модель продуктивності та розглянуто реалізацію запропонованого підходу в сервісах екологічного моніторингу Інституту космічних досліджень НАНУ-НКАУ.*

1. *Предусловия возникновения распределенных систем визуализации*

Прогресс в области использования спутниковых данных для ряда задач, в частности мониторинга экологических параметров окружающей среды, наблюдения и прогнозирования развития природных и техногенных катастроф, привел к возникновению необходимости в эффективных и недорогих специализированных системах визуализации геопространственных данных – информации о некоторых пространственных характеристиках земной поверхности и околоземного пространства.

На текущем этапе развития было создано ряд специализированных сервисов визуализации и предоставления данных, преимущественно web-ориентированных [**2**]. Уточнение существующих и создание новых моделей на основе интеграции результатов этих сервисов привели к возникновению проблем стандартизации существующих систем, часто настолько серъезным, что сервисы мониторинга приходится разрабатывать заново, выбрасывая предыдущие разработки.

Сложность обработки и большие объемы обрабатываемых данных привели к возникновению распределенных систем визуализации и распределенных каталогов данных, близких по своей структуре к Grid-системам. Очевидность стандартизации интерфейсов в таких системах не вызывает сомнений, что оправдывает деятельность в области выработки стандартов предоставления и визуализации. Наиболее успешными на сегодня являются стандарты WMS [**3**], WFS [**4**] и WCS [**5**] консорциума OGC, принятые в международной системе GEOSS.

Несмотря на прогресс в области реализации стандартов и построения конечных систем визуализации, оценка и предсказание параметров производительности таких систем все еще остается в большинстве случаев предметом экспериментального исследования уже реализованной системы. Принимая во внимание стоимость аппаратных и программных средств, ошибки проектирования как результат отсутствия методов оценки рабочих характеристик разрабатываемой системы визуализации приводит к значительным финансовым потерям. Поэтому исследования в области моделирования и оценки производительности систем визуализации имеют, кроме научного, большое прикладное значение.

2. *Подход к моделированию и модель производительности Web-ориентированной распределенной системы визуализации*

Основными характеристиками (метриками) производительности являются: 1. Время отклика (response time), 2. Пропускная способность (throughput), 3. Готовность (availability), 4. Стоимость (cost).

Предлагаемый подход заключается в построении сети массового обслуживания на основе представления компонентов визуализации как систем массового обслуживания с последующим применением методов операционного анализа для получения оценок граничных значений времени отклика и пропускной способности всей системы на основе экспериментальных усредненных характеристик представляемых компонент.

Рассмотрим систему из $k = \overline{1,n}$ ресурсов. Каждый ресурс характеризуется следующими параметрами:

X_k – средняя пропускная способность (производительность) k;

S_k – среднее время обработки запросов на ресурсе k за одно посещение;

V_k – среднее число посещений ресурса k;

D_k – среднее время обработки запроса ресурсом k;

U_k – коэффициент использования ресурса k;

R_k – среднее время отклика на запрос ресурса k;

N – среднее число запросов в сети массового обслуживания;

λ – средняя частота поступления запросов к ресурсу сети массового обслуживания.

Z – среднее время ожидания пользователя перед отправкой запроса;

В дальнейших выкладках слово «среднее» будет пропущено. В случае, когда система открыта (число запросов неограничено) в состоянии равновесия ($\lambda = X$) имеет место зависимость:

$$U_k = X_k \times S_k, \quad X_k = \lambda \times V_k \Rightarrow U_k = \lambda \times D_k \le 1$$

для всех k. Поэтому

$$U_{\max} = \lambda \times D_{\max} \le 1,$$

где $D_{\max}$ – необходимость обслуживания на ресурсе с максимальным временем обслуживания. Введем параметр частоты насыщения запросов:

$$\lambda_{sat} = \frac{1}{D_{\max}}.$$

Время отклика для открытой системы:

- Задержки при поступлении запросов отсутствуют:

$$D = \sum_k D_k \Rightarrow R = D;$$

- n запросов поступают одновременно каждые $\frac{n}{\lambda}$ единиц времени.

В случае закрытых систем (количество запросов в системе ограничено) предельные значения метрик производительности:

для больших нагрузок:

$$U_k(N) = X(N) \times D_k \le 1, \forall k \Rightarrow X(N) \le \frac{1}{D_{\max}};$$

для малых нагрузок:

$$X = \frac{1}{D+Z}.$$

Асимптоты в случае отсутствия задержек:

- для каждого запроса $X = \frac{1}{D+Z}$;
- для N запросов $X = \frac{N}{D+Z}$ (верхняя граница).

Асимптоты в случае, когда каждый запрос задерживается другими $N-1$ запросами:

- суммарное время пребывание в системе $(N-1)D + D = ND$;
- суммарное время между обработкой каждого запроса $ND + Z$;
- пропускная способность для одного запроса $\frac{1}{ND+Z}$;
- нижний предел $X(N) = \frac{N}{ND+Z}$.

Точка пересечения верхней и нижней асимптот называется точкой насыщения. После точки насыщения значение производительности остается постоянным и не зависит от числа запросов:

$$\frac{1}{D_{\max}} = \frac{N^*}{D+Z} \Rightarrow N^* = \frac{D+Z}{D_{\max}}.$$

В общем случае, не детализируя классы запросов, для пропускной способности X справедливо

$$\frac{N}{ND+Z} \le X(N) \le \min\left(\frac{1}{D_{\max}}, \frac{N}{D+Z}\right).$$

Для времени отклика R:

$$\max\left(D_{\max}, \frac{D+Z}{N}\right) \le \frac{R(N)+Z}{N} \le \frac{ND+Z}{N}, \quad \max\left(ND_{\max} - Z, D\right) \le R(N) \le ND.$$

Таким образом можно получить граничные значения производительности компонент системы визуализации, а также оценить производительность всей инфраструктуры в целом.

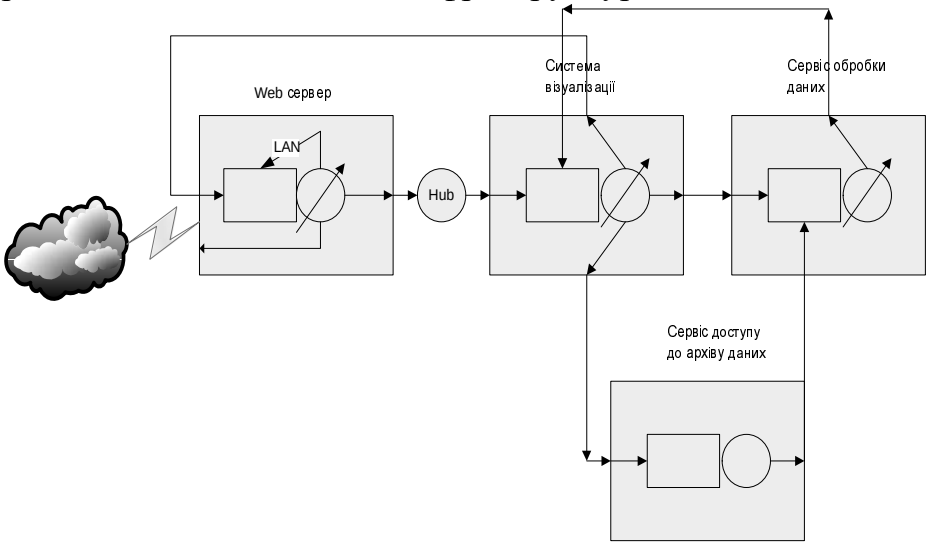


Рис. 1. Система визуализации как сеть массового обслуживания.

Рассмотрим применение предлагаемого подхода на примере системы визуализации, разрабатываемой Институтом космических исследований НАНУ-НКАУ. Компоненты системы визуализации могут быть представлены как системы массового обслуживания с задержками, не зависящими от нагрузки (рис.1).

На основе построенной модели была реализована система визуализации для ряда сервисов экологического мониторинга. Было проведено тестирование общей производительности системы, полученные результаты совпадают с модельными для более чем 80% случаев. Полученная точность является высокой по сравнению с

результатами исследований методами, применяемыми для оценки производительности Web-систем.

3. Примеры реализации

Рассмотренная в предыдущем разделе система визуализации геопространственных данных нашла применение в сервисах оценки биоразнообразия в Причерноморском регионе [**6**] и мониторинга водных ресурсов в Днепровском лимане [**7**].

При реализации сервисов использовано бесплатное программное обеспечение с открытым кодом UMN MapServer 4.10 (визуализация геопространственных данных, поддержка стандартов предоставления WMS и WCS), CartoWeb 3.3 (интерфейс пользователя, балансировка нагрузки, разграничение доступа, интернационализация), Apache 1.34+mod_ssl (web-сервер, разграничение доступа, поддержка защищенного канала передачи). В качестве базовой операционной системы при разработке сервисов, реализации серверов доступа и обработки данных использована Gentoo Linux. Реализованная система не требует дополнительных затрат на лицензирование или закупку специализированного ПО.

Выводы. В данной работе предложен подход к моделированию распределенных Web-ориентированных систем визуализации геопространственных данных. Построенная на основе предлагаемого подхода модель для системы визуализации ИКИ НАНУ-НКАУ при верификации на реализованной системе показывает точность более 80%, что является очень хорошим результатом по сравнению с существующими подходами для Web-сайтов. Полученная модель использована при построении системы визуализации для систем мониторинга экологических параметров окружающей среды на основе спутниковых данных.

Список литературы

1. Shelestov A., Kravchenko O., Ilin M. Geospatial data visualisation in Grid system on Ukrainian segment of GEOSS/GMES// Proc. of the V-th International Conference “Information Research&Applications”. — Varna (Bulgaria). — June 26-30, 2007. — Vol. 2. — P. 422-428
2. Кравченко О.М., Шелестов А.Ю. Застосування реалізацій стандартів OGC для створення розподілених систем візуалізації та надання геопросторових даних, Проблеми програмування. – 2006. - №2-3. – С. 135-139. ISSN1727-4907
3. WMS 1.1.1 specification, http://www.opengeospatial.org/standards/wms.
4. WFS specification, http://www.opengeospatial.org/standards/wfs.
5. WCS specification, http://www.opengeospatial.org/standards/wcs.
6. Сервис оценки биоразнообразия в Южно-Причерноморском регионе, www.dos.ikd.kiev.ua.
7. Сервис мониторинга водных ресурсов Днепровского лимана, **www.geo-ukraine.org.ua**.

Получено 14.05.07

УДК 629.735.072.8.08

ПРИКЛАД ВИБОРУ АРХІТЕКТУРНИХ РІШЕНЬ ПІДСИСТЕМИ РОБОТИ З ВІЗУАЛЬНИМИ МОДЕЛЯМИ

К.О. Холод, С.В. Кузнецов

Національний авіаційний університет, Інститут комп’ютерних технологій

У статті розглянуті підходи до вибору архітектурних рішень підсистеми роботи з візуальними моделями на прикладі генератору каналу комп’ютерного генератору зображення імітатора візуальної обстановки комплексного авіційного тренажеру.

Для забезпечення більшої безпеки польотів, а також перекваліфікації пілотів, ICAO було передбачено поряд з виробництвом повітряного судна розробляти і його комплексний авіаційний тренажер, який забезпечує можливість об'єктивної оцінки якості підготовки льотчиків і екіпажа у всіх режимах, у тому числі попередньому польотові, а також у режимах позаштатних ситуацій і при виконанні бойових задач. Він являє собою відтворений у реальному масштабі макет кабіни повітряного судна конкретного типу або конструкції, конкретної моделі або серії. Включає: комплект обладнання та програмне забезпечення, необхідне для відтворення умов експлуатації повітряного судна на землі та в повітрі; систему візуалізації, що відображає позакабінний простір; систему імітації акселераційних впливів.

Комп’ютерний генератор зображення (КГЗ) – компонент імітатора візуальної обстановки, що призначений для побудови та відображення на пристрій індикації візуальної обстановки поза кабіною літака у реальному часі динаміки польоту. Сучасний КГЗ складається з імітатору та серверу повітряного руху, серверу баз даних середовища (СБДО відповідає за моделювання та передачу генераторам каналів об’єктивної інформації про стан атмосфери, земної поверхні, аеродромах та інших літаків в повітрі на землі) та одного чи декількох генераторів каналів, що відображують на пристрої індикації певну частину простору поза кабіною літака. Усі імітатори тренажеру обмінюються інформацією та командами через виделену.

В багатоканальних КГЗ задачі окремого генератору каналу (ГК) істотно відрізняються від одно канальних КГЗ в першу чергу обмеженням спектру задач, які вони вирішують. До цих задачі відносяться: отримання та інтерпретація потоку керуючих команд та даних опису тривимірних просторових моделей, відповідна

модифікація та підготовка передачі до графічного конвеєру графу сцени, підсистема виводу до графічного конвеєра.

Сформульовані наступні загальні вимоги до архітектури:

- мінімальна складність та простота супроводу;
- слабке сполучення, що забезпечує можливість підміни реалізацій інтерпретаторів графа моделі;
- можливість повторного використання;
- кроссплатформенність;
- стратифікація.

Також виділені наступні специфічні вимоги: параметризація керування генерацією й інтерпретацією моделі (це необхідно, наприклад, для моделей небесної сфери й лінії обрію); єдине програмне ядро для генерації моделі і її наступної інтерпретації.

Проектування архітектури здійснюється на основі сформульованої системи вимог з використанням системного підходу, дозволяє вирішити цілий ряд завдань, головним з яких є [1]:

- декомпозиція на агрегати й модулі;
- виявлення рівнів ієрархії;
- дослідження й виявлення системоутворюючих властивостей компонентів та їх взаємні зв'язки на різних рівнях;
- вибір принципів побудови різних конфігурацій залежно від цільової функції.

Реалізовані рішення етапу проектування:

- декомпозиція на основі сформульованої системи вимог;
- виявлення на основі формальних вимог, забезпечуваного якості моделювання й ступені формалізації опису існуючих моделей даних й алгоритмів які доцільно повторно використати, а не розробляти заново;
- вибір методу програмної реалізації модуля (об‘єктно-орієнтований підхід, структурне програмування) на основі ступеня формалізації відповідної частини предметної області;
- застосування паттернів проектування для поділу логіки керування й інтерпретації моделі.

Умовно усі компоненти за функціональністю можна розподілити між 6 рівнями: сервісу операційної системи; системний; моделей; інтерпретації моделей; програмний; комунікаційний.

На кожному з рівнів компоненти за призначенням розподілені по таких ланках: ядро, модулі та допоміжні.

На рис. 1. представлена схема розподілу КГЗ на компоненти та зв’язки між ними.

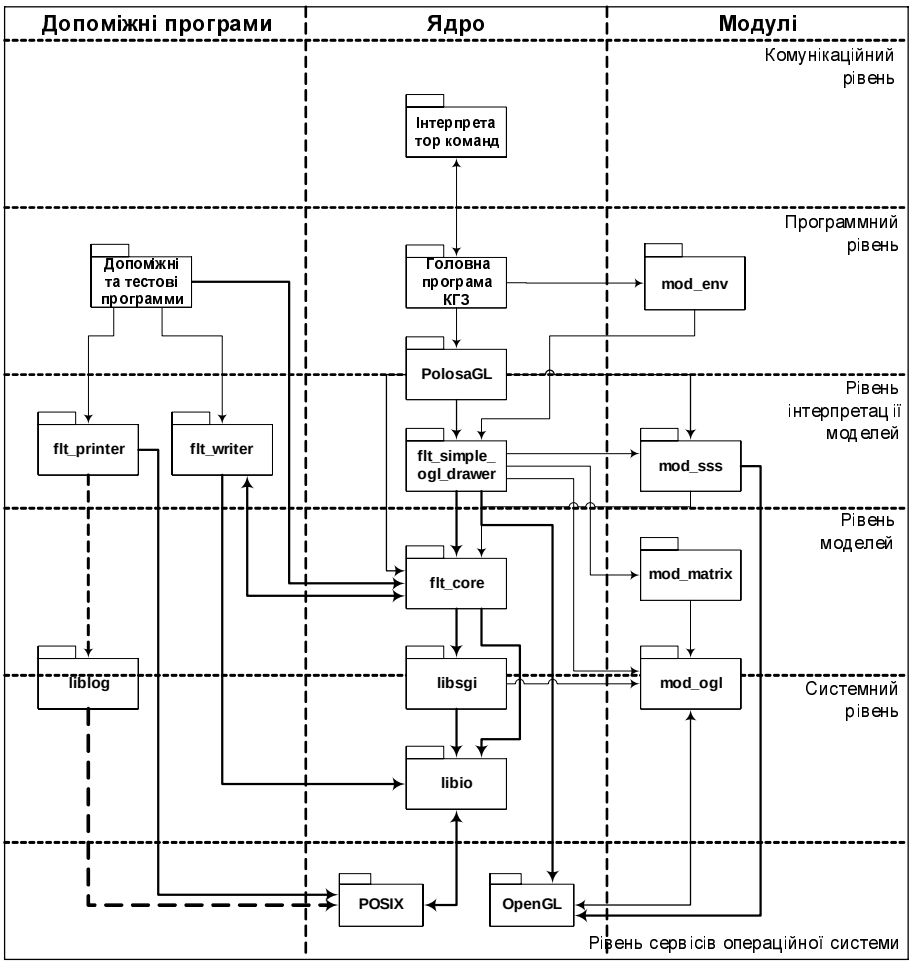


Рис. 1. Архітектура каналу КГЗ

До компонентів рівня сервісу операційної системи компоненти, що відповідають за взаємодію КГЗ з апаратними пристроями через інтерфейси операційної системи. Компоненти системного рівня використовують можливості компонентів нижнього рівня для надання вищим рівням абстрактних інтерфейсів.

mod_OGL – модуль, що накопичує частину параметрів графічного конвеєру, та забезпечує ефективну роботу з ними. **libIO** – бібліотека вводу/виводу, забезпечує платформо-незалежну роботу з бінарними файлами. **libSGI** – бібліотека роботи з текстурами у форматі Silicon Graphics. Формат підтримує стиснення даних та можливість організації багаторівневих текстур. **libLog** – допоміжна бібліотека для організації гнучкого ведення протоколу робити програм.

Компоненти представлення моделей. Основним форматом для представлення полігональних моделей обраний формат OpenFlight (http://multigen.com), що є індустріальним стандартом для промислових систем візуалізації. OpenFlight підтримує: змінні рівні деталізації, обмеження ступенів свободи, звуки, посилання (внутрішні та

зовнішні), реплікацію, анімацію, обмежуючи об’єми, освітлення, вогні та стрічки вогнів.

flt_core – бібліотека, що містить класи які представляють типи вузлів бази даних OpenFlight. Для представлення моделей у форматі OpenFlight використовується паттерн Композит[3]. До складу бібліотеки входить пул-синглетон, який накопичує завантажені моделі та дозволяє керувати ними.

mod_matrix – модуль, що забезпечує роботу з матрицями перетворень. Виконує наступні функції: завантажує початкові матриці з графічного конвеєра, накопичує стеки видових, текстурних, та проекційних матриць перетворень, виконує над матрицями операції переносу, повороту, масштабу та множення, перераховує координати для заданої точки в контексті поточного стану стеку матриць перетворень.

Компоненти рівня інтерпретації моделей. Для обробки графів моделей використовується паттерн Відвідувач [3]. Класи-відвідувачі дозволяють відділити операції на моделями та самих моделей. Виділені два типи класів-відвідувачів: основні та допоміжні. Основні класи-відвідувачі відповідають за відображення моделей. Допоміжні – за виконання над графами-моделями різноманітних операцій по збору та аналізу інформації.

flt_simple_ogl_drawer – бібліотека, що містить основний клас-відвідувач, який за допомогою бібліотеки OpenGL відображує модель.

Знаючи за допомогою інформації з модуля mod_env поточне положення у просторі, відсікає відображення невидимих ділянок моделі та обирає необхідний рівень деталізації для частин графу моделі, які мають альтернативні представлення в залежності від відстані до них. Клас-відвідувач збирає і накопичує інформацію про вогні для подальшого їх відображення у mod_sss. Про кожний з вогнів зберігається: координати у світовій системі координат, видові координати, координати в одиничному кубі, колір на передньому плані, колір на задньому плані. Для розрахунку координат вогнів у видовій системі координат та в одиничному кубі використовується інформація, що накопичується в модулі mod_matrix.

flt_writer – бібліотека, що містить клас-відвідувач, який зберігає задану FLT-модель на диску.

mod_sss – модуль, що відповідає за відображення вогнів (наприклад, вогнів світлосигнальних систем).

Компоненти програмного рівня.

PolosaGL – бібліотека, що містить фасад[3] для доступу до моделлей OpenFlight та їх відображенням за допомогою класу-

відвідувача з бібліотеки flt_simple_ogl_drawer. Використовується головною програмою КГЗ.

Головна програма КГЗ встановлює зв'язок з графічним прискорювачем, завантажує та починає роботу модулів, організовує середовище для зв'язку с інтерпретатором команд, організує відображення сцени.

Допоміжні програми – програми підтримки процесу створення сцен. Прикладом таких програм є генератори ландшафтів та місцевості, параметричних аеродромів, параметричних світлосигнальних систем, рівнів деталізації моделей.

mod_env – модуль, що накопичує інформацію про стан середовища моделювання. Мінімальний набір складає інформація про положення глядача в просторі та поточний час. Інформація оновлюється з головної програми КГЗ.

Компоненти комунікаційного рівня. Інтерпретатор команд – компонент, що приймає із мережі команди та виконує їх над внутрішніми об'єктами КГЗ. Також на ньому можуть бути програми-адаптери для протоколів Глорія, DIS, HLA та інших. Це дозволяє адаптувати КГЗ для сумісної роботи з різноманітними моделюючими комплексами та використовувати його в різних прикладних областях.

Для реалізації вимог що поставлені до системи візуалізації (в основному до пілотажних тренажерів) необхідні дослідження та розробка моделей в таких областях як моделювання поверхні землі, атмосферних явищ, лінії горизонту, вогнів та вогнів світлосигнальних систем, методах корекції перспективних викривлень.

Кожна з приведених областей погано підлягає формалізації. Для того щоб дослідники мали змогу ефективно проводити експерименти з моделями в цих областях цикл апробації моделей повинен бути зменшений до мінімуму. Відкрита архітектура КГЗ сприяє цьому найкращим чином.

Список літератури

1. *Моржов В.И.* Научно-технические методы построения моделирующей среды, обеспечивающей потребные эксплуатационно-технические характеристики авиационного тренажера. Диссертация на соискание ученой степени доктора технических наук. К: КИИГА, 1990 год, 352 стр.

2. *Alfred T. Lee*, Flight Simulation: Virtual Environments In Aviation. Ashgate Publishing, 2005, 134 pages.

3. *Гамма Э., Хелм Р., Джонсон Р., Влиссидес Дж.* Приемы объектно-ориентированного проектирования. Паттерны проектирования. — СПб: Питер, 2001. — 368 с.: ил. (Серия «Библиотека программиста»)

Дата надходження до редколегії: 15 травня 2007 року

УДК 681.327

ПОСТРОЕНИЕ ПОВЕРХНОСТИ ОБЪЕКТОВ СЛОЖНОЙ ФОРМЫ ПО НАБОРУ ТОЧЕК ДЛЯ ВИЗУАЛИЗАЦИИ И МОДЕЛИРОВАНИЯ

Башков Е.А., Пауков Д.П.
Донецкий национальный технический университет

Йдеться про побудову полігонального подання складних поверхонь, що будуються за набором тривимірових точок. Представлена модифікація метода, що використовує розділення простору на осередки та допускає паралельне обчислення.

Построение поверхности объектов сложной формы – задача актуальная для многих отраслей науки и техники, медицины, истории и культуры. Моделирование и визуализация сложных поверхностей лежит в основе трехмерной компьютерной томографии [1], визуализации результатов научных экспериментов [2]. Построение сложных поверхностей часто встречается при проектировании различных механизмов в машиностроении, в строительстве и архитектуре [3]. Многие музеи используют построение сложных поверхностей, как для создания цифровых копий скульптур, так и для повышения качества реставрационных работ [4].

Первые публикации, посвященные проблеме построения поверхностей по набору точек, появились сравнительно недавно: в 1990-х годах [5-6]. За это время было предложено несколько подходов в решении этой задачи: методы, использующие триангуляцию Делоне [7] в качестве первоначальной модели поверхности с последующей обработкой [8-9], использующие пространственное разбиение [5], использующие функцию расстояния до искомой поверхности [10-11], а также использующие техники деформации [12].

Задача построения поверхности по набору точек состоит в следующем. Имеется набор трехмерных точек, описывающих поверхность (замкнутую или разомкнутую). Используя только эти сведения, а также некоторую величину среднего расстояния между соседними точками, необходимо получить полигональное представление поверхности, проходящей вблизи исходных точек.

В этой работе рассматривается модификация метода построения полигонального представления поверхности по набору точек, основанного на пространственном разбиении и функции знакового

расстояния до искомой поверхности [5-6, 13]. В качестве элемента полигональной сетки используется треугольник – как наиболее простая геометрическая фигура, позволяющая эффективно визуализировать поверхность. Вершины треугольников (узлы сетки) вычисляются с помощью метода marching cubes [6]. Размер ребра куба задается дополнительно и определяет размеры треугольников. Первоначальные плоскости вычисляются путем отбора для каждой исходной точки множества ближайших точек, при этом используется заданная величина среднего расстояния между ними. В случае, когда эта величина не задана, её можно подобрать экспериментально.

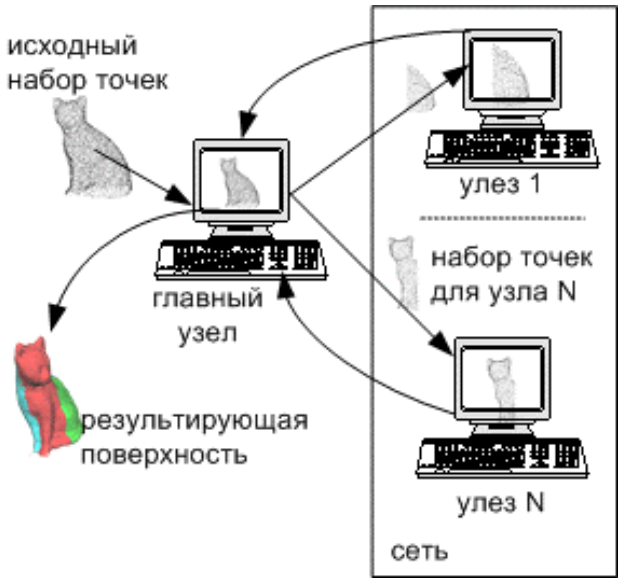


Рис. 1 – Схема построения поверхности по набору точек с помощью N вычислительных узлов.

При выполнении построения сложной поверхности по набору точек на одном вычислительном устройстве возникает проблема недостатка ресурсов памяти, так как сложные поверхности описываются большим количеством точек, и как следствием высокими временными затратами. Решить эту проблему можно путем разделения исходных точек на части, которые могут быть обработаны независимо.

На рис. 1 показана схема метода построения поверхности путем разделения вычислений между несколькими вычислительными узлами работающими параллельно. В качестве параллельных вычислительных узлов может быть использован кластер или многопроцессорная вычислительная машина. Исследуемое пространство разделяется на параллелепипеды в зависимости от количества имеющихся вычислительных узлов. Каждому

вычислительному узлу ставиться в соответствие полученный параллелепипед. Исходные точки разделяются на группы, принадлежащие пространству, заключенному в параллелепипеде, и являются исходными данными построения части поверхности для соответствующего вычислительного узла. После окончания работы всех вычислительных узлов, полученные части поверхности объединяются в единую полигональную модель [10].

Таблица 1 – Результаты построения сеточной модели

№	Средн. расст., ρ	Ребро решет-ки, e	Кол-во точек	Кол-во треуголь-ников	Первон. построение плоскостей, мс	Выбор направления нормалей, мс	Построение сетки, мс	Общее время, мс
стопа человека								
1	5.5	3.0	1 362	7 568	290	1 512	32 446	34 248
2	5.6	3.5	1 257	4 665	210	1 082	16 134	17 426
3	5.5	3.0	1 432	7 552	300	1 552	36 012	37 864
4	6.2	5.0	1 041	1 753	30	982	4 436	5 448
Σ	-	-	5 092	21 538	300	1 552	36 012	37 864
Ω	5.5	3.0	5 092	24 720	480	9 734	256 459	266 673
кошка (4 части)								
1	0.018	0.018	1 763	1 626	60	1 783	5 508	7 351
2	0.018	0.018	3 137	2 744	120	6 749	13 880	20 749
3	0.018	0.018	1 989	1 861	50	2 233	6 529	8 812
4	0.018	0.018	3 111	2 705	80	5 358	12 117	17 555
Σ	-	-	10000	8 936	120	6 749	13 880	20 749
Ω	0.018	0.018	10000	8 719	290	35 451	99 705	135 446
кошка (8 частей)								
1	0.019	0.018	1 599	1 427	66	1 748	4 824	6 638
2	0.019	0.018	164	181	3	89	116	208
3	0.019	0.018	1 323	1 091	31	1 749	2 892	4 671
4	0.019	0.018	1 814	1 553	33	2 376	4 594	7 003
5	0.019	0.018	1 575	1 396	31	1 748	4 243	6 023
6	0.019	0.018	414	436	33	270	423	726
7	0.019	0.018	1 598	1 367	33	1 794	3 705	5 532
8	0.019	0.018	1 513	1 320	33	1 571	3 552	5 156
Σ	-	-	10000	8 771	66	2 376	4 824	7 003

Рассматриваемый метод построения сетки на практике продемонстрировал хорошие результаты. В табл. 1 представлены временные характеристики построения сетки для модели стопы человека и модели кошки, выполненные как на одном вычислительном узле, так и на нескольких узлах. Символом Σ в таблице отмечены результирующие показатели характеристик построения сетки на нескольких узлах. Временные характеристики в этом случае определяются как наибольшее значение времени работы

любого из узлов. Символ Ω означает построение поверхности без разделения исходных точек на части и соответственно без привлечения дополнительных вычислительных узлов. Экспериментальные данный получены на ЭВМ с процессором Intel Pentium III 733 МГц, ОЗУ 512 МВ.

Представленная на рис. 2 сравнительная диаграмма временных затрат демонстрирует, что параллельная обработка частей исходных точек может существенно улучшить временные характеристики построения сложных поверхностей и соответственно повысить сложность обрабатываемых поверхностей.

Рис. 2 – Сравнительная диаграмма времени построения поверхностей в зависимости от количества вычислительных узлов

На рис. 3 показаны три модели поверхности кошки, полученные построением на одной вычислительной машине и разделенные на 4 и 8 частей закрашенные различным цветом. На рис. 4 показаны две модели поверхности стопы человека, полученные построением на одной вычислительной машине и разделенные на 4 части закрашенные различным цветом.

Анализ экспериментальных данных из табл. 1 и рис. 2-4 показывает, что параллельная обработка исходных точек при построении поверхности сложной формы существенно снижает временные затраты: использование 4-х вычислителей параллельно вместо одного уменьшает время построения более чем в 6 раз, 8 вычислительных узлов позволяют снизить время построения поверхности более чем в 15 раз. В таблице не учитывалось время

разделения набора исходных точек на группы и время объединения частей поверхностей в единую модель, т.к. для представленных случаев эти показатели несущественны.

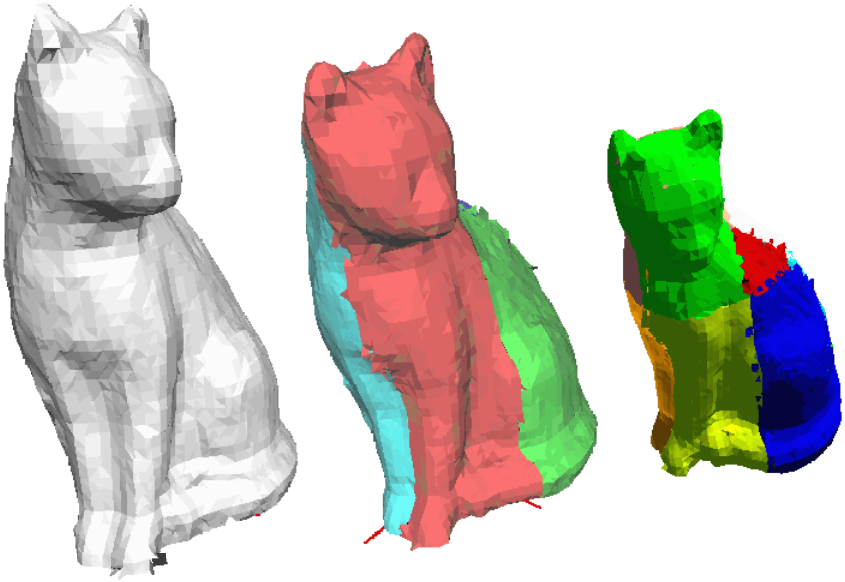


Рис. 3 – Поверхность кошки, построенная по единому набору точек (слева), разделенная на 4 части (в центре) и на 8 частей (справа)

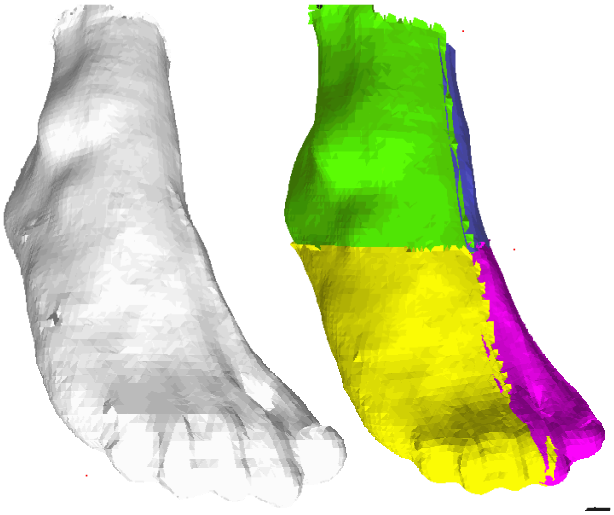


Рис. 4 – Поверхность стопы человека, построенная по единому набору точек (слева) и разделенная на 4 части (справа)

Рассматриваемый метод позволяет существенно повысить сложность моделируемой поверхности, т.к. сложные поверхности представляются большим набором точек, который практически невозможно обработать на одном вычислителе. При этом визуальное

качество результирующей поверхности, полученной рассматриваемым методом, остается примерно таким же, как и поверхности, полученной при обработке сразу всех исходных точек.

Однако следует отметить, что вместе с существенным улучшением времени построения полигональной модели были обнаружены следующие проблемы параллельной обработки данных при построении сеток для сложных поверхностей. Нагрузка различных узлов может существенно отличаться, что влечет за собой простаивание некоторых вычислительных узлов в ожидании завершения работы других узлов. В общем случае сложно выполнить разделение исходных данных на части предназначенные для обработки на узлах так, чтобы все узлы завершили работу одновременно. В общем случае сложно выполнить объединение полученных на каждом узле поверхностей в единую поверхность [10]. В случае если исходные точки располагаются неравномерно, то параметры среднего расстояния между соседними точками для каждого узла могут отличаться, что снижает качество полигональной сетки или требует дополнительного механизма корректировки этого показателя. Дальнейшие исследования необходимо направить на решение вышеуказанных проблем.

Литература

1. X-ray MicroCT examination of bone specimens http://www.tecno.ior.it/VRLAB/researchers/repository/MicroCT/MicroCT_filesInfo.htm
2. Эйнджел Эдвард Интерактивная компьютерная графика. Вводный курс на базе OpenGL, 2 изд.: Пер. с англ. – М.: Издательский дом «Вильямс», 2001. – 592 с.: ил.
3. Официальный сайт ЗАО НПП "НАВГЕОКОМ" http://navgeocom.ru
4. M.Callieri, P.Cignoni, F.Ganovelli and so on “Visualization and 3D Data Processing in the David Restoration” IEEE Computer Graphics and Applications March/April 2004, pp. 16-21.
5. Hoppe H., DeRose T., Duchamp T., McDonald J., Stuetzle W. Surface Reconstruction from Unorganized Points. – University of Washington. – 1992.
6. Lorensen W.E., Cline H.E. Marching cubes: A high resolution 3D surface construction algorithm, Computer Graphics // SIGGRAPH ’87 Proceedings. – 1987. – vol. 21, pp. 163-169.
7. Пауков Д. П. Триангуляция Делоне: итеративные алгоритмы построения триангуляции. // Збірка праць магістрантів Донецького національного технічного університету. Випуск 2. – Донецьк: ДонНТУ. – 2003. – с. 461-469.
8. Cohen-Steiner D., Da F. A Greedy Delaunay Based Surface Reconstruction Algorithm, Research Report. – INRIA. – 2002.

9. Chuan-Chu Kuo, Hong-Tzong Yau A Delaunay-Based Region-Growing Approach to Surface Reconstruction from Unorganized Points. pp. 73-82 2003.
10. Позин А.Г. Использование объёмного метода для восстановления 3D поверхностей // Graphicon proceedings. – 2005.
11. H. Zhao Analysis and visualization of large set of unorganized data points using the distance function, 2003
12. J.Esteve, P.Brunet, A. Vinacua Approximation of a Variable Density Cloud of Points by Shrinking a Discrete Membrane. Computer Graphics. – 2005. - vol. 24, pp. 791-807.
13. Хромова Е.Н., Пауков Д.П., Башков Е.А. Воссоздание поверхности по произвольному набору точек. Подзадача построения плоскости, наименее удаленной от совокупности точек. // Збірка доповідей II республіканської наукової конференції студентів, аспірантів та молодих вчених «Комп'ютерний моніторинг та інформаційні технології». – Донецьк: ДонНТУ. – 2006 р.

Дата поступления в редакцию 14.05.2007

УДК 621.833

METHODS AND FACILITIES FOR 3-D OBJECTS SHADING WITH USAGE LOCAL ILLUMINATION MODELS

Oleksandr Romanyuk
Vinnitsa National Technical University, Vinnitsa, Ukraine

The problem of increasing of shading productivity according to widespread methods Gourand and Phong is examined

1. Introduction

The most complicated and resource-intensive computing at the rendering stage takes place in the shading of 3-D objects, which have voluminosity effect created by color gradation.

The color intensity and coordinates are detected for every pixel on the surface in the process of shading. Taking into consideration, that high resolution displays are used for representing realistic images, shading process takes a lot of time. Especially it reveals when illumination models are used, in particular that which represent specular constituent of the color. That's why the question of the increasing shading production in computer graphic systems is very urgent one.

Gourand and Phong methods are the most widespread for today. This is caused by their relative easiness, satisfactory quality and hardware supply ability. This work proposes new approaches of their realization that allows considerable increase production of them.

2. The methods of increasing of shading productivity

This is theoretically proved [1], that during the shading process according Gourand growth of colour intensities $\Delta I_Г$ and ΔI_B correspondingly along horizontal and vertical rasterization lines are constant values and this eliminates necessity of their computing for every rasterization line. Let's find out interconnection between growth of colour intensities along horizontal and vertical rasterization lines. It's proved that growth of intensities ΔI_H, ΔI_V and ΔI_D in horizontal, vertical and diagonal directions correspondingly are constants and are equal

$$\Delta I_Г = \frac{(I_A - I_C)\cdot \Delta Y_{BC} - (I_B - I_C)\cdot \Delta Y_{AC}}{\Delta X_{AC}\cdot \Delta Y_{BC} - \Delta X_{BC}\cdot \Delta Y_{AC}},$$

$$\Delta I_B = \frac{(I_B - I_C)\cdot \Delta Y_{BC} - (I_A - I_C)\cdot \Delta Y_{AC}}{\Delta X_{AC}\cdot \Delta Y_{BC} - \Delta X_{BC}\cdot \Delta Y_{AC}},$$

$$\Delta I_Д = \Delta I_Г + \Delta I_B,$$

where I_A, I_B, I_C -colour intensities in triangle vertexes (Figure 1).

The received property allows to propose sequence of approaches Gourand paralleling rendering [2].

In the first method first order Serpinsky triangulation is used. It lies in dividing initial triangle to 4 parts by drawing centerlines.

With considering the fact that colour intensities growth is permanent value an approach was theoretically founded. It lies in shading only one of rectangular triangles and transformation of rendering results etc. During the shading rasterization of top triangle 1 only is performed and elementary step growth are to be repeated for all the other triangles with considering that triangles 1, 2 and 3 have identical shading directions and triangle 4 is shaded in the oncoming direction (Figure 2). Before rasterization coordinates of all the rectangular triangle vertexes are to be defined.

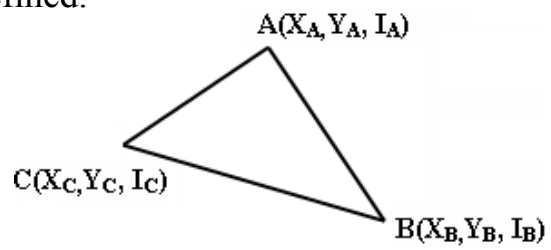


Figure 1. Initial triangle

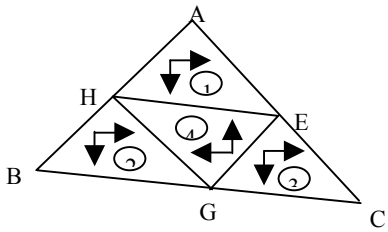


Figure 2. Shading directions of rectangular triangles

Analogously during detection of colour intensities of internal points belonging to the top left rectangular triangle all the actions of code interpolation are to be duplicated for all the other triangles, but according to their vertexes.

Inasmuch initial triangle triangulation is performed by the bisection its edges and growth of initial triangle sides are not always even numbers the task of dividing triangle without non-rasterized points appears. Different divisions' variants were examined and optimal ones were defined.

The second method of Gourand shading production increase is based on the independent forming of even and odd points in the rasterization line. In the preparation cycle colour intensities I_1 , I_2 are to be found according to the first and next points in the rasterization line. The intensities I_K of the first points in scanlines are to be defined by means of

code interpolation of colour intensities along basic edges, and I_2 - with help of expression:

$$I_2 = I_K + \Delta I_\Gamma .$$

$$I_4 = I_2 + 2\Delta I_P, \quad I_6 = I_4 + 2\Delta I_P, \ldots,$$

$$I_{2w} = I_{2w-2} + 2\Delta I_P, \ldots;$$

$$I_3 = I_1 + 2\Delta I_P, \ I_5 = I_3 + 2\Delta I_P, \ I_7 = I_5 + 2\Delta I_P, \ldots,$$

$$I_{2w+1} = I_{2w-1} + 2\Delta I_P, \ldots .$$

One of the possible approach for Gourand rendering improvement lies in the detection of highlights in triangle borders and their further forming [3] .

Suppose normal vectors $\vec{N}_0, \vec{N}_1, \vec{N}_2$ are given in the triangle $(x_0, y_0), (x_1, y_1), (x_2, y_2)$ edges. In the graphic images forming the most common case [2], is when light source and observer are situated in the infinity, h.i. vector $\vec{L}$ has identical direction for all the triangle points. It was proved [6], that in this case maximal values of colour intensity are at the such edge points that satisfies equations

$$t_1 = \frac{\cos\gamma\cos\varphi - \cos\beta}{(\cos\varphi - 1)(\cos\beta + \cos\gamma)},$$

$$t_2 = \frac{\cos\gamma\cos\phi - \cos\lambda}{(\cos\phi - 1)(\cos\lambda + \cos\gamma)}, \ t_3 = \frac{\cos\beta\cos\vartheta - \cos\lambda}{(\cos\vartheta - 1)(\cos\lambda + \cos\beta)} .$$

Conformity of angles between normal vectors is represented in the Figure 3.

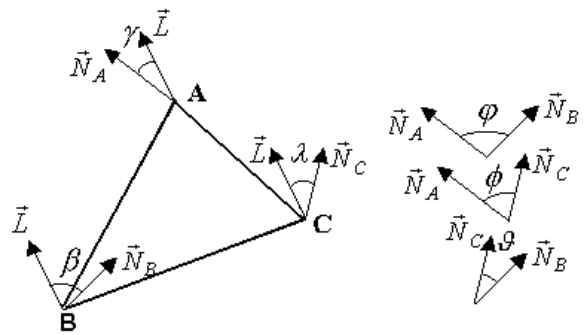


Figure 3. Normal vectors of triangle ABC

Variables t_1, t_2, t_3 are parametric ones and has change interval from 0 to 1 Initial triangle can be divided to several depending on the result of threshold valuations comparing with colour intensities in the points t_1, t_2, t_3 .The triangles received are further shaded with Gourand method. Such an approach increases Gourand rendering quality due to highlights forming in case of their intersection by one or several triangle edges. In

the classic Gourand rasterization realization this shading is absent and highlight is eliminated, that is essential artifact.

In the computer graphics systems Blinn illumination model is widespread, according to what intensity of specular colour constituent is with following formula:

$$I_S = I\, k_S \cos^n \gamma ,$$

where I - intensity of external light source; k_S - coefficient of specular reflection; γ - angle between normal vector $\vec{N}$ and surface normal vector $\vec{H}$, that is received by the adding of light direction vector $\vec{L}$ and observing vector $\vec{V}$; n - coefficient of surface specularity.

The maximal amount of computing in the process of specular colour constituent detection takes place for BRDF computing $\cos^n \gamma$, $n = \overline{1,1000}$. A new BRDF [4] was proposed that has considerably lower power in comparison with such BRDF like $\cos^n \gamma$. It's expressed with formula $cos^k(\sqrt{n/k} \cdot \gamma)$, where k – coefficient, that is detected depending on surface specularity coefficient n ($k << n$).

The usage of various powers k for different diapasons of n values was proposed for guaranteeing of necessary approximation accuracy. For instance in approximation of $cos^n \gamma$ by function $cos^k(\sqrt{n/k} \cdot \gamma)$ with relative error 3% it's enough to take only to values of k: 1, 3. Modification of well known Schlick BRDF was proposed. It lies in using different values of powers depending on specular coefficient. This is the following formula [5]

$$\cos^{\left\lfloor 2^{\log_2 n - 2} \right\rfloor} \gamma / (n - n\cos\gamma + \cos\gamma).$$

A problem of BRDF approximation with function of 2nd and 3rd degree [6, 7] and its hardware realization was examined.

New BRDF approximation, being proposed in this work, represents highlights on the surface of graphic objects with rather high accuracy, at the same time approximating function operates with cosine of angle between vector of normal N and middle vector H. In comparison with Schlick BRDF considerable reduction of approximation error is reached both in highlight epicenter and fading area representation.

With the purpose of increasing the shading productivity it is offered to use in one computing process combined and Phong methods. [8] So, for example, it is possible to divide scanline into digital segments of length, divisible by degree of a two. The values of colour intensities according to the Phong method are determined in endpoints of a segment, and in

intermediate – according to the a Gourand method. The length of the digital segment is able for adaptive changing and is determined by curvature of a surface and direction of vector H.

During the computing specular and diffusive colour constituent for points on 3D surface normalization of normal vectors is required. This is explained by the fact, that cosines of angles that are used in shading function can easily be found by the inner product of unit length vectors. Taking into consideration that normalization is performed for observer's vectors, light source's vector and vector of normal, the problem of computational complexity decreasing for this resource-intensive procedure is urgent. The method of adaptive normal vectors normalization is proposed [9]. The normalization is performed only in cases of considerable influence upon rsult of colour intensity determination.

On the Figure 4 an example of determination of intermediate normal vector between vectors $\vec{N}_a$ and $\vec{N}_b$, that are situated at the scanline endpoints is provided. It's shown, that

$$\frac{d_{(1/2^i)}}{d_{(1/2^{i+1})}} \approx 4 .$$

The following correlation is received for error detection $d_{(1/2^i)}$:

$$d_{(1/2^{i+1})} = 1 - \frac{\sqrt{2 + z_{(1/2)^i}}}{2} .$$

It is proposed at the error $d_{(1/2^i)}$ value lower than certain threshold valuation to use linear interpolation between vectors of normal.

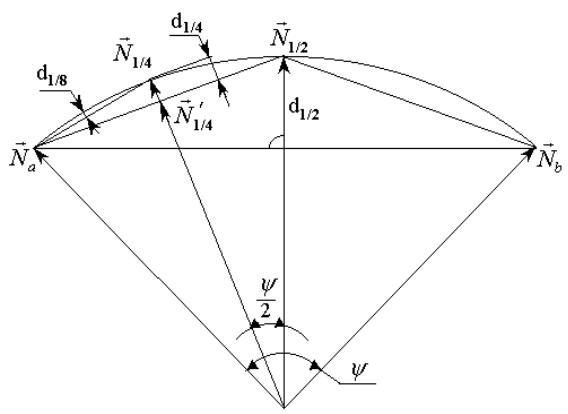


Figure 4. Determination of intermediate vectors of normal

The normal vectors are received by the bisection of angle between given vectors. So, for example,

$$\vec{N}_{(1/2)} = \frac{\vec{N}_a + \vec{N}_b}{\sqrt{2(1+\cos\psi)}} = \frac{\vec{N}_a + \vec{N}_b}{z_{(1/2)}}, \text{ where } z_{(1/2)} = \sqrt{2(1+\cos\psi)}.$$

For the further division following formulas are proposed

$$\vec{N}_{(1/2^{n+1})} = \frac{\vec{N}_a + \vec{N}_{(1/2^n)}}{\sqrt{2 + z_{(1/2^n)}}}, \; z_{(1/2^{n+1})} = \sqrt{2 + z_{(1/2^n)}}.$$

The problem of spherical angular interpolation for the Phong rendering is examined. Recursive correlations are found, according to which normal vector for current point could be computed with help of normal vectors for two previous points, i.e.

$$\vec{N}(t+1) = 2\vec{N}(t)\cos\varphi - \vec{N}(t-1),$$

where φ angle between start and final vector in the scanline. Analogous expression takes place for the determination of diffusive colour constituent

$$I(t+1)_d = 2I(t)_d \cos\varphi - I(t-1)_d$$

The usage of found recursive correlations allows bringing the computation of diffusive colour constituent and inner product to two multiplication operations: two subtraction operations and two shift operations that allows increase productivity to 40%.

The new approach for accelerated diffusive colour constituent determination according to the Phong method is proposed in 3D figures shading. The peculiarity of this approach lies in the synchronous hardware determination of two points at the scanline. For this approach new approximation formula is used for normalization of normal vectors, and new properties of Phong rendering.

The method of accelerated definition of color intensities at shading of 3D graphic objects is proposed. With usage of quadratic interpolation an equation of a surface and color intensities for orthonormal triangle are found according to values of intensities at apexes and at those points on the edges, where color intensities take their extreme values. The obtained results are transferred on a real triangle with usage of affine transformations.

The adaptive method of shading [10], in which two different illumination models are used depending on availability of highlight within the triangle. The increase of productivity is reached due to using in case of availability of highlight within the triangle the illumination model that considers specular colour constituent, and in all the others – simple model, that considerably decreases computation complicity. The method for detecting fact intersection of edge and highlight or its identification within triangle is proposed.

Considering, that highlight on the surface of graphic object occupies a tiny area adaptive approach allows essential increase productivity of the shading.

Not only problems dealt with adaptive usage of illumination models are highlighted but the methods of shading either. At the same time surface curvature is considered and level of specular and diffusive light.

3. Conclusion

The methods being proposed allow considerable increasing of shading productivity, and can be used for designing software and hardware for computer graphics.

4. References

[1] Romanyuk A.N. One of the ways of computation process simplification during the shading of 3D objects according to the Gourand method. //Вимірювальна та обчислювальна техніка в технологічних процесах, №2, 2004, стр. 72-75.

[2] Romanyuk O.N., Chornij A.V. Effective algorithms of rendering procedure parallelization for realistic image forming // Proceedings of the 6th international conference on Development and Application System. - Suceava. – 2002, - pp. 357 - 362.

[3] Romanyuk A.N. The new approach of increasing the quality of 3D objects shading according to the Gourand method // Інформаційні технології та комп'ютерна інженерія, 2005, №3, с. 106-110.

[4] Romanyuk A.N., Chornij A.V. The new approach of specular colour constituent determination // Оптико – електронні інформаційно – енергетичні технології. - 2004.– №2. - С. 85-92.

[5] Romanyuk A.N.., Lyashenko Y.L., Melnikov A.M., Savitska L.A. The new formula for computation BRDF //Вестник Херсонського национального технического университета, №25, 2006. –С.412-416.

[6] Romanyuk O.N., Lyashenko Y.L. Acceleration of painting with guadratic BRDF interpolation // XIII Міжнародна конференція з автоматичного управління (Автоматика -2006). Тези доповідей тринадцятої міжнародної науково-технічної конференції.-Вінниця, УНІВЕРСУМ-Вінниця, 2006. –С. 364.

[7] Romanyuk O.N., Lyashenko Y.L. BRDF approximation with 3rd degree polynomial // Збірник матеріалів п’ятої міжнародної конференції конференції «Інтернет-Освіта-Наука-2006. Том 2.- Вінниця: УНІВЕСУМ-Вінниця, 2006.-С.506-509.

[8]. Romanyuk O.N. The combined usage of binary and code linear interpolation for normalization of normal vectors during the 3D objects shading // Вестник Херсонського национального технического университета, №25, 2006. . –С.408-411.

[9] Romanyuk O.N., Melnikov A.M Adaptive normalization of normal vectors during determination of diffusive and specular colour constituent // Реєстрація, зберігання і обробка даних, 2006, Т. 8, № 3, - С.11-19.

[10] Romanyuk O.N., Chornij A.V. “Methods for Specular Color Component Accelerate Calculation”. IEEE Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications. September 2005,Sofia, Bulgaria, pp. 615-619.

Получено 20.05.07

УДК 004.021::004.92

ИТЕРАЦИОННЫЕ МЕТОДЫ ПОВОРОТА РАСТРОВЫХ ИЗОБРАЖЕНИЙ

В.В. Смолий
Восточноукраинский национальный университет, Северодонецкий технологический институт

У роботі розглянуто питання підвищення продуктивності систем обертання растрових зображень з урахуванням особливостей їх представлення у обчислювальних системах, запропоновано алгоритми для реалізації операцій обертання та висунуто ідеї що до покращення якості зображень при їх виконанні.

Актуальность. Операция вращения является одной из наиболее трудоёмких операций геометрических преобразований изображений. В связи с этим, работы по повышению эффективности и производительности систем геометрических преобразований в компьютерной графике являются крайне актуальными не смотря на высокие темпы роста производительности вычислительных средств.

Для унификации выполняемых преобразований используется аппарат матричных операций в однородной системе координат, позволивший реализовать геометрические преобразования на регулярных аппаратных структурах, выполняемых в виде геометрического сопроцессора.

С целью уменьшения объемов обрабатываемых данных объекты, как правило, представляются каркасной структурой, описываемой наборами вершин элементарных объектов, которые в последующем подвергаются растеризации рисующими процессорами графической подсистемы. Такая модель может интерпретироваться так же как векторное описание объектов, для которого в 2-D системах могут использоваться средства [1-4] CORDIC семейства алгоритмов, отличительной особенностью которых является их хорошая аппаратная реализация и сравнительная простота. Последние работы, например, [6] показали, что необходимый результат при вращении точки можно получить за ограниченное и заранее определенное время работы CORDIC архитектуры – максимум за 15 тактов её работы. В работе [5] рассмотрены некоторые другие методы.

Однако даже использование таких средств является не всегда эффективным. Например, в полиграфических системах, где возникает потребность в обработке больших растровых изображений, системах

распознавания текста, системах для нанесения текстур, компьютерных тренажерных системах, требующих за ограниченное время выполнить поворот большого массива точек.

При повороте растровых изображений кроме проблемы больших объемов вычислений существует так же проблема искажения исходного изображения, накладывающая свои ограничения на построение графических систем.

В данной работе основной задачей является оптимизация вычислений в процессе вращения растрового изображения относительно центра системы координат.

Обоснование метода. С математической точки зрения операция вращения описывается уравнением вида (1):

$$[x' \quad y'] = [x \quad y] \times \begin{bmatrix} \cos(\varphi) & \sin(\varphi) \\ -\sin(\varphi) & \cos(\varphi) \end{bmatrix}. \qquad (1)$$

Данное соотношение определяет, что для выполнения операции поворота в плоскости необходимо выполнить 4 операции умножения и 2 операции сложения с плавающей запятой. Для определения координат растровой точки необходимо еще выполнить 2 операции округления. Итого – 8 операций. По времени выполнения это будет зависеть от архитектуры вычислителя.

Особенностью представления растровых изображений является то, что они могут интерпретироваться как наборы горизонтальных или вертикальных линейных массивов пикселей. В пределах такого массива индексы соседних пикселей отличаются на единицу. В такой интерпретации поворот всего изображения может интерпретироваться как совокупность поворотов линейных объектов, представляющих скан-строки или скан-столбцы.

После преобразования – поворота линейного объекта, геометрия объекта не изменяется, он останется линейным и для его развертки могут быть применены обычные методы, например, широко известный алгоритм Брезэнхэма. Для исходной идентификации линейных объектов, параллельных одной из осей, достаточно одной координаты (в плоскости). Однако, с практической точки зрения, предлагается использовать для этой цели точку P (рис.1) пересечения линейного объекта и координатной оси. Тогда, данная точка может выступать в качестве центра относительной локальной системы координат объекта для его развертки, а координаты любой другой точки A объекта могут быть определены как соответствующее значение дискретного смещения относительно точки P, что

описывается для горизонтальных скан-строк соотношением вида (2):

$$[x \quad y] = [0 \quad y] + [x \quad 0]. \qquad (2)$$

Вектор $[0 \quad y]$ определяет положение точки P, идентифицирующей соответствующую скан-строку, а $[x \quad 0]$ - вектор смещения точки A в пределах данной скан-строки.

В такой интерпретации, операция вращения может интерпретироваться как движение идентифицирующей линию точки по окружности и связанное с этим движением изменение угла наклона самой линии.

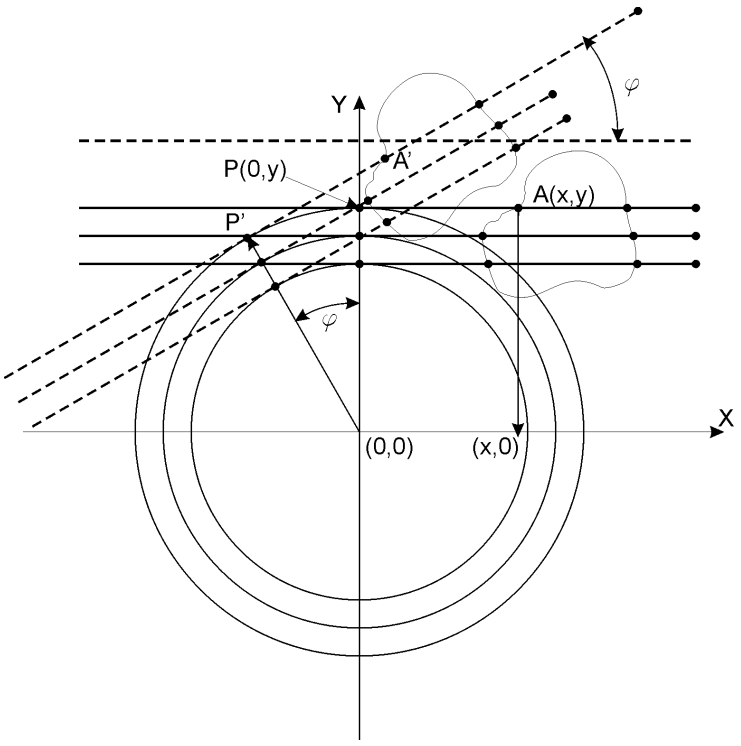


Рис. 1 - Операция вращения скан-строки.

После преобразования, точка P переходит в точку P', координаты которой, в соответствии с уравнением (1), будут определяться как:

$$[x' \quad y'] = [0 \quad y] \times \begin{bmatrix} \cos(\varphi) & \sin(\varphi) \\ -\sin(\varphi) & \cos(\varphi) \end{bmatrix} = y \cdot [-\sin(\varphi) \quad \cos(\varphi)]. \qquad (3)$$

Координаты произвольной точки A на прямой после преобразования и перехода в точку A', в соответствии с выражениями

(1) и (2), могут быть определены как:

$$([0 \quad y]+[x \quad 0])\times\begin{bmatrix}\cos(\varphi) & \sin(\varphi)\\ -\sin(\varphi) & \cos(\varphi)\end{bmatrix}=y\cdot[-\sin(\varphi) \quad \cos(\varphi)]+x\cdot[\cos(\varphi) \quad \sin(\varphi)]. \quad (4)$$

То есть, координаты точки P’ и соответствующее смещение для точки A с учетом операции поворота. Учитывая, что для растровых систем расстояние между точками изменяется дискретно на «1», получим, что на i+1 шаге развертки координаты точки Ai+1 находятся внутри скан строки (локальная линейная система координат) как

$$x_{A_{i+1}}=x_{A_i}+1. \quad (5)$$

В соответствии с выражением (4), подставив в качестве вектора смещения $\lfloor x_{A_{i+1}} \quad 0\rfloor$ значение из выражения (5) и выполнив тривиальные преобразования, получим:

$$\lfloor x_{A_{i+1}} \quad y_{A_{i+1}}\rfloor=\lfloor x_{A_i} \quad y_{A_i}\rfloor+1\cdot[\cos(\varphi) \quad \sin(\varphi)]. \quad (6)$$

Для начального цикла итерации $\lfloor x_{A0} \quad y_{A0}\rfloor=\lfloor x_{P'} \quad y_{P'}\rfloor$. Таким образом, получена итерационная формула для определения координат очередной точки растрового изображения. Для нахождения значений тригонометрических функций могут быть применены методы, предложенные в рассмотренной литературе или другие.

Результаты. Предварительный анализ приведенных соотношений показывает, что для определения координат точки P, идентифицирующей положение скан-строки необходимо выполнить (только для операции поворота) 2 операции умножения с плавающей запятой и 2 операции округления. Для определения координаты произвольной точки A’ необходимо выполнить 2 операции сложения с плавающей запятой и 2 операции округления. То есть, предложенный итерационный метод имеет значительную вычислительную эффективность в сравнении с традиционными методами выполнения поворота растровых изображений.

К недостаткам данного метода относится искажение изображения, характеризуемое смещением точек растра относительно их исходного положения при округлении значений координат и накапливаемое в процессе выполнения ряда последовательных вращений.

Для улучшения качества выводимого изображения и

уменьшения некоторых геометрических искажений можно использовать аппроксимирующие значения яркости точек, определяемые с учетом разницы линейного шага в ортогонально расположенном к координатной оси линейном объекте и объекте, расположенном под непрямым углом. Такая аппроксимация может выполняться, в частности, и с учетом корреляции к соседним линейным объектам.

Литература.

1. J. E. Volder, “The CORDIC trigonometric computing technique,” IRE Trans. Electron. Comput., vol. EC-8, no. 3, pp.330-334, Sep. 1959.
2. J. S. Walther, “A unified algorithm for elementary functions,” in Proc. Joint Spring Compute. Conf., vol.38, Jul. 1971, pp.379-385.
3. K. Maharatna and S. Banerjee, “CORDIC based array architecture for affine transformation of images,” in Proc. Int. Conf. Communications, Computers and Devices, vol. II, Kharagpur, India, Dec. 2000, pp.645-648.
4. C. Wu, A. Wu “Modified vector rotational CORDIC (MVR-CORDIC) algorithm and architecture”, IEEE Trans. Cyrcuits Syst. II, Exp. Briefs, Vol.48, no. 6, pp. 548-561, Jun. 2001
5. G. Hecstra, E. Deprettere, “Fast rotations: Low-cost arithmetic methods for orthonormal rotation,” in Proc. 13th IEEE Symposium of Comp. Arithmetic, 1997, pp.116-125.
6. K. Maharatna, S. Banerjee, E. Grass, M. Krstic, A. Troya, “Modified virtually scaling-free adaptive CORDIC rotator algorithm and arhitecture,”IEEE Trans. Circutes Syst. Video Technol., vol.15, no. 11, pp.1463-1474, Nov. 2005.

Получено 01.06.07

УДК 681.511.2:517.442

АППРОКСИМАЦИЯ ОБРАТНОГО ПРЕОБРАЗОВАНИЯ ЛАПЛАСА АНАЛИТИЧЕСКИМИ ВЫРАЖЕНИЯМИ В СЛУЧАЕ СЛОЖНЫХ ИЗОБРАЖЕНИЙ

О.А. Погорелов
Восточноукраинский национальный университет им. В. Даля

В і обґрунтовується метод приблизного зворотнього перетворення складних зображень за Лапласом, які не мають аналітичних функцій-оригіналів. Аналітична апроксимація оригіналу залежить від змінної, яка визначається аналітично (для вузького діапазону апроксимації), або чисельно (для більш широкого діапазону).

Математические модели линейных систем чаще всего записываются в виде передаточных функций, представляющих собой изображения по Лапласу импульсных весовых функций. Переход от изображения к оригиналу является символьным преобразованием, оригинал и изображение должны быть аналитическими функциями. Однако существуют сложные изображения, оригиналы которых не могут быть представлены аналитически. В этом случае приходится использовать аппроксимацию обратного преобразования Лапласа аналитической функцией, при этом, удовлетворительную точность можно обеспечить лишь в ограниченном диапазоне изменений аргумента [1]. Для некоторых приложений этого вполне достаточно.

Рассмотрим следующую задачу. Необходимо получить обратное преобразование Лапласа функции $F(p,x)$, где $p=\sigma+i\omega$ – комплексный параметр преобразования Лапласа, действительная часть которого равна σ, а мнимая: ω (i – мнимая единица); второй аргумент x означает либо: $x=\mu\cdot p$, либо $x=\dfrac{\mu}{p}$, где μ – малый параметр (вещественное положительное число, меньшее единицы). Если $\mu=0$, то обратным преобразованием функции $F(p,0)$, по условию, является аналитическая функция $f_0(t)$ действительного переменного t. Если же $\mu\neq 0$, то оригинал $f(t)$ изображения $F(p,x)$ не может быть представлен в аналитическом виде. Данная статья посвящена изложению и обоснованию метода аналитической аппроксимации обратного преобразования в этом случае.

Рассмотрим изображение $\Phi(p,y)$, которое отличается от изображения $F(p,x)$ только вторым аргументом, который равен либо: $y=\mu\cdot\sigma$, либо $y=\dfrac{\mu}{\sigma}$. Поскольку функция $f_0(t)$ – аналитическая, то оригинал $\varphi(t,\sigma)$ изображения $\Phi(p,y)$ также является аналитической функцией (y не зависит от переменной ω).

Теорема 1. Если $f(t)$ и $\varphi(t,\sigma)$ – обратные преобразования изображений $F(p,x)$ и $\Phi(p,y)$ соответственно, то справедливо равенство:

$$\int_0^\infty e^{-\sigma t} f(t)dt = \int_0^\infty e^{-\sigma t}\varphi(t,\sigma)dt\,. \qquad (1)$$

Доказательство. Запишем преобразования Лапласа в следующем виде

$$F(p,x)=\int_0^\infty e^{-\sigma t}e^{-i\omega t}f(t)dt\,; \qquad \Phi(p,y)=\int_0^\infty e^{-\sigma t}e^{-i\omega t}\varphi(t,\sigma)dt\,.$$

В соответствии с определением функций $F(p,x)$ и $\Phi(p,y)$ для $\omega=0$ имеем $F(\sigma+i\cdot 0)=\Phi(\sigma+i\cdot 0)$, откуда следует равенство (1).

Следствие. Из-за того, что подынтегральные функции $e^{-\sigma t}f(t)$ и $e^{-\sigma t}\varphi(t,\sigma)$ имеют общий множитель $e^{-\sigma t}$, они практически не равны нулю лишь на ограниченном интервале: $0\le t\le T$.

$$T=T(\sigma)=\frac{\Lambda}{\sigma-\sigma_0}, \qquad (2)$$

где Λ – достаточно большое вещественное положительное число, σ_0 – вещественное положительное число, задающее область определения функции $F(p,x)$ (в полуплоскости $\operatorname{Re} p>\sigma_0$). В силу равенства (1) непрерывные вещественные функции $f(t)$ и $\varphi(t,\sigma)$ должны иметь на интервале $0\le t\le T$ хотя бы одну общую точку t_c (точку пересечения). Можем записать

$$f(t_c)=\varphi(t_c,\sigma). \qquad (3)$$

Из уравнения (3) следует: $\sigma=\sigma(t_c)$. Свойства функции $\sigma(t_c)$ можно установить на основании следующих соображений. Увеличивая переменную σ, мы сужаем интервал $0\le t\le\dfrac{\Lambda}{\sigma-\sigma_0}$ и соответственно уменьшаем значение t_c. Если $\sigma=\infty$, то $t_c=0$ (величина интервала T при этом также равна нулю). Таким образом,

зависимость между переменными σ и t_c близка к обратно-пропорциональной. Перепишем уравнение (1) в следующей форме:

$$\int_0^{tc} e^{-\sigma\tau}\left[f(\tau)-\varphi(\tau,\sigma)\right]d\tau = -\int_{tc}^{T} e^{-\sigma t}\left[f(\tau)-\varphi(\tau,\sigma)\right]d\tau . \qquad (4)$$

Для того чтобы равенство (4) выполнялось, произведение $\sigma \cdot t_c$ должно быть порядка единицы. Точнее: $\sigma(t_c)\cdot t_c = \lambda(t_c)$, где $\lambda(t_c)$ – медленно меняющаяся функция t_c, конкретный вид которой зависит от вида изображения $F(p,x)$. Если $t_c = 0$, то и $T = 0$. Когда $t_c > 0$, $T > t_c$. Значит, в области малых значений переменная t_c растет медленнее, чем интервал T. Оригиналы рациональных изображений $\Phi(p,y)$ имеют следующий вид

$$\varphi(t,\sigma) = \sum_{k=1}^{N} A_k e^{-\alpha_k t} , \qquad (5)$$

где A_k – коэффициенты, $-\alpha_k$ – полюсы функции $\Phi(p,y)$ (некоторые из коэффициентов и полюсов зависят от σ). Функция $f(t)$ на определенном отрезке изменений аргумента также может быть аппроксимирована выражением вида (5). Поэтому при больших значениях t_c уравнение (3) приводит к асимптотическим зависимостям вида: $\sigma = \sigma_0 + A \cdot e^{-\alpha t}$, где $-\alpha$ – наименьший по модулю полюс. Отсюда следует: $t_c = \frac{1}{\alpha}\ln\left(\frac{A}{\sigma-\sigma_0}\right)$. Сравнивая эту формулу с уравнением (2), видим, что и в области больших значений t_c эта переменная также «отстает» в росте от T при изменении σ. Поэтому можем записать: $T(t_c) = \sum_{k=1}^{\infty} c_k t_c^k$, где $c_k \geq 0$ – положительные коэффициенты. Тогда из (2) следует

$$\lambda(t_c) = \sigma_0 t_c + \frac{\Lambda}{c_1 + \sum_{k=2}^{\infty} c_k t_c^{k-1}} .$$

Следовательно, если $\sigma_0 > 0$, то функция $\lambda(t_c)$ неограниченно увеличивается. Если же $\sigma_0 \leq 0$, то функция $\lambda(t_c)$ монотонно уменьшается. Введем функцию комплексного переменного

$$H(p,\sigma) = \frac{p}{p-\sigma}\left[F(p,x) - \Phi(p,y)\right]. \qquad (6)$$

Поскольку функции $F(p,x)$ и $\Phi(p,y)$ – изображения по Лапласу, то изображением по Лапласу будет и функция $H(p,\sigma)$. Это

означает, что существует оригинал: $h(t,\sigma) \div H(p,\sigma)$. Из уравнения (6) следует: $f(t) - \varphi(t,\sigma) = h(t,\sigma) - \sigma\int_0^t h(\tau,\sigma)d\tau$. В точке пересечения

$$\lambda(t_c) = \sigma \cdot t_c = \frac{t_c \cdot h(t_c,\sigma)}{\int_0^{tc} h(\tau,\sigma)d\tau}. \qquad (7)$$

Теорема 2. $\lambda(0)$ – натуральное число.

Доказательство. Разложим оригинал $h(t,\sigma)$ в степенной ряд

$$h(t,\sigma) = a_0(\sigma) + a_1(\sigma)t + a_2(\sigma)t^2 + a_3(\sigma)t^3 + \ldots$$

Тогда из (7) следует

$$\lambda(t) = \frac{a_0 t + a_1 t^2 + a_2 t^3 + a_3 t^4 + \ldots}{a_0 t + a_1 \frac{t^2}{2} + a_2 \frac{t^3}{3} + a_3 \frac{t^4}{4} + \ldots}.$$

Если $a_0 \neq 0$, то $\lambda(0) = 1$. Если $a_0 = 0$ и $a_1 \neq 0$, то $\lambda(0) = 2$. Если $a_0 = 0$ и $a_1 = 0$, а $a_2 \neq 0$, то $\lambda(0) = 3$ и т.д..

Разложим функции $F(p,x)$ и $\Phi(p,y)$ в ряд Маклорена по «неудобным» параметрам x и y соответственно

$$F(p,x) = F(p,0) + \frac{\partial F(p,0)}{1!\partial x}x + \frac{\partial^2 F(p,0)}{2!\partial x^2}x^2 + \frac{\partial^3 F(0)}{3!\partial x^3}x^3 + \ldots$$

$$\Phi(p,y) = F(p,0) + \frac{\partial F(p,0)}{1!\partial x}y + \frac{\partial^2 F(p,0)}{2!\partial x^2}y^2 + \frac{\partial^3 F(0)}{3!\partial x^3}y^3 + \ldots$$

и вычтем одну функцию из другой

$$F(p,x) - \Phi(p,y) = \frac{\partial F(p,0)}{1!\partial x}(x-y) + \frac{\partial^2 F(p,0)}{2!\partial x^2}(x-y)(x+y) + \ldots =$$

$$= (x-y)\left[\frac{\partial F(p,0)}{1!\partial x} + \frac{\partial^2 F(p,0)}{2!\partial x^2}(x+y) + \frac{\partial^3 F(0)}{3!\partial x^3}(x^2 + xy + y^2) + \ldots\right].$$

Если $x = \frac{\mu}{p}$ и $y = \frac{\mu}{\sigma}$, то $(x-y) = \frac{\mu}{p} - \frac{\mu}{\sigma} = \mu\frac{\sigma - h}{p\sigma}$. Если $x = \mu \cdot p$ и $y = \mu \cdot \sigma$, то $(x-y) = \mu(p-\sigma)$. Для первого и второго случаев соответственно имеем

$$H(p,\sigma) = -\frac{\mu}{\sigma} \cdot \left[\frac{\partial F(p,0)}{1!\partial x} + \frac{\partial^2 F(p,0)}{2!\partial x^2}(x+y) + \frac{\partial^3 F(0)}{3!\partial x^3}(x^2 + xy + y^2) + \ldots\right].$$

$$H(p,\sigma) = \mu \cdot p \cdot \left[\frac{\partial F(p,0)}{1!\partial x} + \frac{\partial^2 F(p,0)}{2!\partial x^2}(x+y) + \frac{\partial^3 F(0)}{3!\partial x^3}(x^2 + xy + y^2) + \ldots\right].$$

Если параметр μ достаточно мал, то можно ограничиться

первым членом ряда в квадратных скобках. Это позволяет получить приближенное выражение для оригинала $h(t,\sigma)$ в аналитическом виде. Далее по формуле (7) определяем функцию $\lambda(t)$. Тогда функция $\varphi\left(t,\frac{\lambda(t)}{t}\right)$ является аналитической аппроксимацией искомой функции $f(t)$ в ограниченном диапазоне, примыкающем к началу координат. Если мы хотим увеличить диапазон аппроксимации, при определении $H(p,\sigma)$ необходимо учитывать другие члены ряда в квадратных скобках. Поскольку переменная $\lambda(t)$ фигурирует в левой и правой частях выражения (7), в этом случае для ее вычисления нужно использовать метод последовательных приближений. В качестве примера на рис. 1 показан график функции-оригинала $f(t)$ для изображения $F(p,x)=\frac{1}{(p+0.1)\left(p+\frac{0.5}{1+x}\right)}$, $x=\frac{0.1}{p}$. Рядом показаны график функции $f_0(t)$ и график аналитической аппроксимации по предложенному методу $\varphi(t,\lambda(t))$ (в первом приближении). Как можно видеть, по сравнению с «нулевым приближением» диапазон аппроксимации увеличился более чем в три раза.

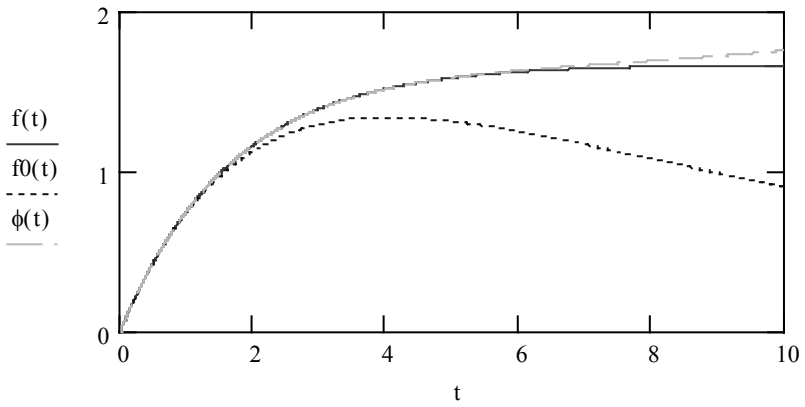


Рис. 1

Литература

1. Погорелов О.А. Приближенное обращение сложных изображений по Лапласу// Інформаційні технології та комп'ютерна інженерія, № 3, 2005. – С. 178-182.

Получено 01.06.07

УДК 681.325.5(07)

МОДЕЛЬ ЦИФРО-АНАЛОГОВОГО ПРЕОБРАЗОВАНИЯ НА ОСНОВЕ ШИМ МИКРОКОНТРОЛЛЕРОВ AVR

М.И.Ледовской
Технологический институт
Южного федерального университета в г. Таганроге

Предлагается модель цифро-аналогового преобразования на основе ШИМ микроконтроллеров AVR, которая предназначена для использования в учебном процессе в качестве электронного методического пособия. Приводятся расчетные формулы, описание лицевой панели и блок-схемы модели, созданной в системе LabVIEW 7.1 фирмы National Instruments.

Важным этапом во многих процессах управления с помощью цифровых систем является цифро-аналоговое преобразование – генерация аналогового сигнала с уровнем напряжения, соответствующим цифровому значению [1]. Примером может служить управление угловой скоростью электродвигателя постоянного тока с помощью микроконтроллера. В данном случае цифровое значение управляющего сигнала, получаемое в микроконтроллере, необходимо преобразовать в аналоговое напряжение для подачи на якорь электродвигателя [2].

Традиционным способом решения данной задачи является использование цифро-аналоговых преобразователей (ЦАП). Однако в случае использования микроконтроллеров AVR более простой альтернативой является выполнение цифро-аналогового преобразования с помощью ШИМ-сигнала. Большинство AVR-контроллеров оснащено таймерами/счетчиками, которые могут работать в режиме широтно-импульсного модулятора [3].

Задача цифро-аналогового преобразования рассматривается в ряде дисциплин, посвященных архитектуре и применению микроконтроллеров. Для успешного усвоения данной темы необходимо использовать в учебном процессе электронные методические пособия, где обеспечивается наглядная визуализация принципа цифро-аналогового преобразования. Современные информационные технологии от фирмы National Instruments позволяют создавать методические пособия, отвечающие указанному требованию. Применение таких пособий в

учебном процессе отвечает задачам инновационного подхода в образовании.

Настоящая работа иллюстрирует пример электронного методического пособия в виде модели цифро-аналогового преобразования на основе ШИМ микроконтроллеров AVR. Данная модель разработана автором в системе LabVIEW 7.1 фирмы National Instruments.

Как известно, способ преобразования цифровых значений в аналоговые напряжения с помощью ШИМ-сигнала заключается в изменении коэффициента заполнения импульсов прямоугольного напряжения пропорционально исходному цифровому значению. Если T – период ШИМ-сигнала, а t_H – длительность импульса, то коэффициент заполнения является отношением $g = t_H / T$ (рис.1) [3].

Предположим, что ШИМ-сигнал образуется с помощью таймера/счетчика T/C1 микроконтроллера AVR. Тогда коэффициент заполнения ШИМ-сигнала g прямо пропорционален содержимому регистра OCR1:

$$g = \frac{Z}{TOP}, \tag{1}$$

где $TOP=2^N-1$ – максимальное значение счетчика; N – разрешение широтно-импульсного модулятора, которое задается программно в виде 8, 9 или 10 двоичных разрядов; Z – содержимое регистра OCR1, причем $0 \le Z \le TOP$.

Частота ШИМ-сигнала, формируемого на выводе OC1 микроконтроллера, определяется по формуле

$$f_{ШИМ} = \frac{f_{T/C1}}{2^{N+1} - 2}, \tag{2}$$

где $f_{T/C1}$ – частота T/C1. Данная частота зависит от программно задаваемого коэффициента предварительного деления $K_{T/C1}$, который принимает значения 1, 1/8, 1/64, 1/256 или 1/1024:

$$f_{T/C1} = \frac{f_{CK}}{K_{T/C1}}, \tag{3}$$

где f_{CK} – тактовая частота микроконтроллера.

Если полученный ШИМ-сигнал пропустить через ФНЧ, то в идеальном случае образуется среднее арифметическое значение постоянного напряжения U_M:

$$U_M = (U_H - U_L) \cdot g + U_L = (U_H - U_L) \cdot \frac{Z}{2^N - 1} + U_L, \tag{4}$$

где U_L и U_H – уровни логического “0” и логической “1” соответственно (рис.1). Таким образом, получаемый аналоговый сигнал U_M

прямо пропорционален исходному цифровому значению Z, находящемуся в регистре OCR1.

При использовании ФНЧ с достаточно высокой крутизной АЧХ и/или достаточно низкой частотой среза можно добиться такого подавления гармоник, что точность сигнала постоянного напряжения будет также достаточно высокой. Однако снижение частоты среза увеличивает время установления выходного сигнала фильтра. Поэтому на практике находят компромисс между желаемой точностью выходного сигнала и временем его установления. Зачастую частота среза ФНЧ выбирается из соотношения $f_g=0{,}25\cdot f_{ШИМ}$, где $f_{ШИМ}=1/T$ – частота ШИМ-сигнала (частота первой гармоники).

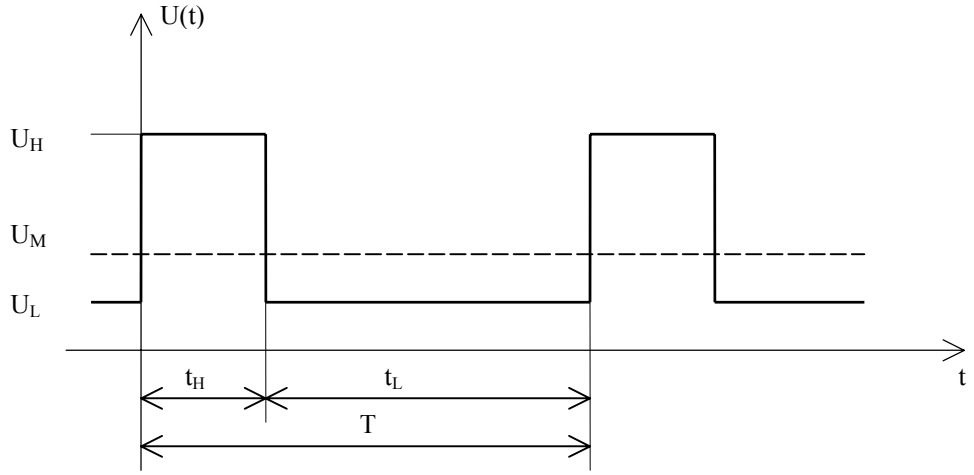


Рис.1. ШИМ-сигнал

Ниже приводится описание модели цифро-аналогового преобразования на основе ШИМ микроконтроллеров AVR, которая разработана в системе LabVIEW 7.1. Лицевая панель модели содержит следующие блоки (рис.2):

- Блок **AVR контроллер** содержит регуляторы **Регистр OCR1 T/C1, Тактовая частота AVR, Делитель частоты T/C1, Уровни лог.0 и лог.1,** а также кнопку **STOP.**
- Блок **Измеритель ШИМ-сигнала** содержит регулятор **Периоды**, а также индикаторы **Частота**, **Заполнение**, **Импульс** и индикатор-осциллограф ШИМ-сигнала.
- Блок **ФНЧ Баттерворта** содержит регуляторы **Частота среза, Порядок** и **Маска погрешности**.
- Блок **Аналоговый сигнал** содержит индикатор-осциллограф аналогового сигнала и индикатор-осциллограф погрешности цифро-аналогового преобразования.

Блок-схема модели обеспечивает выполнение следующих функций (рис.3):

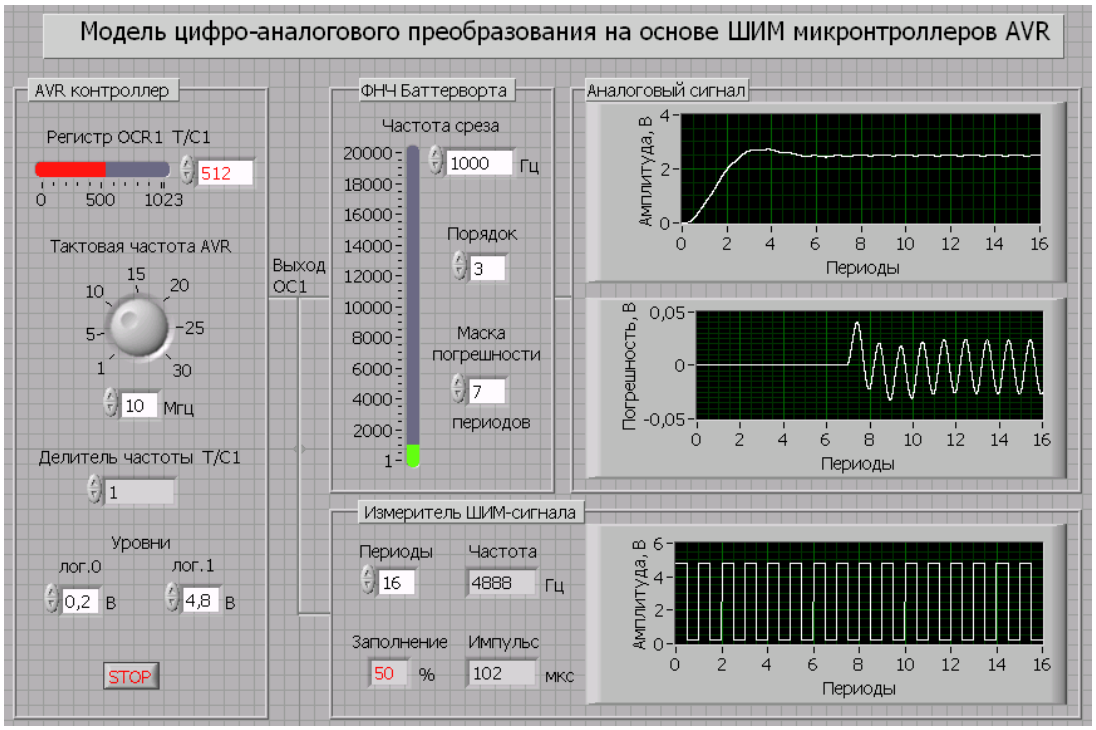


Рис.2. Лицевая панель модели

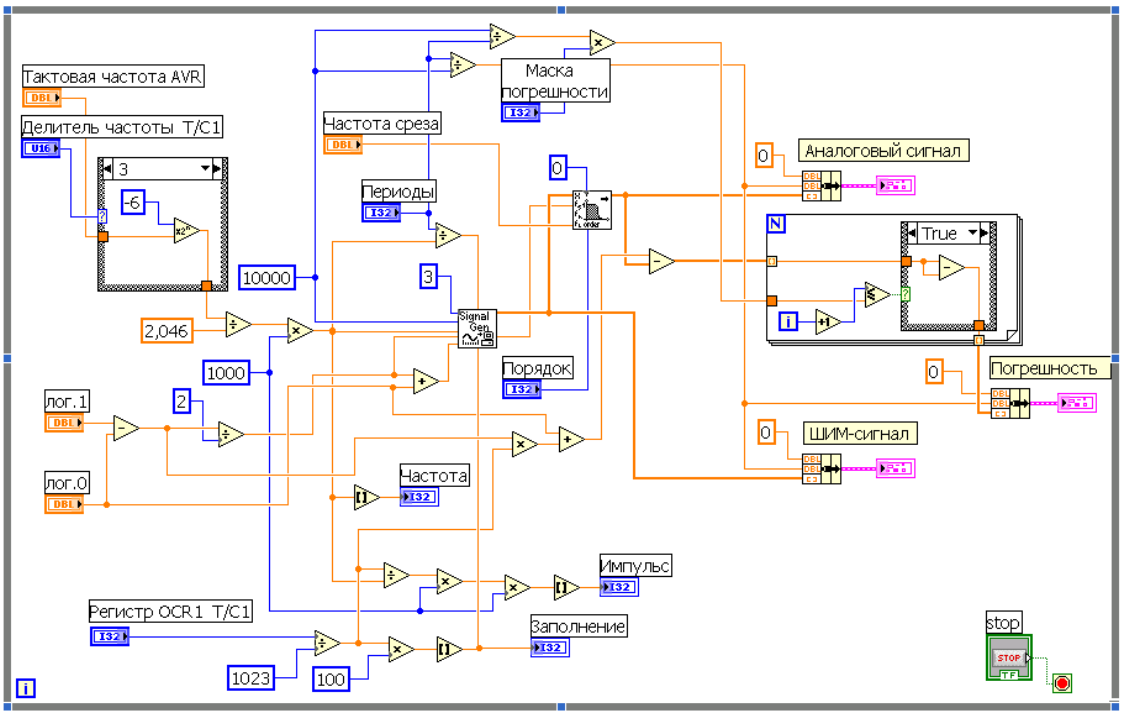


Рис.2. Блок-схема модели

1. Формирование ШИМ-сигнала, исходя из установленных значений регуляторов **Регистр OCR1 T/C1, Тактовая частота AVR, Делитель частоты T/C1, Уровни лог.0 и лог.1**. При этом моделируется режим неинвертирующего широтно-импульсного модулятора T/C1 микроконтроллеров AVR с разрешением N=10 и TOP=1023. В качестве исходных соотношений используются формулы (1), (2), (3).
2. Вычисление параметров ШИМ-сигнала **Частота**, **Заполнение**, **Импульс.** Длительность ШИМ-сигнала определяется, исходя из состояния регулятора **Периоды.**
3. Фильтрацию ШИМ-сигнала, исходя из установленных значений регуляторов **Частота среза, Порядок.** Данная задача решается путем моделирования ФНЧ Баттерворта.
4. Определение погрешности цифро-аналогового преобразования путем сравнения выходного сигнала ФНЧ с эталонным значением аналогового сигнала, вычисляемым по формуле (4). При этом с помощью регулятора **Маска погрешности** обеспечивается маскирование переходного процесса для наблюдения погрешности установившегося значения ФНЧ.

Предлагаемая модель предназначена для следующих учебных целей:

1. Наглядная иллюстрация принципа цифро-аналогового преобразования на основе ШИМ микроконтроллеров AVR.
2. Исследование параметров ШИМ-сигнала с помощью регуляторов **Регистр OCR1 T/C1, Тактовая частота AVR, Делитель частоты T/C1, Уровни лог.0 и лог.1**.
3. Исследование погрешности цифро-аналогового преобразования с помощью регуляторов **Частота среза** и **Порядок**.

Библиографический список

1. Олссон Г., Пиани Д. Цифровые системы автоматизации и управления. СПб.: Невский диалект, 2001. – 557с.
2. Ледовской М.И. Целочисленный алгоритм ПИД-регулятора для микроконтроллеров. Материалы международной научной конференции «Проблемы развития естественных, технических и социальных систем»- часть 4 – Таганрог: Изд. «Антон», ТТИ ЮФУ, 2007, с.42-47.
3. Трамперт В. Измерение, управление и регулирование с помощью AVR–микроконтроллеров. – К.: «МК-Пресс», 2006. – 208с.

Получено 01.06.07

УДК 612.172.4 (07)

ИСПОЛЬЗОВАНИЕ СТРУКТУРНО-ЛИНГВИСТИЧЕСКОГО МЕТОДА АНАЛИЗА ДЛЯ РАСПОЗНАВАНИЯ СТРУКТУРЫ ЭКГ-СИГНАЛОВ

А.В.Максимов
Технологический институт
Южного федерального университета в г. Таганроге

Рассмотрен комбинированный метод анализа ЭКГ-сигналов с использованием сплайн-аппроксимации.

Для распознавания структуры ЭКГ-сигнала (ЭКС) активно используют структурно-лингвистические методы, когда дискретизированный ЭКС заменяется символьной строкой, составленной из некоторого представительного алфавита. Использование таких методов позволяет значительно сократить объем, который необходим для хранения ЭКС и, таким образом увеличить скорость анализа сигнала.

Исходя из проведенного анализа, известных алгоритмов аппроксимации и свойств ЭКС, предлагается использовать комбинированный алгоритм аппроксимации ЭКС. Так, как ЭКС предварительно сегментирован, то мы заранее знаем в каких зонах следует ожидать успешного применения сплайн аппроксимации или кусочно-линейной интерполяции 1-го порядка (ИПП). Для сплайнов такими зонами являются зоны ЭКС: P-комплекса, ST-T-комплекс и диастолический интервал, т. е. интервал T-P. Для QRS-комнлекса используем ИПП аппроксимацию и для конечных элементов QRS, в случае большой ошибки ИПП (эту ситуацию легко заметить по возрастанию интервалов аппроксимации свыше 10), применяем аппроксимацию квадратичными параболами. Аппарат сплайнов здесь плохо пригоден ввиду небольших участков QRS, имеющих сглаженную форму.

В результате аппроксимации получаем представление ЭКС в виде набора участков, каждый из которых характеризуется типом аппроксимирующей функции: линейная, квадратичная, кубическая, и соответственно набором коэффициентов: два коэффициента в случае линейной функции, три— квадратичной, четыре — кубической функции. В дальнейшем удобно рассматривать все функции как кубические с нулевыми значениями первых одного или двух коэффициентов. Для формирования алфавита необходимо построить четырехмерное пространство, где осями являются четыре коэффициента аппроксимирующей функции, тогда каждый участок ЭКС даст точку в этом про-

странстве. Близкие по виду участки группируются в кластеры, которые можно обозначить символами некоторого алфавита. Для разделения кластеров удобно применять метод разделения по Евклидову расстоянию Dm_k точек в четырехмерном пространстве. Начальную группировку кластеров желательно сформировать при формировании обучающей выборки ЭКС. Для этого алгоритму анализа необходимо предоставить определенный набор ЭКС и графически воспроизвести представление того же ЭКС с использованием наиболее близких кластеров. Полученное представление сравнивается с исходным ЭКС. Если интегральная погрешность представления из кластеров превышает заданный порог погрешности, то делается попытка улучшить его, определив, какой кластер вносит наибольшую погрешность. Для этого поочередно каждый кластер заменяется аппроксимирующей функцией соответствующего участка и вычисляется погрешность. Эта операция повторяется для всех участков. Максимум погрешности указывает на участок, который должен образовать новый кластер. Процесс создания нового кластера должен проводиться осторожно, поскольку велика вероятность создания «артефактных» кластеров. Реальный кластер, соответствующий некоторому новому варианту сигнала обычно занимает некоторое промежуточное или несильно удаленное мести среди ему подобных. Меру этого подобия (расстояние Dm_k) следует рассчитывать отдельно для линейных, квадратичных и кубических кластеров. В результате данной процедуры одномерный ЭКС заменяется последовательностью символов, каждый из которых представляет собой кластер в 4-х мерном пространстве и имеет некоторые характеристики (атрибуты): 1—длительность, 2—тип, 3—массив коэффициентов. Данная информация может быть использована при проведении синтаксического разбора сигнала. К недостаткам этого метода следует отнести оторванность такого разбора от векторной природы ЭКС. При этом длины цепочек в соседних отведениях получаются разными, т. к. разная сложность сигналов в этих отведениях. Символы, при независимом анализе ЭКС несинхронны, что не позволяет при синтаксическом анализе учесть одновременно происхождение процесса в разных отведениях. Данный недостаток приводит к существенным диагностическим ошибкам и является одной из тех причин, в силу которых структурно-лингвистический метод находит ограниченное применение в системах автоматического анализа ЭКС, поэтому предлагается модифицированный метод, свободный от указанных недостатков.

Основой для согласованного символьного представления является понятие мультисимвола, т. е. символа, представляющего элемент

N-сигналов, синхронных по времени и имеющих идентичную длительность. Можно предложить два варианта формирования такого символа. Первый вариант заключается в прореживании символьных цепочек, предварительно обработанных вышеописанным алгоритмом. Рассмотрим подробнее алгоритм прореживания. Для данного алгоритма входом является N-символьных цепочек с атрибутами Sm, m=l…N. Каждая цепочка имеет длину Lm-символов. Введем массив Cm, где каждый элемент массива отслеживает текущую позицию разбора в строке Cm. В результате алгоритма, формируется массив из N-строк, одинаковой длины Pm.

Полученный массив строк Pm образован из символов, синхронных во времени, т.е. обладающих сходным первым атрибутом (длительностью). Остальные атрибуты могут быть, и обычно бывают, различными. Дальнейший шаг по формированию единой цепочки — объединение групп символов, синхронных во времени, в единый символ из нового алфавита. Комбинаторный метод здесь не подходит, т. к. обычно число символов в исходном алфавите порядка 200, и число сочетаний из 200 по N получается достаточно большим.

Поэтому формирование мультисимвола из N-символов одномерного алфавита будем производить с помощью специальной грамматики, учитывающей амплитудные и скоростные характеристики входящих символов. Поскольку длина строки фиксирована и физически понятны особенности разделяемых классов (изоэлектрическая линия, плавный склон, закругленная вершина, крутой склон, острая вершина), можно построить достаточно простую грамматику для получения набора из не более 1000-1500 символов. Если при построении такой грамматики воспользоваться атрибутом <тип сегмента> из алгоритма сегментации, процесс формирования алфавита еще более упростится. Например, в сегменте <P-комплекс> априорно не может быть никаких острых вершин и острых склонов, и если таковые встретились, то они являются следствием помех или артефактов, и соответствующее правило должно сводить их к существующим (разрешенным) вариантам P-комплекса. Метод быстро работает, в совокупности с независимым представлением достаточно компактен, но обладает одним недостатком — дает длинные символьные строки, часто состоящие из мало различающихся (по смыслу) мультисимволов. Поэтому, для устранения этих недостатков, при наличии достаточных вычислительных ресурсов, можно воспользоваться вторым способом построения мультисимвольной строки. Для этого алгоритм аппроксимации, например, первого порядка (или сплайн), работает синхронно во всех N-отведениях. Получающиеся коэффициенты образуют 4-N-

мерное пространство, в котором и происходит процесс разделения кластеров. Основная проблема здесь — выбор критериев разделения, которую можно решить путем поиска: сначала отведение, дающее максимальный вклад и погрешность, а затем кластера, который следует разбить на два подкластера (т. е. образовать новый). Поскольку процесс аппроксимации происходит синхронно, результирующая строка во всех отведениях получается одной длины, а мультисимвол имеет следующие атрибуты: длительность, массив из N-типов аппроксимирующих функций, массив из 4N-коэффициентов. Дальнейший ход синтаксического разбора будем строить исходя из того, что мультисимвольная цепочка тем или иным методом уже создана.

Дальнейшее распознавание ЭКС, представленных в виде мультисимвольной строки, производится методом грамматического разбора. Грамматический разбор в общем случае позволяет определить: принадлежит ли данная цепочка символов множеству всех цепочек, порождаемых грамматикой языка, или не принадлежит, т. е. с точки зрения теории формальных грамматик и языков, результатом синтаксического разбора является величина логического типа — «Да/Нет». Для нас более важен побочный продукт разбора, а именно, какие конкретно нетерминалы и в каком порядке встретились при разборе цепочки. Собственно последовательность нетерминалов и представляет собой результат разбора, т. е. структуру сигнала, описанного понятиями более высокого порядка, чем мультисимвол. Факт же принадлежности цепочки к языку, означает либо наличие артефактов цепочке, либо необходимость расширения множества правил в грамматике. Для отлаженной системы такие случаи должны быть исключены. Это означает, что при неуспешном разборе цепочки должен производиться анализ того нетерминала, для которого оказались неуспешными попытки применения всех его образующих грамматических правил. После нахождения такого нетерминала необходимо попытаться расширить грамматику путем создания нового правила. Данный процесс желательно выполнять в интерактивном режиме, иначе возможно неконтролируемое возрастание грамматики, что приведет к росту времени анализа и большому количеству ошибочных заключений. Данную процедуру далее будем называть — нейтрализацией ошибки разбора.

В теории грамматического разбора известно два основных метода разбора — сверху вниз и снизу вверх. Алгоритм снизу вверх предопределяет движение от строки терминалов к основному символу грамматики. Обычно подобный разбор осуществляется с помощью матриц предшествования. Для случая анализа сигналов, когда число

нетерминалов порядка 200—300, размер матриц вырастает до 40—90 тыс. элементов, что является неприемлемым. Поэтому, значительно удобнее использовать алгоритм сверху вниз, когда используется рекурсивная процедура для вывода строки терминалов из основного символа грамматики. При этом чрезвычайно важно, чтобы наиболее вероятные правила стояли первыми, а редко используемые альтернативы правил будут задействованы реже. При несоблюдении этого условия начинает расти количество возвратов с признаком неуспешного анализа.

Данный алгоритм позволяет решить формальную часть синтаксического разбора, т. е. ответить на вопрос: принадлежит ли входная строка множеству строк данного языка. Однако для практических целей важен ответ на вопрос — из каких крупных элементов состоит фрагмент анализируемого сигнала, т. е. какова его морфология. Для решения этой задачи используем понятие атрибутов.

Атрибутом называется некоторое поле данных, связанное с терминалом или нетерминалом. Атрибутов может быть несколько, и их тип может различный. Кроме атрибутов в атрибутной грамматике вводятся понятия атрибутных правил.

Синтезируемый атрибут получает свое значение исходя из значений атрибутов терминалов и нетерминалов, находящихся в правой части правила. Он является атрибутом нетерминала.

Наследуемый атрибут — это атрибут нетерминала, находящегося в правой части правила и получающий свое значение на основании атрибутов нетерминалов, находящихся в левой части правила.

Таким образом, синтезируемые атрибуты позволяют организовать движение вверх по дереву разбора, а наследуемые— вниз по дереву разбора. Атрибутные правила могут иметь достаточно сложный вид и структуру, в частности в качестве результатов работы атрибутных правил могут быть словесные заключения о характера ЭКГ. В принципе, подход с использованием атрибутных грамматик позволяет полностью формализовать обработку сигнала, от распознавания его структуры до формирования заключения.

Библиографический список

1. Браверман Э.М., Мучник И.Б. Структурные методы обработки эмпирических данных. М.: Наука, 1983. 463с.
2. Кнут Д.Э. Искусство программирования для ЭВМ. Получисленные алгоритмы, т2, М.: Мир, 1977. 723с.

Получено 01.06.07

УДК 621.317.765

НЕОБХОДИМОСТЬ В РАЗРАБОТКЕ МЕТОДИК ПРОВЕДЕНИЯ ИССЛЕДОВАНИЙ И ИСПЫТАНИЙ ДЛЯ ИНТЕЛЛЕКТУАЛЬНОГО ДАТЧИКА ИЗМЕРЕНИЯ АБСОЛЮТНОГО ДАВЛЕНИЯ

А.В. Пирский

Таганрогский Технологический Институт Южного Федерального Университета

Показана проблема при проведении испытании интеллектуальных датчиков абсолютного давления. Рассмотрены существующие методики проведения испытаний и обоснована необходимость в разработке новой методики для интеллектуального датчика абсолютного давления.

Сегодня с развитием микроэлектроники и микрокомпьютерной техники сформировалось понятие – интеллектуальный датчик (ИД). Под ИД понимается то, что в одном корпусе совмещены аналоговый датчик (Чувствительный элемент) и миниатюрный микропроцессорный модуль, выполняющий оцифровку, фильтрацию величины давления и осуществляющий информационный обмен с ведущей системой. Применение цифрового датчика давления позволяет отказаться от использования длинных аналоговых линий от датчика до ведущей системы, которые чувствительны к электромагнитным шумам, приводящим к потере точности измерения за счет наводок. Использование ИД обеспечивает:

– возможность двустороннего обмена цифровой информацией при настройке и эксплуатации датчика;
– диагностику датчика и электроники;
– изменение диапазона измерений в широких пределах;
– стабильный результат измерений при изменяющихся режимах работы.

Превосходство интеллектуальных датчиков достигается в результате:

– высокой точности измерений – погрешность до 0,1%, погрешность от влияния температуры окружающей среды – до 0,09% / 10 C^0;
– снижения эксплуатационных и капитальных затрат, благодаря экономии кабельной продукции и минимизации расходов на пуско-

наладочные работы, удаленной диагностике и настройке параметров прибора;

– использования в любой системе управления, в которой применяется реализованный сетевой протокол;

– возможности унификации приборного парка, уменьшения ремонтно-эксплуатационных запасов, благодаря сокращению модельного ряда приборов и самостоятельному выбору инженерных единиц измерения.

Широкое распространение получили интеллектуальные датчики измерения абсолютного давления (ИДАД). При разработке ИДАД решается задача с выбором аппаратных, алгоритмических и программных средств. При выборе этих средств проектировщик и изготовитель датчиков проводит ряд исследований направленных на выявление слабых мест выбираемых средств. А также каждый проектировщик и изготовитель ИД стремится совершенствовать свою продукцию с целью повышения ее эксплуатационных характеристик [1]. Для этого нужны методики по проведению исследований и испытаний. Существуют традиционные и стандартные методики и программы испытаний, например, отечественная МИ1997. Вместе с тем существуют и не стандартные методики, целью которых является исследование используемых в датчиках материалов, конструкции датчика, технологии изготовления и индивидуальных особенностей датчика (например, в ИДАД применяются алгоритмы компенсации температурной ошибки ЧЭ). Каждая фирма-производитель постоянно совершенствует конструкцию выпускаемых изделий, технологию производства и применяемые материалы. Главной целью таких модернизаций является повышение основных метрологических характеристик при одновременном улучшении показателей надежности. Данная цель не может быть достигнута без использования собственных программ и методик испытаний. Только правильно разработанные и правильно проведенные испытания позволяют техническому специалисту оценить потенциальные возможности устройства и принять необходимые инженерные решения.

Целью работы является разработка специализированной методики проведения исследования и испытания (МПИ) для ИДАД. Таким образом, для разработки ИДАД ставится задача в формировании специализированной методики проведения исследования и испытания (МПИ). Для решения поставленной задачи было решено проводить построение МПИ ИДАД на основе уже существующей МИ1997, как общепринятой.

МИ1997 является рекомендацией [1] распространяемой на измерительные преобразователи типа “Сапфир”, предназначенные для непрерывного преобразования избыточного давления, разрежения, избыточного давления-разрежения, абсолютного давления и разности давлений в унифицированный токовый сигнал, в том числе на преобразователи разности давлений, используемые для измерения расхода и уровня. Допускается применять данную рекомендацию для поверки других измерительных преобразователей (датчиков).

Основополагающими тенденциями при разработке и совершенствовании программ и методик испытаний являются следующие [1]:

– повышение критерия достоверности результатов измерений при проведении тестовых, технологических или исследовательских испытаний;

– введение новых контролируемых параметров и характеристик;

– увеличение числа внешних факторов, одновременно влияющих на испытуемый датчик;

– введение промежуточных технологических испытаний;

– повышение уровня технологического запаса.

МИ1997 предусматривает выполнение следующих базовых операций во время испытания:

1) Внешний осмотр;
2) Опробование:
   - проверка работоспособности самого преобразователя ЧЭ;
   - функционирование корректора нуля;
   - герметичность преобразователя.
3) Определение основной погрешности преобразователя:
   - сравнение полученного значения с полученным значением образцового преобразователя;
   - основную погрешность проверяют при пяти значениях измеряемой величины.
4) Определение вариации выходного сигнала преобразователя:
   - вариацию выходного сигнала определяют при каждом проверяемом значении измеренного параметра, кроме значений, соответствующих нижнему и верхнему пределам измерений. Вариацию рекомендуется определять при определении основной погрешности.

МИ1997 была разработана для проведения испытаний традиционных датчиков, имеющих ЧЭ преобразующий физическую величину в электрический сигнал. Современные ИД содержат также дополнительные аналоговые и цифровые электрические цепи

преобразования, АЦП и ЦАП, а также алгоритмические и программные средства микроконтроллера, являющегося сердцем ИД. Все эти дополнения также нуждаются в контроле и в исследовании. Также с созданием цифровых датчиков появилась потребность в контроле дополнительных наиболее важных параметров – обнаружение пиковых помех и времени отклика. В МИ1997 эти параметры не подвергались проверкам.

В частности в ИДАД стоят задачи:

– температурной компенсации;
– проверки на достоверность измеренных данных, связанной с обнаружением пиковых помех для обеспечения требуемой достоверности данных;
– компенсации нелинейности;
– компенсации, связанной с дрейфом нуля и гистерезисом.

Решение поставленных задач реализовано алгоритмическим путем, реализацией программы работающей в микроконтроллере цифрового модуля ИДАД. И для проведения испытаний и исследований работоспособности вышеизложенных алгоритмов, а также на правильность реализации этих алгоритмов, необходима разработка новых методик. В связи с тем, что ИДАД относится к новому поколению ИД и содержит цифровые модули обработки и передачи данных, то следует выделить потребность в испытаниях ИДАД по определению:

– времени отклика;
– динамической погрешности измерения.

Новая методика должна в полной мере раскрывать все вероятные ошибки в построении ИДАД, а также обеспечить проверку работоспособности ИДАД в режимах функционирования. Это позволит повысить качество разрабатываемых ИДАД. При проведении испытаний и исследований ИДАД было решено использовать проверки из МИ1997, дополнив их проверками, относящимися к специфике построения ИДАД, изложенными выше.

Библиографический список

1. Гусаров В.В. Тенденции в совершенствовании программ и методик испытаний датчиков давления // Мир измерений. — 2002. — № 7-8. — С. 37-43.

2. МИ 1997-89 // Государственная система обеспечения единства измерений – Преобразователи давления измерительные. Методика поверки. / Москва, 1989. – 32 с.

Получено 01.06.07

УДК 681.323

ВЫСОКОПРОИЗВОДИТЕЛЬНЫЕ РАСПРЕДЕЛЕННЫЕ ИНТЕЛЛЕКТУАЛЬНЫЕ МИКРОКОМПЬЮТЕРНЫЕ СИСТЕМЫ МОНИТОРИНГА СЛОЖНЫХ ДИНАМИЧЕСКИХ ОБЪЕКТОВ

О.Н.Пьявченко
Технологический институт федерального государственного образовательного учреждения высшего профессионального образования «Южный федеральный университет» в г.Таганроге, кафедра микропроцессорных систем

В работе предлагается при создании перспективных распределенных интеллектуальных микрокомпьютерных систем мониторинга разнесенных в пространстве быстродействующих сложных динамических объектов (процессов) выбирать наиболее подходящее среди возможных разнообразных архитектурных решений.

В техногенном обществе по мере развития информационно-технологической революции возрастает роль микрокомпьютерных систем мониторинга, обеспечивающих управление и безопасность функционирования различных сложных динамических объектов во всех сферах человеческой деятельности. При этом при наличии прогресса в области нанотехнологий открываются, действительно, безграничные возможности для создания и применения систем мониторинга с различными архитектурой и техническими характеристиками.

Цель настоящей работы – обратить внимание на возможность использования многообразных архитектурных решений при создании перспективных распределенных интеллектуальных микрокомпьютерных систем (РИМКС) мониторинга разнесенных в пространстве быстродействующих сложных динамических объектов (процессов).

РИМКС мониторинга представляет собой построенную на основе микропроцессоров и микроконтроллеров совокупность программно-аппаратных компьютерных средств, целевой функцией которой является представление пользователю и/или в информационную систему более высокого уровня информации о состоянии наблюдаемого (контролируемого) объекта, полученной в результате сбора и интеллектуальной обработки сигналов датчиков [1,2,3].

При этом ***РИМКС мониторинга*** принимает информацию от датчиков объекта (процесса), представляет ее в форме удобной для хра-

нения и обработки, обрабатывает по заданным алгоритмам, оценивает состояние объекта, выдает результаты на устройства отображения и посылает их в коммуникационные каналы.

Информация о состоянии наблюдаемого объекта формируется в виде
информационного объекта. В состав информационного объекта могут включаться:

1) массивы цифровых значений переменных и параметров;
2) описания состояний контролируемых переменных и параметров;
3) граничные условия изменения переменных состояния;
4) заложенные при проектировании формы отображения состояний объекта и его переменных;
5) звуковые, световые и др. сигналы, которые должны формироваться при переходе состояния объекта или его переменной в соответствующую область;
6) рекомендации оператору, регламентирующие его действия и т.д.

В зависимости от габаритов и расположения в пространстве различают распределенные и локальные объекты наблюдения.

Распределенный в пространстве объект наблюдения состоит из локальных объектов (ЛО). Состояния распределенного объекта характеризуются информационными объектами ЛО, которые формируются локальными информационными микрокомпьютерными системами ***(ЛИМКС)*** [4].

Функциональные особенности ***РИМКС мониторинга***:

1) наличие большого числа удаленных на различные расстояния (от дециметров до километров) от центра обработки информации источников неоднородных аналоговых, цифровых и частотных сигналов;

2) решение совокупности взаимосвязанных задач первичной и вторичной обработки сигналов;

3) необходимость сбора и обработки значительных объемов информации в высоком темпе (в режиме "жесткого") реального времени;

4) обеспечение высокой достоверности и надежности собранных и обработанных данных;

5) оценки состояний как непосредственно измеряемых, так и вычисляемых переменных.

Соответственно в основу построения архитектуры ***РИМКС мониторинга*** могут быть положены следующие принципы:

1) распределение средств обработки информации в соответствии с особенностями топологии и функционирования объекта наблюдения;

2) приближение обработки информации к местам считывания сигналов с объекта;

3) цифровая обработка данных и сетевой обмен результатами обработки;

4) способность воспринимать множество различных сигналов;

5) разнородность средств обработки информации;

6) иерархическая структура организации средств обработки данных;

7) синхронное пошаговое считывание и обработка показаний датчиков.

Очевидно, что на базе этих принципов можно синтезировать множество архитектурных решений.

В качестве примера рассмотрим техническую (технологическую) распределенную интеллектуальную микрокомпьютерную систему мониторинга реализованную в виде трехуровневой иерархии компьютерных средств (рис.1), которые объединяются в систему благодаря специализированному программному обеспечению, цифровым промышленным сетям (ЦПС) и локальным сетям (ЛС) [2,5,6].

Распределенные вычисления и интеллектуальная обработка информации реализуются на всех уровнях системы. На каждом уровне компьютерные средства работают параллельно.

На нижнем уровне рядом с объектом располагаются локальные информационные микрокомпьютерные системы (ЛИМКС), которые содержат интеллектуальные микропроцессорные модули ИММ [7], обеспеченные аналоговыми и цифровыми интерфейсами (АЦИ) для связи с датчиками (Д) локального объекта (ЛО). В ИММ сигналы датчиков после аналоговой обработки оцифровываются и преобразуются в значения локальных переменных, характер изменения которых оценивается и прогнозируется.

Информация о текущих и прогнозируемых значениях и состояниях переменных через ЦПС передается на средний уровень ***РИМКС мониторинга***. На среднем уровне эта информация используется в промышленных функциональных компьютерах (ПФК) для оценок и прогнозов состояний локальных объектов. Наряду с этим ПФК синхронизируют работу ЛИМКС и управляют сбором информации с них.

Рис. 1. Обобщенная архитектура технической (технологической) распределенной интеллектуальной микрокомпьютерной системы мониторинга

На верхнем уровне ***РИМКС мониторинга*** располагаются АРМ операторов, которые строятся на основе персональных компьютеров (ПК) общего и промышленного назначения. На АРМ операторов реа-

лизуются интеллектуальные алгоритмы оценки состояния распределенного объекта и его системы компонентов (групп сильно связанных локальных объектов). При этом результаты анализа представляются в удобной для восприятия и хранения форме, используя банки данных, знаний и специальное программное обеспечение. АРМ соединяются между собой и с промышленными функциональными компьютерами локальной сетью, например сетью Ethernet [8].

Кроме того, при помощи АРМ оператором задаются и управляются общесистемные режимы работы ***РИМКС мониторинга***.

В ***РИМКС мониторинга*** логические потоки данных представляются 5-ю направлениями:

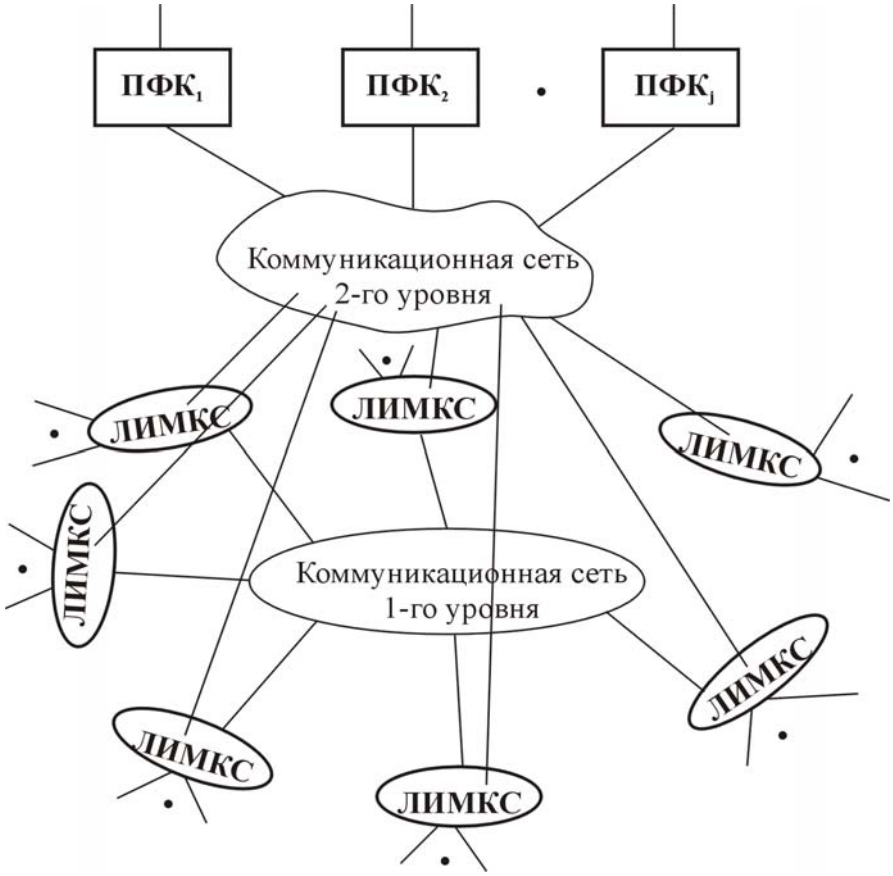
- потоки ввода и обработки сигналов датчиков (первичных преобразователей), которые осуществляются ЛИМКС;
- потоки сбора ПФК данных, полученных в ЛИМКС;
- поток сетевого обмена между ПФК и АРМ;
- поток команд управления от АРМ к ПФК;
- поток команд управления от ПФК к ЛИМКС.

На верхнем уровне оценивается состояние всего объекта (процесса). Оценка выполняется с учетом заданных и/или формируемых ограничений и прогноза состояния объекта (процесса), полученного на основе оценок состояний системных компонентов, а также оценок локальных переменных состояния и параметров.

Для хранения и представления в удобной форме оценок состояния объекта (процесса), системных компонентов, локальных переменных и параметров реализуется подсистема отображения и архивирования. В случае необходимости подобные подсистемы размещаются и на более низких уровнях иерархии.

Технические характеристики и, прежде всего, производительность ***РИМКС мониторинга*** зависят от соответствующих параметров узловых вычислительных средств, так и от структурных особенностей системы и организации в ней процессов сбора и обработки информации сетевого обмена данными и управления в темпе реального времени.

В случае, когда в процессе сбора и обработки информации реализуется обмен данными каждой ЛИМКС с каждой и результаты обработки данных концентрируются в ПФК, в ***РИМКС мониторинга*** реализуется иерархическая распределенная интеллектуальная микрокомпьютерная система сбора и обработки информации (***РИМКС СОИ***) (рис.2).



ПФК – промышленный функциональный компьютер;
ЛИМКС – локальные интеллектуальные микроконтроллерные системы

Рис. 2. Распределенная интеллектуальная микрокомпьютерная система сбора и обработки информации

В ***РИМКС СОИ*** выделяются минимум две коммуникационные сети:

– коммуникационная сеть 1-го уровня, обеспечивающая обмен данными между ЛИМКС;

– коммуникационная сеть 2-го уровня (КС2), при помощи которой сформированные ЛИМКС данные собираются в ПФК.

Коммутационная сеть 1-го уровня (КС1) должна удовлетворять следующим требованиям:

1) реализовать соединения (обмен информацией) каждой ЛМКС с каждой;

2) иметь жестко ограниченное время обмена

$$T_{КС1} < T,$$

где T – длительность шага считывания и обработки данных в РИМКС;

3) обеспечивать возможность синхронного, асинхронного и смешанного обмена данными между ЛИМКС;

4) допускать организацию совмещения межсистемного обмена информацией со сбором и обработкой сигналов в ЛИМКС

$$T_{КС1}\,(1 - \eta_{КС1}) \geq \min T_{СБj},$$

где $\min T_{СБj}$ – минимальные затраты времени сбора и обработки данных в ЛИМКС, а $\eta_{КС1}$ – коэффициент совмещения процессов;

5) минимальные затраты оборудования и высокая надежность при обеспечении необходимой производительности.

В основу построения КС1 могут быть положены различные конфигурации отличающиеся топологиями соединений между ЛИМКС, количеством и типом сетевых каналов в них [2, 5, 6, 9]:

1) цепи;
2) звезды;
3) общие шины:
4) кольца:
5) соединения по полному графу;
6) распределенные коммутаторы, реализованные в мини-ВС МИНИМАКС, мини-ВС СУММА, мини-ВС семейства МИКРОС;
7) соединения по схемам Гиперграфов.

Выбранный вид конфигурации КС1 должен удовлетворять приведенным выше требованиям.

Требования к коммуникационной сети 2-го уровня:

1) реализовать соединения ЛИМКС с $ПФК_j$ по заданной схеме;
2) иметь жесткое ограничение время концентрации данных

$$T > T_{КС2} \approx T_{КС1};$$

3) обеспечивать возможность синхронного, асинхронного и смешанного обмена данными между ЛИМКС;
4) допускать (по возможности по максимуму) совмещение работы с работой КС1

$$T_{КС2} \geq (1 - \eta_{КС2})T_{КС1},$$

где $\eta_{КС2}$ – коэффициент совмещения.

Перечисленные требования отличаются от требований к КС1 пунктами 1, 4.

Конфигурации коммуникационной сети 2 целесообразно строить по схеме древовидных топологий [2,5,6], используя коммутирующие интеллектуальные микроконтроллерные модули (КИММ) для формирования пакетов результаты обработки данных в ЛИМКС. КИММ реализует следующие функции:

1. Асинхронный контролируемый прием данных по *n (n>1)* каналам.

2. Выделение данных, на которые "есть подписка".

3. Отслеживание поступления всех подписанных данных.

4. Компоновка данных в сообщение для передачи на верхний уровень.

5. Фиксация завершения подготовки сообщения.

6. Отправка сообщения наверх без запроса либо после запроса.

Пример простейшей древовидной конфигурации приведен на рис. 3.

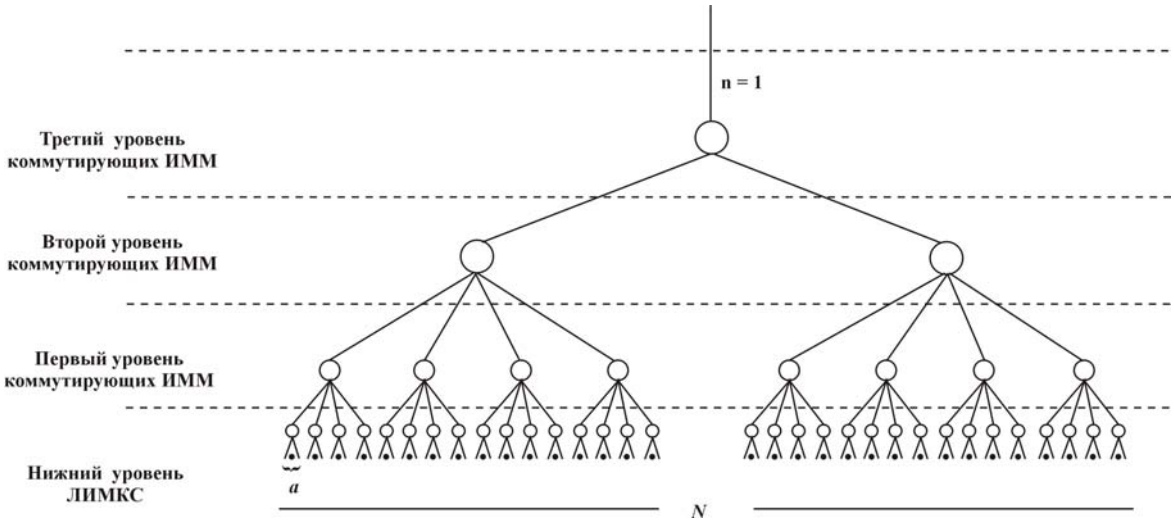


Рис.3. Распределенная интеллектуальная микроконтроллерная система сбора и обработки информации

Количество уровней электронного распределенного коммутатора

$$\alpha = \left[\frac{lg_2 N - lg_2 n}{lg_2 a} \right],$$

где N – количество входных переменных РИМС;

n – количество выходных переменных коммутатора;

a – число входов как ЛИМКС, так и КИММ.

Количество уровней электронного распределенного коммутатора при n = 1

$$\alpha = \left[\frac{lg_2 N}{lg_2 a} \right],$$

где [E] означает округление с избытком неправильной дроби E до целого числа.

В заключение заметим, что из-за ограниченного объема в настоящей работе основной акцент сделан на иллюстрацию возможности построения различных конфигураций ***РИМКС мониторинга*** и не

рассматривались особенности организации и синхронизации сбора и обработки информации и т.д. Однако и в этом случае видны перспективы получения многообразных архитектурных решений при разработках высокопроизводительных ***РИМКС мониторинга*** технических (технологических) быстродействующих сложных динамических объектов.

Литература

1. Пьявченко О.Н., Педошенко А.М, Пцарева М.М. Распределенные интеллектуальные микрокомпьютерные системы: Учеб пособие./ Под ред. О.Н.Пьявченко. – Таганрог: Изд-во ТРТУ, 2004. – 118 с.
2. Кругляк К. Промышленные сети: цели и средства/ СТА 4/2002. – С.6-17.
3. http://www.energoprj.ru.
4. Пьявченко О.Н. Проектирование локальных микрокомпьютерных систем. – Таганрог: Изд-во ТРТУ, 2005. – 238 с.
5. Компьютерные сети. Принципы, технологии, протоколы / В.Г. Олифер, Н.А. Олифер. - СПб: Издательство "Питер", 2000. – 672 с.: ил.
6. Марков Г. Современные коммуникационные сети. Технологии и интерфейсы./ Электронные компоненты, № 4, 2007. – С. 63-69.
7. Пьявченко О.Н. Концептуальное представление о прецизионных интеллектуальных микропроцессорных модулях ввода, измерений и обработки аналоговых сигналов./ Известия ТРТУ – Таганрог: Изд-во ТРТУ, 2007, №3 (75) – С,3-13
8. Севбо В., Орлов А., Лошаков А. Многоканальная распределенная системы синхронного сбора данных "жестко" реального времени, построенная на основе Ethernet-технологий: СТА 3/2007. – С.40-45.
9. Хорошевский В.Г. Архитектура вычислительных систем: Учеб. пособие для вузов. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2005. – 512 с.

Получено 01.06.07

УДК 681.3

ОБЕСПЕЧЕНИЕ УСТОЙЧИВОСТИ ЧИСЛЕННОГО МОДЕЛИРОВАНИЯ ДИНАМИЧЕСКИХ СИСТЕМ

О.А. Дмитриева, Л.П.Фельдман
Донецкий национальный технический университет

У роботі розглядаються проблеми стійкості багатокрокових паралельних блокових методів розв'язання задачі Коши для звичайних диференційних рівнянь, досліджується їх збіжність, виводяться оцінки погрішності.

При численном моделировании динамических систем, описываемых обыкновенными дифференциальными уравнениями, очень часто возникают вопросы, связанные с обеспечением устойчивости решения. Дополнительные трудности появляются при численном решении жестких систем. Это обусловлено необходимостью использования специальных методов, позволяющих выбирать шаг интегрирования исходя лишь из требований точности, а не устойчивости, или из требований сходимости итерационного процесса решения неявных уравнений. К настоящему времени предложен и практически используется целый ряд последовательных методов решения жестких уравнений [1]. В то же время проблема разработки и обоснования параллельных методов и алгоритмов, ориентированных на эффективное использование в многопроцессорных системах, все еще остается открытой. Причем одним из вариантов решения этой проблемы является направление, определяющее разработку новых методов и алгоритмов.

В настоящей работе рассматривается устойчивость параллельных методов решения жестких систем. Для задачи Коши

$$\frac{dx}{dt} = f(t,x), \;\; x(t_0) = x_0 \qquad (1)$$

уравнения одношагового разностного метода [2] для блока *n,* содержащего *k* точек, можно записать в виде

$$\frac{u_{n,i} - u_{n,0}}{i\tau} = b_i F_{n,0} + \sum_{j=1}^{k} a_{i,j} F_{n,j}, \;\; i = \overline{1,k}, \; n = 1,2,... \qquad (2)$$

где $F_{n,i} = f(t_{n,i}, u_{n,i})$, *i*- номер точки в блоке, τ - шаг интегрирования.

Условие устойчивости Далквиста [1] для разностных уравнений (2) выполняется, так как для каждого i характеристическое уравнение разностного метода имеет вид

$$\lambda^{i} = 1,\ i = \overline{1,k}, \qquad (3)$$

все i простых корней которого лежат на окружности единичного радиуса, и наивысший порядок аппроксимации (2) равен $p = k+1$ [2]. Таким образом выполнены условия сходимости решения разностной задачи при $\tau \to 0$ к решению исходной задачи (1) на конечном отрезке $0 \le nk\tau \le T$. Однако выполнение условий устойчивости разностных уравнений по Далквисту является недостаточным при проведении практических расчетов на больших интервалах t, так как не гарантирует абсолютную устойчивость разностного метода. Исследование устойчивости одношаговых блочных методов (2) в настоящей работе проводилось на модельном одномерном уравнении

$$\frac{dx}{dt} = \lambda x,\ t > 0, \qquad (4)$$

Устойчивость двухточечного метода

$$\begin{aligned} u_{n+1} &= u_n + \mu * (5u_n + 8u_{n+1} - u_{n+2}) / 12 \\ u_{n+2} &= u_n + \mu * (u_n + 4u_{n+1} + u_{n+2}) / 12 \end{aligned} \qquad (5)$$

обеспечивало выполнение следующих условий

$$|q_1| = \left| \frac{6 - \mu^2}{2(3 - 3\mu + \mu^2)} \right| \le 1 \ и \ |q_2| = \left| \frac{3 + 3\mu + \mu^2}{3 - 3\mu + \mu^2} \right| \le 1, \qquad (6)$$

где $\mu = \lambda \tau$.

Можно показать, что двухточечный метод устойчив для $\mu \le 0$, т.е. он абсолютно устойчив, если устойчиво решение исходного дифференциального уравнения. Для доказательства необходимо определить множество точек комплексной плоскости, в которых выполняются условия $|q_1| \le 1$, $|q_2| \le 1$. Границей области является множество точек, для которых $|q_1| = 1$. Положив $q_1 = e^{i\varphi}$, получим границу устойчивости первого уравнения.

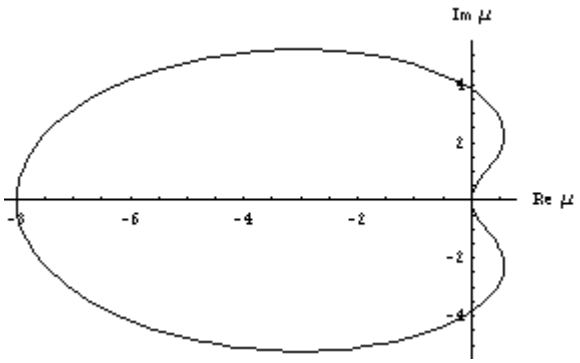


Рис. 1. Граница устойчивости первого уравнения метода

Для точек, расположенных внутри этой кривой, выполнено условие $|q_1|>1$, поэтому область устойчивости первого уравнения метода представляет собой внешность кривой. Областью устойчивости второго уравнения метода является левая полуплоскость Re(μ)<0, так как для действительных отрицательных μ имеет место $|q_2|<1$. Поскольку области устойчивости обоих уравнений содержат левую полуплоскость (μ <0), то метод является А-устойчивым.

В работе доказана $A(\alpha)$- устойчивость четырехточечного метода

$$\begin{aligned}
u_{n+1} &= u_n + \mu*(251u_n + 646u_{n+1} - 264u_{n+2} + 106u_{n+3} - 19u_{n+4})/720\\
u_{n+2} &= u_n + \mu*(29u_n + 124u_{n+1} + 24u_{n+2} + 4u_{n+3} - u_{n+4})/90\\
u_{n+3} &= u_n + 3\mu*(9u_n + 34u_{n+1} + 24u_{n+2} + 14u_{n+3} - u_{n+4})/80\\
u_{n+4} &= u_n + 2\mu*(7u_n + 32u_{n+1} + 12u_{n+2} + 32u_{n+3} + 7u_{n+4})/45
\end{aligned} \qquad (7),$$

которая обеспечивается выполнением следующих условий

$$\begin{aligned}
|q_1| &= \left|\frac{60 - 60\mu + 15\mu^2 + 5\mu^3 - 3\mu^4}{60 - 120\mu + 105\mu^2 - 50\mu^3 + 12\mu^4}\right| \le 1,\\
|q_2| &= \left|\frac{60 - 15\mu^2 + 2\mu^4}{60 - 120\mu + 105\mu^2 - 50\mu^3 + 12\mu^4}\right| \le 1,\\
|q_3| &= \left|\frac{60 + 60\mu + 15\mu^2 - 5\mu^3 - 3\mu^4}{60 - 120\mu + 105\mu^2 - 50\mu^3 + 12\mu^4}\right| \le 1,\\
|q_4| &= \left|\frac{60 + 120\mu + 105\mu^2 + 50\mu^3 + 12\mu^4}{60 - 120\mu + 105\mu^2 - 50\mu^3 + 12\mu^4}\right| \le 1
\end{aligned} \qquad (8).$$

Границей области устойчивости 1-го уравнения является множество точек, для которых $\left|q_1\right|=1$. Положив $q_1 = e^{i\varphi}$, получим границу устойчивости.

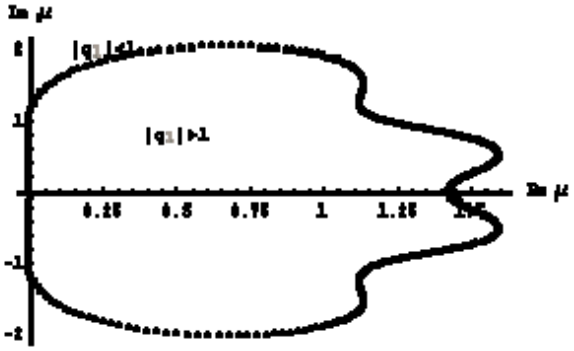


Рис. 2. Граница устойчивости первого уравнения метода

Повторив последовательность действий для оставшихся уравнений системы, получим

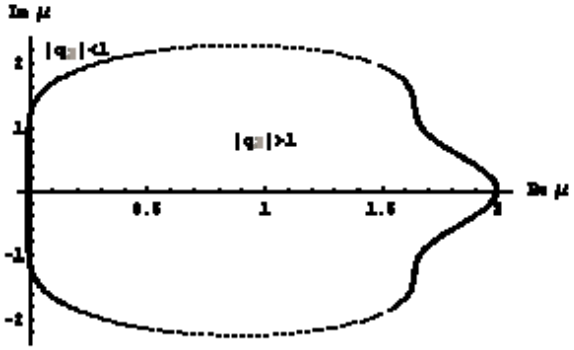


Рис. 3. Граница устойчивости второго уравнения метода

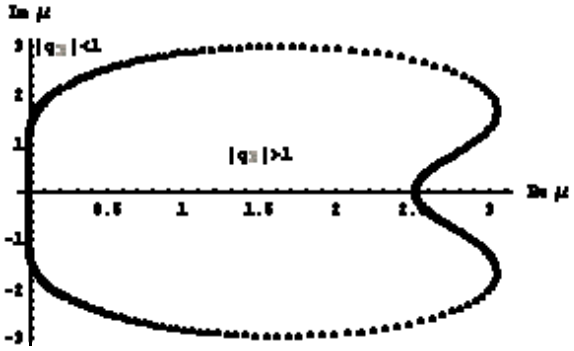


Рис. 4 . Граница устойчивости третьего уравнения метода

Областью устойчивости четвертого уравнения метода является мнимая ось, т.е. множество точек, для которых $\mu=i\alpha$, где α –

произвольное действительное число, тогда областью является левая полуплоскость Re(μ)<0.

Таким образом можно утверждать, что одношаговый четырехточечный метод является A(α) устойчивым.

Решение нелинейной системы уравнений (1) может быть осуществлено либо с помощью итерационного процесса, который позволяет проводить вычисления параллельно для каждого узла блока, либо с помощью метода Ньютона. Полученные для обыкновенных дифференциальных уравнений результаты могут быть распространены и на системы.

Представленные в работе методы являются многошаговыми, т.е. новые значения выражаются через найденные ранее. Таким образом, для многошаговых методов необходимы решения в предшествующих точках, тогда как на первом шаге интегрирования известно лишь начальное значение для уравнения или вектор начальных значений для системы ДУ. При нахождении стартовых значений существенным является то, что они должны быть вычислены с той же степенью точности, с которой будет работать многошаговый метод. При осуществлении сближения одношаговым методом может возникнуть ситуация, когда число операций, затрачиваемых на разгонку, может превысить трудоемкость решения всей задачи. К тому же, такой алгоритм для общего случая является последовательным. В работе предлагаются 2 комбинированных алгоритма для определения начального отрезка. Первый основан на первоначальном использовании любого одношагового метода (например, Рунге-Кутты), но не на всем требуемом промежутке, а только в *2k-1* точках. Выбрав из рассчитанных точек *k* и, удвоив шаг, можно начать счет многошаговым методом. Продвижение вдоль начального отрезка будет обеспечиваться постоянным удвоением шага. Еще один способ рекомендует однократное применение одношагового блочного метода, а затем продолжение вычислений многошаговым методом, удваивая шаг при переходе к расчету значений в новом блоке.

Коэффициент, связывающий размерности шагов используемых методов, должен быть пропорционален степени двойки, так как проще сделать лишнюю итерацию на этапе разгонки, чем потом выполнять расчеты с шагом, заведомо меньше допустимого. Комбинированные подходы позволяют сократить количество точек, в которых значения функции будут вычисляться последовательно, до величины $2(k-1)$, или же вовсе избавиться от последовательных вычислений. К тому же, поскольку значения функции в каждой точке блока вычисляются

одновременно, это еще, примерно, в k раз ускоряет вычисление разгоночных значений.

Рассмотренные методы могут использоваться при численном решении жестких систем обыкновенных дифференциальных уравнений. Предложенная методика позволяет также исследовать устойчивость параллельных многошаговых методов [2] с любым числом точек в блоке. Проведенные численные решения одношаговыми блочными методами тестовых жестких систем практически подтвердили их надежность и эффективность.

Литература

1. Современные численные методы решения обыкновенных дифференциальных уравнений. - М.: Мир, 1979. – 312 с.
2. Фельдман Л.П., Дмитриева О.А. Эффективные методы распараллеливания численного решения задачи Коши для обыкновенных дифференциальных уравнений.// Математическое моделирование, том 13, № 7, 2001. – С. 66-72.

Поступление в редакцию 15.05.2007 г.

УДК 681.5

ПАРАЛЛЕЛЬНЫЙ РЕКУРСИВНЫЙ АЛГОРИТМ БЫСТРОГО МАТРИЧНОГО УМНОЖЕНИЯ

Л.П.Фельдман, И.А. Назарова, А.Е. Шматько
Донецкий национальный технический университет

В статті розглянуто паралельний алгоритм на основі рекурсивного алгоритму швидкого матричного множення. Розроблені обчислювальні схеми відображення методу на паралельні структури різної топології. Отримані порівняльні характеристики з традиційними алгоритмами, проведено чисельний експеримент.

Матричное умножение – это базовая операция линейной алгебры и доминирующая вычислительная часть многих научных приложений. Существует множество традиционных параллельных алгоритмов вычисления матричного произведения для плотнозаполненных матриц [1-2]. В этой статье анализируется эффективность параллельного алгоритма быстрого матричного умножения с использованием модификации рекурсивного алгоритма Штрассена-Винограда.

В оригинале алгоритм Штрассена-Винограда – это алгоритм умножения блочных матриц половинного размера, где каждый блок квадратный, т.е. размерности матриц должны быть четными числами. Метод Штрассена-Винограда состоит из 7 блочных умножений матриц и 15 блочных сложений\вычитаний матриц (рис. 1).

Идея Штрассена может быть применена рекурсивно для нахождения произведений блоков матриц $M_i, i=\overline{1,7}$. Если исходные матрицы A и B имели размерность $m \times m$, то алгоритм быстрого умножения можно использовать многократно, получая на самом нижнем уровне получим блоки 1×1. Однако нет никакой необходимости опускаться вниз до уровня блоков единичного порядка. При достаточно малых размерах блока ($k < k_{min}$) может оказаться выгодным вычислять блоки, используя стандартный алгоритм.

Вычислительная сложность предложенной схемы алгоритма быстрого умножения определяется функцией порядка исходных матриц и минимального порядка умножаемых блоков: m, m_{min}. Пусть при вычислении матричного умножения алгоритм Штрассена рекурсивно вызывался d раз, тогда порядок умножаемых матриц равен $m_{min} = m / 2^d$. На первом шаге алгоритм предусматривает 7 обращений

к самому себе с матрицами порядка $m/2$ и 15 операций типа сложение для матриц того же порядка. Далее идет развертка рекурсии до достижения минимального размера блока и умножение блоков по традиционному алгоритму МУ.

$$
\begin{array}{lll}
S_1 = A_{21} + A_{22} & M_1 = S_2 S_6 & T_1 = M_1 + M_3 \\
S_2 = S_1 - A_{11} & M_2 = A_{11} B_{11} & T_2 = T_1 + M_4 \\
S_3 = A_{21} - A_{12} & M_3 = A_{12} B_{21} & T_3 = M_5 + M_6 \\
S_4 = A_{12} - S_2 & M_4 = S_3 S_7 & C_{11} = M_2 + M_3 \\
S_5 = B_{12} - B_{11} & M_5 = S_1 S_5 & C_{12} = T_1 + T_3 \\
S_6 = B_{22} - S_5 & M_6 = S_4 B_{22} & C_{21} = T_2 - M_7 \\
S_7 = B_{22} - B_{12} & M_7 = A_{22} S_8 & C_{22} = T_2 + M_5 \\
S_8 = S_6 - B_{12} & &
\end{array}
$$

$$
A = \left\langle \frac{A_{11}}{A_{21}} \middle| \frac{A_{12}}{A_{22}} \right\rangle, B = \left\langle \frac{B_{11}}{B_{21}} \middle| \frac{B_{12}}{B_{22}} \right\rangle, C = \left\langle \frac{C_{11}}{C_{21}} \middle| \frac{C_{12}}{C_{22}} \right\rangle
$$

Рис. 1. Метод быстрого умножения матриц Штрассена-Винограда

Алгоритм быстрого умножения рекурсивен, поэтому имеется возможность построить полиалгоритм из некоторого традиционного алгоритма умножения матриц на верхнем уровне рекурсии и метода Штрассена на нижнем уровне, и наоборот. Алгоритм Штрассена является блочным, поэтому естественно комбинировать его со стандартным алгоритмом, также использующим блочное разбиение данных.

Пусть имеется мультикомпьютер из p^2 процессоров, объединенных коммуникационной сетью топологии тор. Исходные матрицы распределены по сетке процессоров на блоки $k \times k$, $k = m/p$, количество блоков равно $q^2 = p^2$. Построим полиалгоритм из блочного систолического алгоритма УМ между процессорами и серии применения метода Штрассена на каждом процессоре. Блоки исходных матриц и результата с координатами $<i, j>$ хранятся в соответствующем процессоре с теми же координатами. Предварительно по вычислительной схеме блочного систолического умножения выполняются:

1) $i \leftarrow A -$ косой сдвиг влево по строкам для блоков матрицы A;

2) $B \uparrow^j -$ косой сдвиг вверх по столбцам для блоков матрицы B.

На каждом из p шагов алгоритма производится умножение блоков матриц A и B, хранимых в процессоре с номером $<i, j>$ и сложение с уже вычисленным значением блока матрицы результата, расположенным на этом же процессоре C_{ij}. Для первого шага это значение равно нулевому блоку. Затем производится одиночный сдвиг влево по строкам параллельно для блоков матрицы $A: A_{ij} \leftarrow A_{i,j+1}$ и одиночный сдвиг вверх по столбцам для блоков матрицы $B: B_{ij} \leftarrow B_{i+1,j}$. Умножение блоков матриц выполняется внутри одного процессорного элемента, что позволяет избежать дополнительных пересылок данных, по рекурсивному алгоритму Штрассена. На нижнем уровне рекурсии применяется стандартный алгоритм умножения матриц, глубина рекурсии равна d. Разработанный алгоритм является масштабируемым для любого числа процессорных элементов и порядка исходных матриц.

Общее время выполнения арифметических операций для полиалгоритма равно:

$$T_{p,comp} = \left[2\left(\frac{7}{8}\right)^d \frac{m^3}{p^2} + \left(5\left(\frac{7}{4}\right)^d - 4 \right) \frac{m^2}{p} \right] t_{op}.$$

Время обменных операций для описанной схемы определяется, как и для блочного систолического алгоритма, и равно:

$$T_{p,comm} = 4(p-1) \cdot \left(t_s + \frac{m^2}{p^2} \cdot t_w \right).$$

Определение потенциальных и реальных характеристик параллелизма осуществлялось с помощью пакета *Mathematica®* (Wolfram Research Inc.):

$$E = \frac{T_1}{T_p \cdot p^2}, T_1(m) = \left[2\left(\frac{7}{8}\right)^d m^3 + \frac{15}{4} m^2 \left(1 + \frac{7}{4} + \frac{7^2}{4^2} + \ldots + \frac{7^{d-1}}{4^{d-1}} \right) \right] t_{op}.$$

Очевидно, что динамические характеристики параллельных вычислительных схем МУ зависят от соотношения между числом процессоров и размерностью матриц. Для алгоритма Штрассена и полиалгоритмов на его основе существенным параметром является величина глубины рекурсии, d.

Анализ аналитических выражений, характеризующих выполнение параллельных алгоритмов, а также проведенный численный эксперимент, позволяют сделать следующие выводы:

1) предложенная параллельная вычислительная схема полиалгоритма на основе быстрого умножения Штрассена имеет лучшее время выполнения по сравнению с блочным систолическим алгоритмом в $(8/7)^d$ раз для матриц больших размерностей и невысокой глубине рекурсии;

2) высокий уровень рекурсии отрицательно сказывается, как на времени выполнения, так и на объеме используемой памяти.

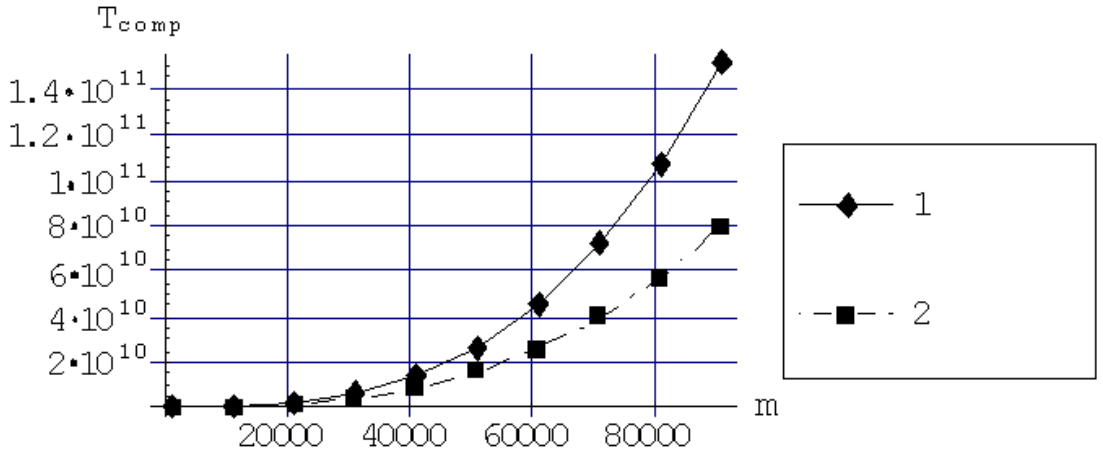


Рис. 2. График зависимости времени реализации параллельных алгоритмов от размерности матриц, $d = 4$
1)блочного систолического, 2) Штрассена+блочный систолический

Аналитические исследования и численный эксперимент, показали, что применение комбинации алгоритма Штрассена и блочного систолического умножения матриц эффективно для матриц больших размерностей и малой глубины рекурсии. Недостатками этого подхода являются большие затраты памяти и несколько худшая численная устойчивость, хотя и достаточная для большинства практических задач. Перспективным направлением дальнейших исследований является разработка, исследование эффективности, определение области применения полиалгоритмов метода Штрассена на основе других традиционных алгоритмов умножения матриц.

Литература

1.Деммель Дж. Вычислительная линейная алгебра. Теория и приложения. Пер. с англ. – М.: Мир, 2001. – 430с.

2. Голуб Д., Ван Лоун Ч. Матричные вычисления: Пер. с англ. – М.: Мир, 1999. – 548с.

Получено 20.05.07

УДК 681.3

ИСПОЛЬЗОВАНИЕ ДИНАМИЧЕСКОЙ СИНХРОНИЗАЦИИ В СИСТЕМЕ РАСПРЕДЕЛЕННОГО ЛОГИЧЕСКОГО МОДЕЛИРОВАНИЯ

Ладыженский Ю.В., Попов Ю.В., Тесленко Г.А.
Кафедра ПМиИ, ДонНТУ

Рассмотрена структура программной системы для распределенного логического моделирования. Предложена организация подсистемы распределенного моделирования с динамическим типом синхронизации. Предложены алгоритмы переключения протокола синхронизации.

Введение. Цифровые системы логического управления на основе микросхем, программируемых пользователем, широко применяются в промышленности. Моделирование цифровых систем является важным этапом проектирования, обеспечивающим проверку корректности функционирования. Высокая размерность и сложность логических схем приводит к большим затратам времени и памяти при последовательном моделировании.

Использование алгоритмов параллельного и распределенного вычислений [1, 2] является перспективным направлением для ускорения моделирования. Создание на их основе эффективных по быстродействию параллельных и распределенных программных и аппаратных средств моделирования цифровых устройств является актуальной научно-технической проблемой.

Задачей работы является исследование динамических алгоритмов синхронизации с целью повышения быстродействия распределенного логического моделирования цифровых систем.

Динамический протокол синхронизации. Особенностью динамического протокола, используемого в системе моделирования, является возможность переключения от консервативной синхронизации к оптимистической и обратно в процессе моделирования. Логический процесс (ЛП) автоматически изменят свой тип на консервативный в случае возникновения частых откатов, либо на оптимистический в случае частых блокировок при моделировании. Переключение в оптимистический режим синхронизации выполняется путем установки поля типа логического процесса. Переключение в консервативный режим предполагает в начале использование промежуточного режима работы, в котором

обрабатываются только безопасные и антисообщения. Так как при оптимистической синхронизации в списке состояний могут находиться события, которые потенциально могут привести к откату в других логических процессах, то перед переключением необходимо гарантировать корректное состояние логического процесса. В промежуточном режиме выполняется откат ошибочных событий с помощью антисообщений, новые состояния не сохраняются в очередь, т.к. обрабатываются только безопасные события. В результате очередь состояний очищается, что гарантирует корректное состояние логического процесса перед его переключением в консервативный режим.

С целью повышения эффективности моделирования динамический протокол синхронизации может быть реализован аппаратно[3, 4].

Структура программной системы. Программная система для распределенного логического моделирования состоит из трех подсистем [5]: подсистема подготовки схемы и входного воздействия для моделирования, подсистема моделирования и подсистема анализа результатов моделирования (рисунок 1).

Подсистема подготовки схемы и входного воздействия для моделирования предназначена для создания и редактирования схем и входного воздействия.

В подсистеме моделирования каждый моделирующий процессор выполняет моделирование заданного участка схемы с заданным воздействием. В результате моделирования создается отчет, который является входными данными для подсистемы анализа результатов моделирования. Эта подсистема позволяет построить временные диаграммы, диаграммы причинно-следственных связей и получить статистическую информацию из отчетов моделирований.

Программная система позволяет последовательно моделировать цифровые системы, либо выполнять распределенное моделирование с использованием консервативного, оптимистического, динамического алгоритмов.

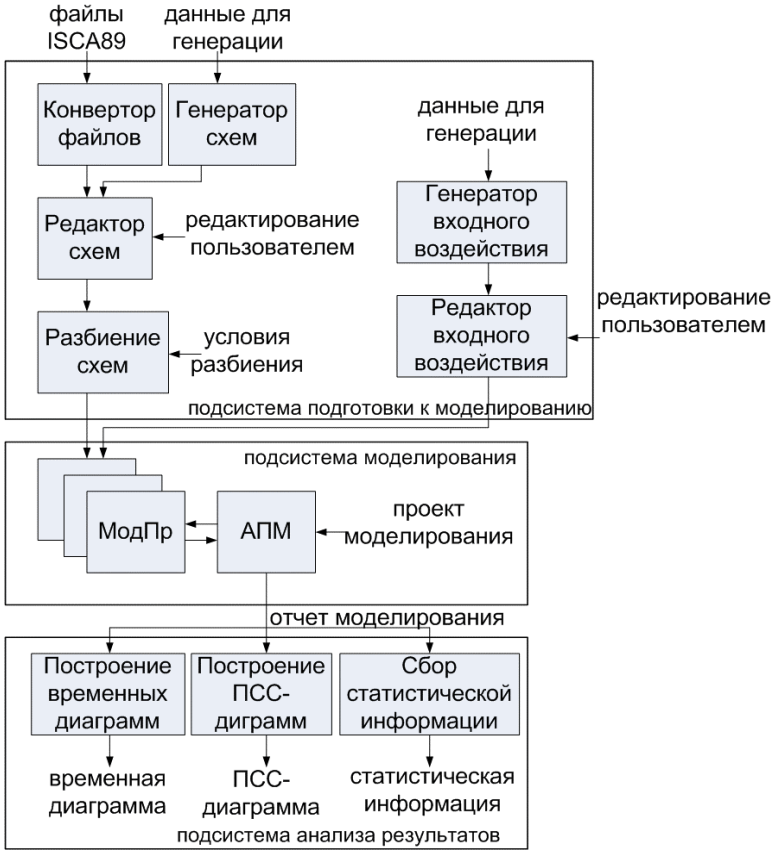


Рис. 1 – Структура программной системы для распределенного моделирования

Организация подсистемы распределенного моделирования. Подсистема моделирования состоит из следующих программных компонент (рисунок 2):

- администратор процесса моделирования (АПМ), специальная программа для управления моделирующими процессами. Администратор позволяет осуществлять выбор схемы, входного воздействия, а также задавать необходимые опции для моделирования. При моделировании синхронизация осуществляются только средствами моделирующих процессоров, без участия АПМ. После завершения моделирования администратор собирает отчеты всех моделирующий процессоров. Для обмена данными между администратором процесса моделирования и моделирующими процессами может использоваться локальная сеть или Интернет[6].

- множество моделирующих процессоров (МодПр), каждый из которых реализует алгоритм синхронизации вычислений.

Моделирующий процессор содержит такие объекты: сервисный модуль, координаторы процесса моделирования, библиотеку функций элементов схемы.

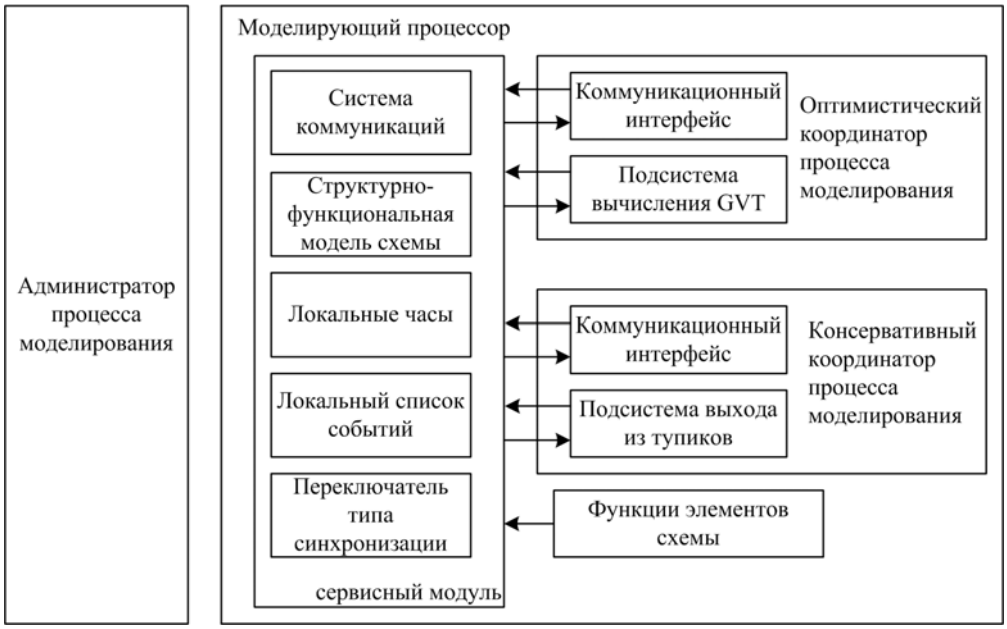


Рис. 2 – Структура подсистемы распределенного моделирования

Общие для разных протоколов синхронизации объекты «система коммуникаций», «локальные часы», «локальный список событий», «структурно-функциональная модель схемы» и «переключатель типа синхронизации» содержатся в сервисном модуле.

Локальный список событий содержит все события, которые возникнут в моделируемой системе. События в этом списке упорядочены в порядке увеличения виртуального времени их наступления. Локальные часы показывают текущее виртуальное время. Структурно-функциональная модель схемы содержит структуру моделируемой цифровой системы и значения сигналов во всех узлах схемы.

Переключатель типа синхронизации во время моделирования выбирает оптимальный протокол синхронизации с целью сокращения времени моделирования.

Координатор процесса моделирования (КПМ) управляет порядком работы всех объектов моделирующего процессора, содержит цикл обработки событий. В подсистеме моделирования в один момент времени может работать только один КПМ, консервативный или оптимистический. В состав каждого из КПМ входит коммуникационный интерфейс, предназначенный для управления глобальным порядком обработки событий.

Консервативный КПМ содержит коммуникационный интерфейс и подсистему выхода из тупиков. Консервативный коммуникационный интерфейс включает в себя канальные часы,

входные, выходные буферы событий для каждого канала связи с остальными МодПр.

Оптимистический КПМ содержит коммуникационный интерфейс и подсистему вычисления глобального виртуального времени GVT. В оптимистическом коммуникационном интерфейсе содержатся часы, показывающие глобальное виртуальное время, входной и выходной буфер, входная, выходная очереди событий, и список всех состояний системы.

КПМ и сервисный модуль образуют сетевой моделирующий процессор (МодПр). Один МодПр соответствует одному логическому процессу и запускается на отдельном компьютере.

Алгоритмы переключения типа синхронизации в подсистеме моделирования. Динамический тип синхронизации реализуется за счет переключения в процессе моделирования между оптимистическим и консервативным алгоритмами [7]. Для переключения типа синхронизации необходимо запустить соответствующий координатор процесса моделирования. На рисунках 3 и 4 показаны алгоритмы переключения протоколов синхронизации в подсистеме распределенного моделирования из консервативного в оптимистический и обратно.

КПМ представляет собой вычислительный поток операционной системы, запускаемый из сервисного модуля. При запуске МодПр в памяти динамически создаются все его объекты, включая консервативный и оптимистический координаторы процесса моделирования в виде двух отдельных вычислительных потоков. По умолчанию запускается КПМ, указанный в проекте моделирования, КПМ другого типа находится в «спящем» состоянии.

Переключение консервативного МодПр в оптимистический выполняется следующим образом. Инициатор переключения рассылает всем сообщения о необходимости переключения. Далее, все МодПр сохраняют состояние своего коммуникационного интерфейса (внешние события входной и выходной очередей). Затем моделирование останавливается, путем остановки вычислительного потока КПМ. Каждый МодПр рассылает всем подтверждения о готовности к переключению типа синхронизации. После принятия подтверждения от всех МодПр выполняется запуск и установка начальных параметров оптимистического КПМ. Происходит изменение состояния коммуникационного интерфейса (заполнение входной, выходной очередей внешними событиями). Далее

Рис. 3 – Алгоритм переключения консервативного протокола синхронизации в оптимистический

Рис. 4 – Алгоритм переключения оптимистического протокола синхронизации в консервативный

моделирование продолжается в соответствии с оптимистическим типом синхронизации.

Если МодПр решает стать консервативным, он также рассылает всем сообщения о необходимости переключения. После получения такого сообщения МодПр переходят в промежуточный режим работы, в котором обрабатываются только безопасные и антисообщения. Далее алгоритм соответствует описанному выше.

Заключение. Рассмотрена структура существующей программной системы для распределенного логического моделирования. Предложена организация подсистемы распределенного моделирования с динамическим типом синхронизации. Описаны алгоритмы переключения типа синхронизации.

Использование алгоритма динамической синхронизации позволяет повысить эффективность распределенного логического моделирования по сравнению с классическими алгоритмами.

В дальнейшем планируется провести исследование критерия переключения типа синхронизации.

Литература

1. C.J.R. Shi, D.Lungeanu. Distributed simulation of VLSI circuits via lookahead-free self-adaptive optimistic and conservative synchronization. In Proc. ICAAD, pages 500-504, Nov 1999.
2. D.Lungeanu and C.-J.R. Shi. Parallel and distributed vhdl simulation. In Proc. DATE, pages 658-662, March 2000.
3. Ладыженский Ю.В., Тесленко Г.А. Аппаратный метод повышения эффективности алгоритмов распределенного логического моделирования цифровых систем. // Наукові праці Донецького національного технічного університету. Серія: «Обчислювальна техніка та автоматизація». Випуск 106 – Донецьк: ДонНТУ, 2006. – 220с. – С.77-81.
4. Ладыженский Ю.В., Тесленко Г.А. Аппаратные методы повышения эффективности распределенного логического моделирования. // Інтернет-Освіта-Наука-2006, п’ята міжнародна конференція ІОН-2006, 10-14 жовтня, 2006. Збірник матеріалів конференції. Том 2. – Вінниця: УНІВЕРСУМ-Вінниця, 2006. – 420с. –С.746-749.
5. Ladyzhensky Y.V., Popoff Y.V. Software system for event-driven logic simulation // IEEE EWDWT, Odessa, September 15-19, 2005, p.119-122
6. Ladyzhensky Y.V., Popoff Y.V. Architecture of internet access to distributed logic simulation system // IEEE EWDWT, Sochi, September 15-19, 2006, p.339-343
7. Ладыженский Ю.В., Попов Ю.В., Тесленко Г.А. Программная система для распределенного логического моделирования с динамическим протоколом синхронизации // Радиоэлектронные и компьютерные системы. – 2007. - №8(27). – С.25-29.

Получено 01.06.07

УДК 681.326

МОДЕЛИРОВАНИЕ И ТЕСТИРОВАНИЕ НЕЙРОМОРФНЫХ МИКРОЧИПОВ НА ПЕРЕКЛЮЧАТЕЛЬНОМ УРОВНЕ

А.И. Андрюхин
Донецкий национальный технический университет

Розглядаються проблеми моделювання та діагностування нейроподібних мікрочипів, зроблених з МОН-структур.

Основой будущей вычислительной техники и искусственного интеллекта будут нейрокомпьютерные системы. Базовой технологией их аппаратной реализации в ближайшие 10-15 лет, невзирая на исследования и разработку RTD, SFT, других наноэлектронных элементов, останется МОП-технология [1].

Идеальным решением реализации сенсорных систем восприятия на базе твердотельной электроники являются в настоящее время аналоговые СБИС и этот вопрос рассматривается в различных работах, из которых укажем [2-4]. Перспективные направления развития элементной базы представлены в [5].

Создание нейроморфных микрочипов представляет собой отображение (морфинг) нервных связей на кремниевые электронные цепи. Первые значительные результаты связаны с именами Карвера Мида (Carver Mead) и М.Маховальд, которые попытались изготовить сетчатку из кремния и воспроизвели три из пяти ее слоев. В дальнейших исследованиях моделировали другие основные части зрительной и слуховой систем. В 2001 г. изготовили все пять слоев сетчатки и смоделировали визуальные сообщения, посылаемые мозгу выходными нейронами сетчатки. Этот кремниевый чип сетчатки, Visio1, воспроизводит реакцию основных четырех типов клеток сетчатки, которые вместе составляют около 90% оптического нерва.

Основные математические модели нейронных структур

Нейрон – это сложная система, которая, согласно значений внешних и внутренних параметров, демонстрирует разнообразные виды электрической активности, выделяет специальные химические вещества — нейромедиаторы, реагирует на их воздействие. Состояние нейрона характеризуется значениями значительного числа переменных, как мембранный потенциал V, различные мембранные токи (калиевый, натриевый, ток кальция и др.), концентрации нейромедиаторов и т.д. Однако возможно для большинства проблем

нейроморфинга использовать три переменные, которые меняются во времени: мембранным потенциалом *V(t)* и двумя видами тока — быстрым *Iб(t)* и медленным *Iм(t)*, которые влияют на изменение *V(t)*. Остальные переменные можно считать либо параметрами, либо зависимыми переменными. Согласно выбранных переменных нейрон можно рассматривать как электрическую цепь, для которой справедливы классические уравнения Кирхгофа.

Классической моделью электрогенеза нервной клетки является модель Ходжкина_Хаксли(Hodgkin-Huxley).

На основе этой модели появились более простые модификации, как модель Морриса-Лекара (ML), система Фиц Хью - Нагумо (FHN),

двумерная и трехмерная системы Хиндмарш-Розе (3-мерная) (HR2, HR3) и др[6,7]. В нелинейной динамике рассматриваются хаотическое поведение, осцилляции, синхронизации в этих и других нейронных системах.

Реализации нейронных структур на основе КМОП-технологии

Аналоговые реализации аксонов, синапсов, нейронов и мембраны (рис.1,2) рассматриваются в [3,7,8].

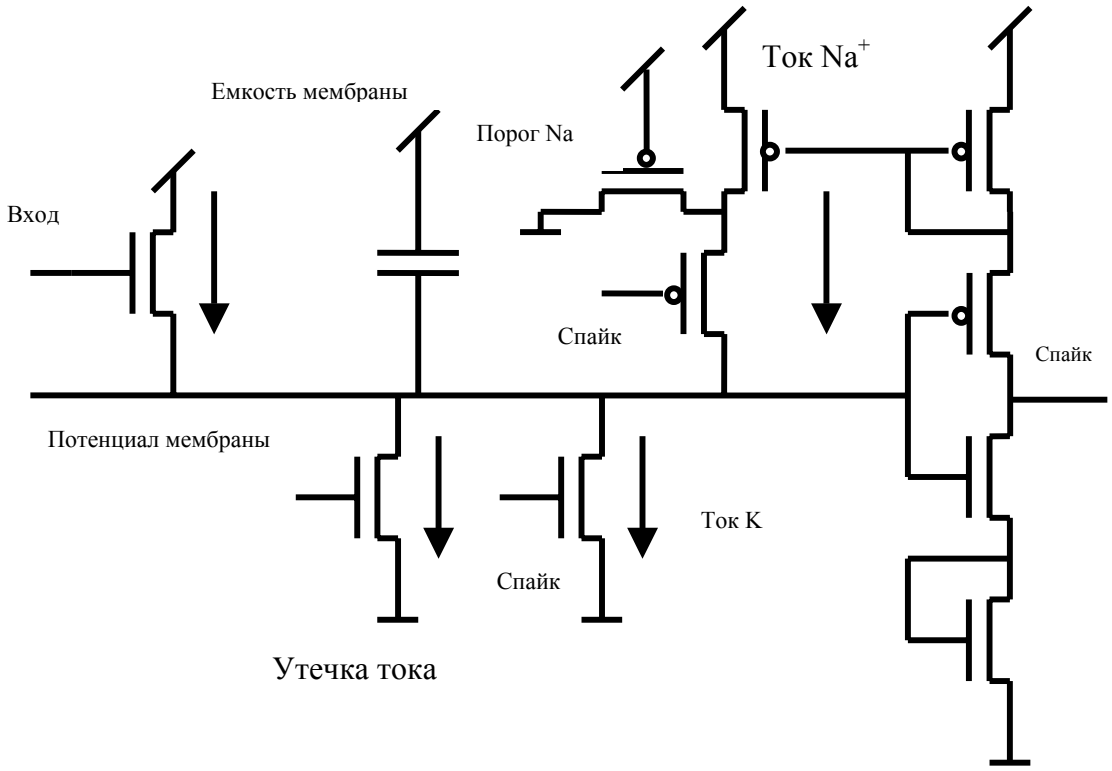


Рис.1.Реализация клеточной мембраны на МОП-структурах

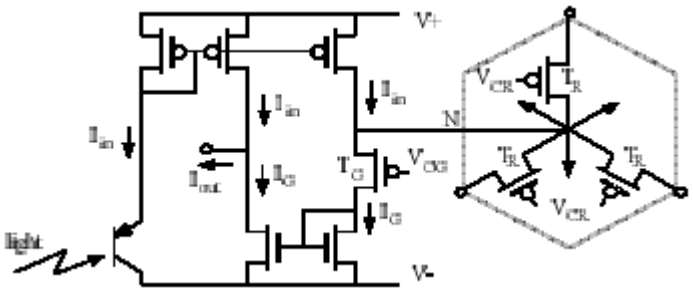


Рис.2.Ячейка искусственной сетчатки

Влияние полевых образований на нейроподобные МОП-структуры

Достаточно медленные в сравнении с психическими явлениями физические характеристики процесса генерации нервного импульса при функционировании мембраны указывают на необходимость учета полевых образований, основой которых служит осцилляторная активность нейронных комплексов. Именно понимание и реализация этой активности приведет к значительному продвижению аппаратной реализации основных функций мозга [8,9].

Моделирование и тестирование нейроморфных микрочипов

Приведенные выше аппаратные реализации основных блоков аналоговых сенсорных нейроноподобных систем, реализованных на МОП-структурах, моделируются с требуемой точностью на основе переключательного подхода путем выбора подходящего алфавита моделирования. Диагностирование МОП-структур основанное на их моделировании на переключательном уровне, рассматривается в [10-13] .

Заключение

При успешном развития нейроморфинга появится возможность создавать вживляемые кремниевые сетчатки для слепых и звуковые процессоры для глухих, а также практически эффективные визуальные, звуковые и обонятельные чипы для роботов.

Задача тестирования нейроморфных микрочипов, являющихся аналоговыми системами, требует, как минимум, переключательного уровня и развития работ в этом направлении.

Литература

1.D.Goldhaber-Gordon,M.S.Montemerlo,J.Christopher Love, G.J.Opiteck, J.J. Ellenbogen. Overview of Nanoelectronic Devices//Proceedings of the IEEE, vol.85,No. 4,April 1997,pp.521-540.

2.Chris Diorio, Sunit Mahajan, Paul Hasler, Bradley Minch, Carver Mead. A High-Resolution Nonvolatile Analog Memory Cell//Proceedings of the 1995 IEEE International Symposium on Circuits and Systems, vol. 3, pp. 2233–2236, 1995.

3.Eric A.Vittoz. Analog VLSI Signal Processing:Why, Where and How? Journal of VLSI Signal Processing, vol.8, and Analog Integrated Circuits and Signal Processing, pp.27-44, July 1994.

4.C.Diorio, P. Hasler, B. A. Minch, C.Mead.A Complementary Pair of Four-Terminal Silicon Synapses//Analog Integrated Circuits and Signal Processing, 13, 1997,pp. 153–166.

5.Воронов А.В. О материалах и технологиях элементной базы.Труды 8 Всероссийской конференции «Нейрокомпьютеры и их применение» НКП-2002, Москва, 21-22 марта 2002 г.,с.795-805.

6.И.В.Заенцев. Нейронные сети:основные модели.Воронеж,ВГУ,1999, с.76.

7.Handbook of Brain Theory and Neural Networks, M.Arbib Ed.,Bradford Books,MIT Press,1995.

8.V. Beiu, José M. Quintana, María J. Avedillo. VLSI Implementations of Threshold Logic—A Comprehensive Survey// IEEE Transactions on Neural Networks, vol. 14, No. 5, September 2003, p.1217-1243.

9.Савельев А.В. К вопросу субстанциональности нейрокомпьютеров. Труды 8 Всероссийской конференции «Нейрокомпьютеры и их применение» НКП-2002, Москва, 21-22 марта 2002 г.,с.1247-1249.

10. Андрюхин А.И. Алгоритмы параллельного логического моделирования и псевдослучайной генерации тестов для МОП-структур // Микроэлектроника. -1995, N 5. -с. 331-336.

11.Андрюхин А.И. Параллельное логическое моделирование МОП-структур на переключательном уровне. // Электронное моделирование. -1996, N 2, -с. 88-92.

12. Андрюхин А.И. Параллельное многозначное логическое моделирование исправных и неисправных псевдобулевых схем. // Электронное моделирование -1997, N 1. -с. 58-63.

13.A.I.Andruckin. Simulation and Testing of MOS-structures on switch level.//Сборник научных трудов конференции «Моделирование-2006»,Киев,16-18 мая 2006,с.99-102.

Дата поступления в редакцию 27.05.07

УДК 004.032

ПОСТРОЕНИЕ СТАТИСТИЧЕСКИХ МОДЕЛЕЙ ЭФФЕКТИВНОСТИ ПАРАЛЛЕЛЬНЫХ ПРОГРАММ

В.Н.Белецкий, С.А.Резникова, А.А.Чемерис
Институт проблем моделирования в энергетике им. Г.Е.Пухова
НАН Украины

В статье рассмотрен подход к оценке и прогнозированию выполнения параллельных программ на многопроцессорных вычислительных системах. Приведен пример оценки выполнения задачи матрично-векторного умножения в компьютерах с общей памятью.

В настоящее время сфера применения многопроцессорных вычислительных систем (МВС) непрерывно расширяется, охватывая все новые области в самых различных отраслях науки, бизнеса и производства. Стремительное развитие кластерных систем создает условия для использования многопроцессорной вычислительной техники в реальном секторе экономики.

Наряду с расширением области применения, по мере совершенствования МВС происходит усложнение и увеличение количества задач в областях, традиционно использующих высокопроизводительную вычислительную технику.

Благодаря широкому использованию параллельных вычислительных систем, как распределенных, так и многопроцессорных, а также систем распараллеливания можно получить значительное ускорение вычислений по сравнению с однопроцессорными компьютерами. Это приводит к сокращению времени вычислений, увеличению производительности и уменьшению затрат. Очевидно, что эти значения есть функцией от параметров вычислительной системы – числа процессоров, объема памяти, пропускной способности каналов связи и т.д. Есть также зависимость от параметров задачи, которая выполняется, а особенно важным фактором есть качество распараллеливания алгоритма задачи.

Разработка параллельных программ остается все еще сложным и трудоемким процессом. Необходимо ответить на вопрос, является ли применение МВС оправданным для конкретной задачи, при каких значениях параметров задачи и архитектуры вычислительной системы можно обеспечить максимальную эффективность использования компьютера. В идеальном случае, количество процессоров определяет, во сколько раз быстрее выполнится распараллеленная

задача. Но в действительности ускорение меньше из-за организации доступа к общей памяти, необходимости синхронизации потоков, задержек в пересылках данных, а также зависимостей между переменными.

Известно, что существенным является влияние на эффективность размера программы, архитектуры вычислительных средств, количества процессоров, способа распараллеливания, количества и размера потоков, синхронизации, операционной системы и многих других факторов.

Существующие методы исследования эффективности можно разделить на две группы:

1. Расчет на основании результатов трассировки выполнения параллельной программы. Возможен только в тех системах программирования, в которых предусмотрена возможность получения значений времени выполнения отдельных конструкций параллельной программы [1].

2. Использование моделей параллельных алгоритмов, исследующих граф алгоритма решения задач. Инструмент для создания таких моделей включается в программные пакеты разработки параллельных программ, выполняются на последовательных компьютерах и не учитывают все особенности конкретной МВС [2].

Очевидно, что предпочтительней было бы использование моделей, которые бы одновременно учитывали и характеристики алгоритма, и параметры программы, а также архитектуры, включая способ доступа к памяти и т.д. При использовании таких моделей можно было бы ответить на вопросы предварительной оценки эффективности выполнения алгоритма без запуска программы с конкретными данными и получить оценку, при каких параметрах алгоритма его выполнение будет наиболее эффективным.

Решением этой проблемы может быть использование статистических методов построения моделей, которые способны прогнозировать производительность вычислений или время выполнения программы, не запуская программу с конкретными данными. Такие модели, добавленные к программному обеспечению, помогут сделать предварительные оценки эффективности выполнения программы на конкретной ВС и подобрать параметры задачи для получения максимального быстродействия при решении на данной ВС.

Модель, которая описывает исследуемый процесс, можно в общем случае представить в виде

$$Y = F(x_1, x_2, ..., x_n), \quad (1)$$

где Y – зависимая переменная, а x_i – независимые переменные.

Решение о виде функции регрессии принимается на основе анализа статистических данных и чаще всего для построения моделей достаточно линейных функций вида

$$Y = b_0 + b_1 x_1 + b_2 x_2 + ... + b_k x_k + \varepsilon, \quad (2)$$

Правильность построенной модели в значительной мере зависит от соответствующего множества переменных, которые имеют соответствующее влияние на исследуемый параметр. Переменные, используемые в модели должны:

1) иметь возможность изменения в широком диапазоне;
2) иметь сильную корреляцию с исследуемой переменной;
3) иметь слабую корреляцию с другими независимыми переменными.

“Оптимальный” состав факторов, включаемых в статистическую модель, является одним из основных условий ее “хорошего” качества. Проблема выбора “оптимальных” факторов обычно решается на основе содержательного и количественного (статистического) анализа тенденций рассматриваемых процессов.

На этапе содержательного анализа решается вопрос о целесообразности включения в модель тех или иных факторов, исходя из “здравого” смысла.

На этапе статистического анализа решается вопрос о том, насколько те или другие независимые переменные влияют на значение независимой переменной, какие из них нужно оставить в модели, а какие исключить.

При построении модели мы будем использовать апостериорный подход к уточнению состава факторов статистической модели. Таким образом, на этом этапе построения модели определяется множество переменных, входящих в выражение (1). Далее производится оценка коэффициентов b_i (i=0,1,…,k) выражения (2). Для этого применяют известные статистические методы, например, метод наименьших квадратов

Для иллюстрации и доказательства применимости данного подхода для анализа эффективности алгоритмов распараллеливания приведем пример, в котором умножается матрица на вектор. В данном примере оцениваем время выполнения программы для разной размерности задачи при разном количестве потоков.

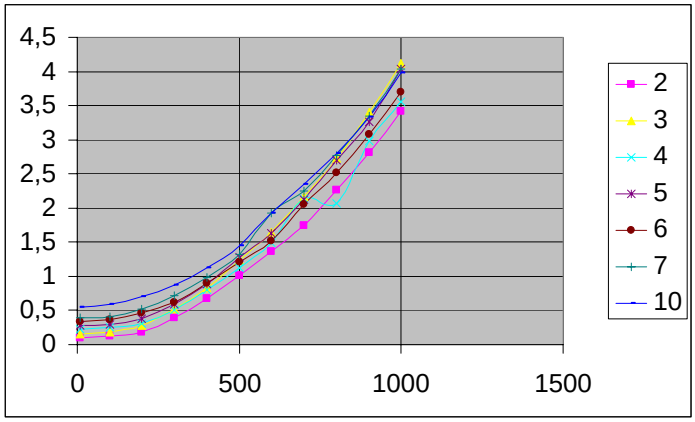


Рис. 1

На рис. 1 приведены результаты тестовых выполнений задачи при различных значениях размерности задачи *N* и числа потоков программы *P*. При этом статистические данные получены при выполнении программы для размерностей задачи в диапазоне 10 – 1000 уравнений с шагом 100. Эти данные отображены по оси *x* на рис. 1. Программа запускалась соответственно как 1, 2, 3, 4, 5, 6, 7 и 10 потоков. По оси *y* отображены значения времени выполнения программы для соответствующих *N* и *P*. Статистика выполнения программы получена на двухпроцессорном компьютере с архитектурой SMP, который построен на процессорах Pentium III/800 MHz.

Анализ полученных статистических данных, построение и верификация модели осуществлялась в программе STATISTICA, который содержит модули автоматического определения коэффициентов модели, а также генерирует оценки правдоподобия модели. Для нашей задачи была построена регрессионная модель *T*=f(*N*,*P*), где *T* – время решения задачи, в виде

$$T = b_0 + b_1 \cdot N + b_2 \cdot P \ . \qquad (3)$$

На основе полученных числовых значений модель принимает вид

$$T = -0{,}542 + 0{,}0036 \cdot N + 0{,}0546 \cdot P \qquad (4)$$

Квадрат коэффициента детерминации R^2=0,0299 свидетельствует о хорошем соответствии линии регрессии и экспериментальных данных.

Полученную модель можно использовать для прогнозирования времени вычисления данной задачи при различных значениях размерности.

Табл.1

Размерность задачи	Количество потоков	Время, рассчитанное на основании модели, сек	Время, полученное на основании эксперимента, сек
1100	4	3,638	3,078
1200	5	4,052	4,015
1400	6	4,827	4,922

Как видно из табл. 1 построенная модель может быть использована для оценки времени выполнения программы при выбранных значениях независимых переменных.

Выводы. Анализ эффективности параллельных вычислений для оценивания получаемого ускорения вычислений и степени использования всех возможностей компьютерного оборудования при параллельных способах решения задач - один из важных аспектов использования МВС.

Использование статистических методов для анализа времени выполнения алгоритмов на параллельных компьютерах является актуальной задачей. При этом могут быть получены ответы как на вопросы эффективности выполнения алгоритмов на конкретной вычислительной системе, так и на вопросы соответствия выполняемого алгоритма и данной параллельной ВС. Основным достоинством применения такого подхода есть возможность прогнозирования эффективности выполнения программ без необходимости ее выполнения на МВС.

Литература

1. Коновалов Н.А., Крюков В.А., Погребцов А.А., .Сазанов Ю.Л. C-DVM – язык разработки мобильных параллельных программ, Москва: изд. ИПМ им. М.В.Келдыша РАН, 1997.
2. Wolfe M., High Performance Compilers for Parallel Computing, Addison-Wesley Publishing Company, 1995.

Дата поступления в редакцию 14.05.07

УДК 681.518

МОДЕЛИРОВАНИЕ ЦИФРОВЫХ СХЕМ С НЕИСПРАВНОСТЯМИ НА РАСПРЕДЕЛЁННЫХ СИСТЕМАХ КЛАССА MIMD

Иванов Д.Е., к.т.н., с.н.с., отдел ТУС, Институт Прикладной Математики и Механики, Скобцов Ю.А., д.т.н., проф., кафедра АСУ, Донецкий национальный технический университет.

Предлагается алгоритм распределённого моделирования цифровых схем с неисправностями на системах класса MIMD. Данный алгоритм основывается на модели, использующей один компьютер в качестве ведущего. Рассматривается эффективность и ускорение параллельной реализации. Приведены численные результаты машинных экспериментов.

Введение. Параллельные вычисления на кластерных структурах являются одной из парадигм преодоления нехватки процессорных ресурсов при решении ресурсоёмких задач. Одной из таких задач в технической диагностике является моделирование работы неисправных цифровых устройств [1-3], которое применяется для определения диагностических свойств тестовой последовательности (например, полноты теста).

Принципиально распределенное моделирование с неисправностями можно осуществить двумя путями:

- разбиением теста на части и моделированием каждой такой части на отдельном процессоре системы; данный подход применялся в [4-5];
- разбиением схемы на подсхемы и моделированием подсхемы на отдельном процессоре. При этом необходимо построение протокола взаимодействия всех компонент вычислительного кластера. Такой подход применён в [6-7].

Одним из недостатков второго подхода авторы считают возрастающие накладные расходы на передачу информации необходимой для моделирования (описание подсхем и входные воздействия), построение протокола взаимодействия и балансировки нагрузки, а также синхронизации вычислительных потоков. Поэтому авторы для реализации выбрали первый подход. Заметим также, что в этом случае для схем большой размерности передача информации между процессорами (описание схемы и тестовых наборов) будет сведена к минимуму и выполняться только один раз, и следует ожидать высокой эффективности распределённой реализации алгоритма.

Алгоритм распределённого моделирования цифровых схем с неисправностями. В качестве последовательного алгоритма, предназначенного для распределённой реализации, авторы выбрали разработанный ими ранее алгоритм одиночного параллельного распространения неисправностей. Данный алгоритм показывает хорошие временные характеристики для контрольных схем [8] и подробно описан в [3].

Как отмечалось выше, в качестве базовой выбрана схема с одним ведущим компьютером. Весь вычислительный процесс разбит на несколько потоков: один поток-сервер и несколько (доступных в данный момент) процессов-клиентов. Опишем кратко работу сервера и клиентов. Один из процессоров в кластере выполняет серверную роль: поиск клиентов процесса, ввод описания схемы и тестовой последовательности, разбиение теста на подтесты, передача данных описаний клиентам и получение результатов. Алгоритм работы серверного процесса приведён на рис.1. Способ взаимодействия сервера с клиентами приведён на рис.2. Отметим, что при такой реализации сведены к минимуму и выполняются только один раз следующие элементы взаимодействия:

```
распределённое_моделирование(схема, тест)
{
  число_клиентов = поиск_клиентов();
  если( число_клиентов != 0 )
  {
    ввод_схемы();
    ввод_теста();
    построение_полного_списка_неисправностей();
    рабиение_списка_неисправностей();
    для( i=0 ; i< число_клиентов ; i++ )
    {
      передать_клиенту_i_описание_схемы();
      передать_клиенту_i_часть_списка_неисправностей();
      передать_клиенту_i_тест();
    }
    для( i=0 ; i< число_клиентов ; i++ )
    {
      получить_список_непроверенных_неисправностей();
    }
    формирование_отчёта();
  }
}
```

Рис.1 Алгоритм работы процесса-сервера.

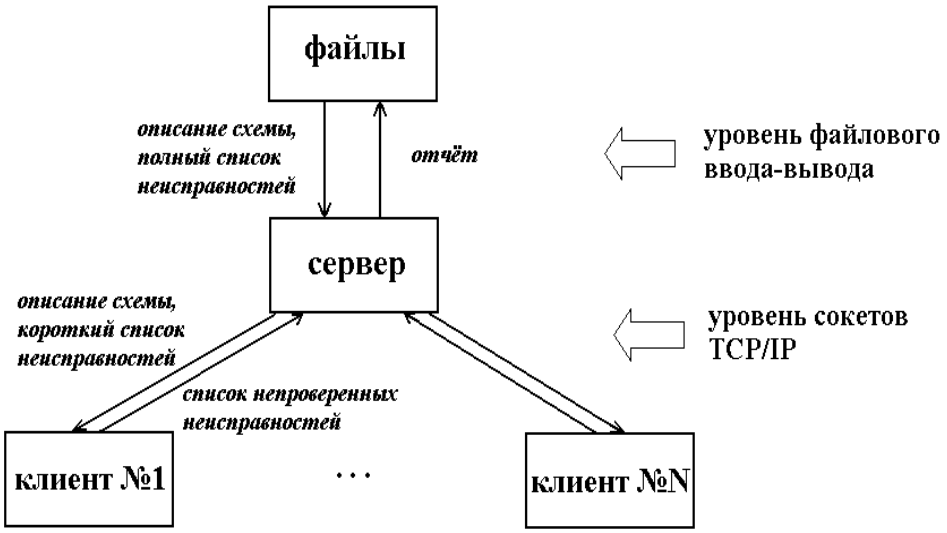


Рис.2 Диаграмма потоков данных при распределённом моделировании.

- передача описания схемы на процессоры-клиенты;
- передача тестовых воздействий на процессоры клиенты;
- синхронизация окончания вычислений.

Каждый процессор-клиент выполняет следующие функции: принимает полное описание схемы, принимает подтест, выполняет моделирование и отсылает его результаты на сервер. Ключевой алгоритм моделирования подробно описан в [1,3].

Программная реализация и анализ. Предложенный распределённый алгоритм реализован программно в среде программирования C++ Builder 6.0. В качестве среды взаимодействия выбраны блокирующие сокеты: процесс, ожидающий приёма данных, будет остановлен до тех пор, пока соответствующий процесс не передаст их.

Машинные эксперименты проводились в компьютерном классе, оборудованном локальной сетью со скоростью 100Мбит/с под управлением операционной системы Windows XP. Проводился замер следующих временных характеристик: общее время процесса моделирования, число событий при моделировании исправной схемы и схем с неисправностями, общее число событий. Для сравнительной базы бралась работа алгоритма авторов [3] на персональном компьютере соответствующей конфигурации. Для анализа была выбрана схема средней размерности S9234.ben [8]. Результаты экспериментов для различного числа процессоров-клиентов приведены в табл.1. В качестве анализа отметим следующие особенности:

Таблица 1. Экспериментальные данные для схемы S9234.ben

Характеристика	Однопроцес-сорная реализация	Многопроцессорная реализация (число процессоров)					
		1	2	3	4	6	8
длина теста	1000	1000					
общее время моделирования, сек.	330	336	194	138	107	86	79
общее число событий, млн.	440,05	440,05 (1,00)	441,81 (1,00)	443,21 (1,00)	443,78 (1.01)	447,79 (1,01)	449,93 (1,02)
число событий моделирования исправной схемы, млн.	0,48	0,48 (1,00)	0,95 (1,98)	1,42 (2,96)	1,90 (3,96)	2,85 (5,93)	3,80 (7,92)
число событий моделирования схем с неисправностям и, млн.	439,58	439,58 (1,00)	440,86 (1,00)	441,79 (1,00)	441,88 (1,00)	444,94 (1,01)	446,13 (1,01)
ускорение	-	**0,98**	**1,7**	**2,39**	**3,08**	**3,83**	**4,17**
эффективность	-	**0,98**	**0,85**	**0,8**	**0,77**	**0,64**	**0,52**

- общее число событий практически не изменяется в зависимости от числа клиентов (коэффициент от 1.00 для числа клиентов 1-4 до 1.02 для числа клиентов 8);
- число событий моделирования исправной схемы растёт практически линейно (коэффициент от 1.00 для числа клиентов 1 до 7.92 для числа клиентов 8), что связано с необходимостью восстанавливать поведение исправной схемы на каждом узле кластера;
- суммарное число событий моделирования неисправной схемы также практически не растёт.

Для параллельной реализации алгоритма принято вычислять параметры ускорение и эффективность реализации [9].

Ускорение, рассчитываемое для параллельной реализации алгоритма для p процессоров определяется формулой:

$$S_p(n) = \frac{T_1(n)}{T_p(n)},$$

где p – число процессоров в параллельной реализации алгоритма, n – параметр вычислительной сложности алгоритма, $T_i(n)$ - время выполнения параллельного алгоритма на системе с i процессорами.

Эффективность использования процессоров при параллельной реализации алгоритма рассчитывается по формуле:

$$E_p(n) = \frac{T_1(n)}{p \cdot T_p(n)} = \frac{S_p(n)}{p}.$$

Имея экспериментальные данные, из параллельного закона Амдаля можно получить f - долю последовательностных вычислений алгоритма: $f = 0.10$ (для p=4) и $f = 0.13$ (для p=8). Отсюда, применяя данный закон при $p \to \infty$, получим максимально возможное ускорение алгоритма, которое изменяется от 10раз (при использовании для расчётов p=4) до 7.7раз (используя p=8).

Заключение. В работе предложен эффективный алгоритм распределённого оделирования цифровых схем с неисправностями, использующий схему со статическим разбиением теста. Проведены машинные эксперименты, на основании которых оценена эффективность распараллеливания исходной задачи моделирования.

Литература

1. Niermann T.M., Cheng W.-T., Patel J.H. PROOFS: A Fast, Memory-Efficient Sequential Circuits Fault Simulator // IEEE Trans. CAD. – 1992.– P.198-207.
2. Kung C.P., Lin C.S. HyHope: A Fast Fault Simulator with Efficient Simulation of Hypertrophic Faults // Proc. of International Test Conference. - 1994. - P.714-718.
3. Иванов Д.Е., Скобцов Ю.А. Параллельное моделирование неисправностей для последовательностных схем // Искусственный интеллект. – 1999. - №1. – С.44-50.
4. P.A. Duba, R.K. Roy, J.A. Abraham and W.A. Rogers, “Fault simulation in a distributed environment”, in Proceedings of the 25th ACM/IEEE Design Automation Conference, pp.686-691, June 1988.
5. T. Marcas, M. Royals and N. Kanopoulos, “On distributed fault simulation”, IEEE Computer, vol. 7, pp. 40-52, Jan. 1990.
6. S. Ghost, “NODIFS: a novel, distributed circuit partitioning based algorithm for fault simulation of combinational an sequential digital design on loosely coupled parallel processors”, tech. rep., LEMS, Division of Engineering, Brown University, Providence, RI, 1991.
7. Ладыженский Ю.В., Попов Ю.В. Программная система для исследования протоколов синхронизации при распределённом событийном логическом моделировании // Наукові праці Донецького національного технічного університету, Серія “Обчислювальна техніка та автоматизація”.- 2004.- Выпуск №74.- С.201-209.
8. Brgles F., Bryan D., Kozminski K. Combinational profiles of sequential benchmark circuits // International symposium of circuits and systems, ISCAS-89. – 1989. – P.1929-1934.
9. Гергель, В.П., Стронгин, Р.Г. Основы параллельных вычислений для многопроцессорных вычислительных систем. - Н.Новгород, ННГУ, 2003, (2 изд.).

Работа поступила в редакцию 14.05.07.

УДК 681.3

МОДЕЛИРОВАНИЕ РАБОТЫ ПРИЛОЖЕНИЙ НА СЕТЕВЫХ ПРОЦЕССОРАХ

Грищенко В.И., Ладыженский Ю.В.
Донецкий Национальный Технический Университет

Увеличение сетевого трафика и необходимость решения задач обслуживания сети со скоростью канала данных повлекло за собой создание сетевых процессоров (СП). В предложенной статье описывается комплексная методика моделирования сетевых процессоров (КМСП) приводятся результаты моделирования, а также указываются недостатки методики и пути ее развития.

Методика моделирования. В работах [1, 2] представлена комплексная методика исследования производительности сетевых процессоров. Она состоит из четырех основных этапов (см. рис. 1): выбор приложения и архитектуры, имитационное моделирование работы приложения, аналитический расчет эффективности выполнения приложения на выбранной архитектуре и визуализация результатов.

Имитационное моделирование включает расчет площади и энергопотребления кэша с использованием инструмента CACTI [3] и моделирование работы приложения в системе SimpleScalar [4]. CACTI представляет собой консольную утилиту, позволяющую на основании параметров кэша (размер, ассоциативность, технология производства и др.) определять занимаемую площадь, объем рассеиваемой энергии, скорость доступа и другие характеристики. Характеристики кэша не зависят от выполняемого приложения, поэтому могут быть рассчитаны один раз, и многократно использоваться в последствии.

Система SimpleScalar используется для детального моделирования работы RISC-процессора. Использование SimpleScalar позволяет получить характеристики выполнения приложения: интенсивность промахов кэша данных и инструкций, число операций чтения/записи и общее количество инструкций. Для анализа в SimpleScalar приложение должно быть предварительно скомпилировано с использованием специализированного кросс-компилятора, поставляемого вместе с системой.

Рис. 1 – Структура методики моделирования

Таблица 1 – Интенсивность промахов кэша данных для приложения AntNet (объем кэша — 1 КБ)

Функция	Файл	Количество промахов	Процент от общего числа
antnet_route_packet()	antnet.h	1273879	81.88
antnet_init_router()	antnet.h	107213	6.85
fread()	fread.c	46155	2.98
malloc()	malloc.c	33130	2.11
_free_internal()	free.c	19383	1.22
antnet_read_packet()	antnet.h	12756	0.83

Встроенные механизмы профилирования позволяют находить «узкие места» исследуемых приложений. В таблице 1 приведены результаты профилирования алгоритма маршрутизации AntNet, который подробно описан в работах [6, 7]. Можно видеть, что более 80% промахов кэша приходится на поиск в таблице маршрутизации и это является самым узким местом алгоритма.

Аналитическая модель. Выбор структуры сетевого процессора непосредственно влияет на используемую аналитическую модель. В проведенных исследованиях использовалась кластерная структура процессора, представленная на рисунке 2. В ней СП состоит из однотипных вычислительных ядер, которые объединяются в кластеры

(группы) и внутри кластера разделяют общий интерфейс к внешней памяти. Каждое ядро имеет раздельные кэши команд и данных, и аппаратно поддерживает работу нескольких потоков. Переключение между потоками осуществляется с нулевой задержкой и происходит в те моменты, когда программа ожидает завершения команды обращения к ОЗУ.

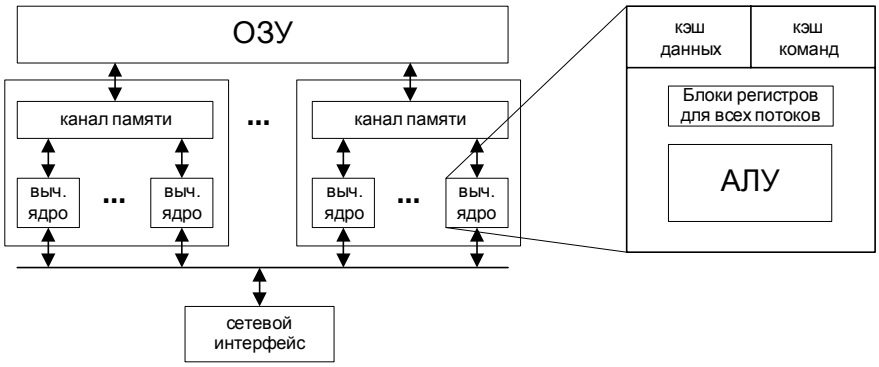


Рис. 2 – Обобщенная структура СП

При такой архитектуре вычислительная мощность всего СП определяется как сумма мощностей всех его ядер:

$$IPS = \sum_{j=1}^{m} \sum_{k=1}^{n} \rho_{p_{j,k}} \cdot f_{p_{j,k}} \qquad (1)$$

где $\rho_{p_{j,k}}$ - загруженность ядра (j, k), команд/такт, $f_{p_{j,k}}$ - тактовая частота процессора, такт/секунда, m – число кластеров в СП, n – количество процессорных ядер в каждом кластере.

Рассматривать ядра с различной производительностью имеет смысл только в тех случаях, если они выполняют различные приложения, но для данной структуры все ядра будут параллельно обрабатывать поток данных, и, как следствие, будут испытывать одинаковую нагрузку.

Загруженность отдельного многопоточного процессорного ядра с кэшами данных и команд определяется по формуле [9]:

$$\rho_p(u) = 1 - \frac{1}{\sum_{i=0}^{u} \left(\frac{1}{p_{miss} \tau_{mem}} \right)^i \frac{u!}{(u-i)!}} \qquad (2)$$

где ρ_p - загруженность вычислительного ядра, u - число аппаратно поддерживаемых потоков, p_{miss} - совокупная интенсивность промахов кэша, τ_{mem} - время доступа к памяти.

Время доступа к памяти учитывает время ожидания запроса в очереди (τ_Q), время физического отклика памяти (τ_{DRAM}) и время передачи данных через кэш ($\tau_{transmit}$):

$$\tau_{mem} = \tau_Q + \tau_{DRAM} + \tau_{transmit} \quad (3)$$

Интенсивность промахов кэшей зависит от приложения, выполняемого на СП:

$$p_{miss} = mi_c + (f_{load} + f_{store}) \cdot md_c \quad (4)$$

где mi_c и md_c - интенсивности промахов кэшей команд и данных соответственно, f_{load} и f_{store} - частота появления в коде программы команд чтения и записи в память.

Результаты моделирования. С помощью представленной методики были исследованы тестовый набор сетевых приложений CommBench [5] и алгоритм маршрутизации AntNet [6, 7]. Набор CommBench охватывает практически все задачи, которые выполняет современный маршрутизатор: фрагментация, маршрутизация, планирование очередей, обработка медиа-потока, шифрование данный и др. Результаты моделирования набора CommBench показаны в таблице 2, детальное же их рассмотрение можно найти в работе [8]. Представленные данные соответствуют однопроцессорной системе, многоядерные процессоры будут рассматриваться в дальнейших исследованиях. В качестве критерия оптимизации выбрано отношение вычислительной мощности к площади процессора (IPS/A).

Алгоритм AntNet использует принципы поведения колонии муравьев при нахождении кратчайших путей к источнику пищи. Каждый муравей оставляет за собой след из активных веществ — феромонов. И когда следующий муравей окажется рядом, то он с большей вероятностью выберет то направление, по которому до него прошел другой муравей, тем самым он увеличивает концентрацию феромонов на этом маршруте. Чем больше муравьев изберет маршрут, тем больше на нем будет феромона и тем больше вероятность того, что остальные последует за ними. Таким образом, через некоторое время вся колония будет двигаться по одному пути.

В алгоритме AntNet в качестве муравьев используются активные агенты — специализированные пакеты, несущие информацию о состоянии сети на своем пути. Когда такой муравей двигается к адресу назначения — он собирает статистику о задержках на каждом маршрутизаторе, а по возвращении он обновляет таблицы маршрутизации в соответствии с собранной статистикой.

Таблица 2 — Результаты исследования пакета CommBench

Приложение	Описание	Оптимальный кэш данных, КБ	Оптимальный кэш команд, КБ
FRAG	Фрагментация	2	2
DRR	Планирование очередей	2	2
RTR	Маршрутизация	4	32
JPEG (сжатие)	Упаковка изображения в формат JPEG	2	1
JPEG (распаковка)	Восстановление изображения из формата JPEG	16	32
REED (кодирование)	Избыточное кодирование	4	1
REED (декодирование)		4	1
CAST (кодирование)	Шифрование	16	32
CAST (декодирование)	Дешифрование	16	64
ZIP (сжатие)		4	2
ZIP (распаковка)		4	2

Для алгоритма AntNet оптимальным оказывается процессор с кэшем данных объемом 2 КБ и кэше команд объемом 4 КБ. Однако показано, что при увеличении объема кэша данных до 128 КБ производительность возрастает вдвое, а двукратный прирост производительности может иметь существенное значение.

Совершенствование модели. Кластерная структура сетевого процессора позволяет исследовать многоядерные устройства с параллельной обработкой данных, однако существуют коммерческие процессоры, организованные в виде конвейеров (например, CISCO PFX [10] или семейство сетевых процессоров от Intel [11] с настраиваемой системой внутренних связей). В конвейерных структурах актуальным становится вопрос передачи информации между вычислительными ядрами. В настоящий момент распространены три подхода: передача данных через внешнюю память (рис. 3.а), использование специализированной внутренней памяти, интегрированной в кристалл процессора (рис. 3.б) и

использование разделяемых регистров (рис. 3.в). Второй и третий способ похожи между собой, их отличие заключается только в максимальных объемах передаваемой информации и скорости доступа к данными. В случае использования внешнего ОЗУ объемы памяти обмена многократно возрастают, но также увеличивается и время отклика системы.

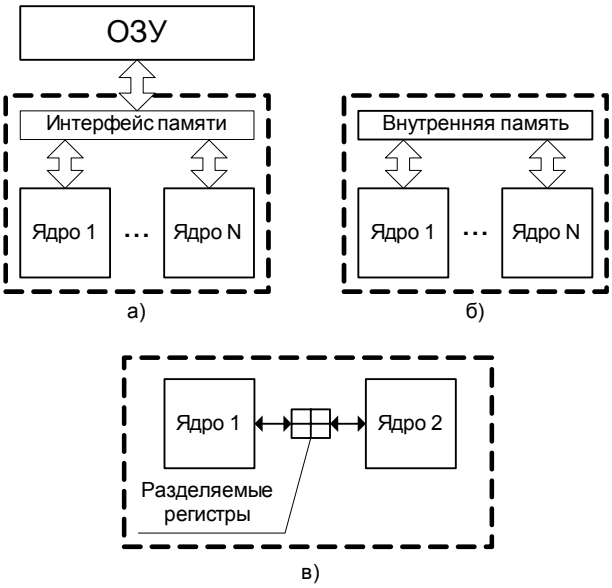


Рис. 3 – Способы организации взаимодействия между процессорными ядрами

В существующих СП часто наиболее трудоемкие функции реализуется аппаратно в виде сопроцессоров. Они позволяют решать специфические задачи (криптография, сбор статистики) практически на частоте канала данных. Учет работы сопроцессоров позволит повысить точность моделирования и получать более достоверные результаты.

Выводы. В статье представлена методика моделирования сетевых процессоров, показаны ее структура и назначение отдельных этапов моделирования. Продемонстрированы результаты применения методики для исследования тестового набора приложений. Предложены пути совершенствования модели для исследования перспективных архитектур сетевых процессоров.

Литература

1. Tilman Wolf, Mark A. Franklin, "Performance Models for Network Processor Design," IEEE Transactions on Parallel and Distributed Systems, vol. 17, no. 6, pp. 548-561, Jun., 2006.

2. Ладыженский Ю.В., Грищенко В.И. Моделирование сетевых процессоров пакетной обработки данных. ИНТЕРНЕТ-ОБРАЗОВАНИЕ-НАУКА, пятая международная конференция ИОН-2006. 10-14 октября, 2006. Сборник материалов конференции. Том 2. – Винница: УНИВЕРСУМ-Винница, 2006, стр. 417-422.

3. Premkishore Shivakumar and Norman P. Jouppi, “CACTI 3.0: An integrated cache timing, power and area model,” Tech. Rep. WRL Research Report 2001/2, Western Research Laboratory, Palo Alto, CA, Aug. 2001.

4. Doug Burger and Todd M. Austin, “The SimpleScalar tool set, version 2.0,” Tech. Rep. 1342, Department of Computer Science, University of Wisconsin in Madison, June 1997.

5. Tilman Wolf and Mark A. Franklin, “CommBench - a telecommunications benchmark for network processors,” in Proc. of IEEEInternational Symposium on Performance Analysis of Systems and Software (ISPASS), Austin, TX, Apr. 2000, pp. 154–162.

6. Di Caro G. Dorigo. M. AntNet: Distributed Stigmergetic Control for Communications Networks: Journal of Artifical Intelligence Research. – 1998. - №9 – pp. 317-365.

7. Ладыженский Ю.В., Мирецкая В.А. Исследование муравьиных алгоритмов маршрутизации в компьютерных сетях. ИНТЕРНЕТ-ОБРАЗОВАНИЕ-НАУКА, пятая международная конференция ИОН-2006. 10-14 октября, 2006. Сборник материалов конференции. Том 2. – Винница: УНИВЕРСУМ-Винница, 2006, стр. 375-377.

8. Грищенко В.И., Ладыженский Ю.В. Исследование архитектуры кэша сетевых процессоров // Радіоелектронні і комп’ютерні системи. – 2007. – №6 (25). С.186-191.

9. Anant Agarwal, “Performance tradeoffs in multithreaded processors,” IEEE Transactions on Parallel and Distributed Systems, vol. 3, no. 5, pp. 525–539, Sept. 1992.

10. N. Shah, "Understanding network processors," Tech. Rep. Version 1.0, EECS, University of California, Berkeley, September 2001.

11. Intel IXP1200 Network Processor Family. Microcode Programmer’s Reference Manual. Intel Corporation. March 2002.

Работа поступила в редакцию 13.06.2007.

УДК 681.3.06

МЕТОД ПОЛУЧЕНИЯ ОПТИМАЛЬНОГО БАЛАНСА НАГРУЗКИ ДЛЯ РАСПРЕДЕЛЕННОГО ЛОГИЧЕСКОГО МОДЕЛИРОВАНИЯ

В.А. Куркчи, Ю.В. Ладыженский
Кафедра ПМИ, ДонНТУ

Наведено опис алгоритму динамічного балансування навантаження, що базується на розв'язанні транспортній задачі. Запропоновано модифікацію алгоритму, що дозволяє гарантовано отримувати оптимальній баланс навантаження за рахунок незначного підвищення рівню міграції. Наведено результати експериментального порівняння алгоритмів.

Введение. В течение процесса логического моделирования при использовании вычислительной сети нагрузки на рабочие станции с течением времени могут значительно изменяться. Возникающая разбалансированность нагрузки приводит к неэффективному использованию оборудования. Поэтому важно периодически балансировать нагрузку [1].

С точки зрения распределенного логического моделирования среди ряда существующих методов балансировки нагрузки [2-6] особого внимания заслуживает метод, минимизирующий уровень миграции [7].

Чем меньше задач будет передано в процессе балансировки нагрузки, тем скорее можно будет продолжить процесс моделирования. Поэтому минимизация уровня миграции очень важна при балансировке нагрузки.

В нашей работе [8] был предложен алгоритм, позволяющий балансировать нагрузку с минимальным уровнем миграции. Алгоритм основан на решении транспортной задачи. Этот алгоритм позволяет в среднем на 23% сократить количество передаваемых подзадач [9], однако качество его решений может быть улучшено.

Целью данной работы является модификация указанного алгоритма для получения оптимальных решений.

Постановка задачи. Пусть P есть количество процессоров, а S множество всех подзадач, которые должны быть решены. Множество $C=\{C_1,C_2,\ldots,C_P\}$ является размещением множества всех подзадач на процессоры. Каждая подмножество подзадач C_i обрабатывается на одном процессоре i. Подмножества C_i и C_j не пересекаются. Объединение всех

множеств C_i есть множество S. Обозначим через l_i=| C_i | нагрузку на один процессор и

$$l_{avg} = \frac{1}{P} \sum_{C_i \in C} l_i$$

среднюю нагрузку на процессор.

Рассмотрим граф $G(C,E)$, где $(C_i,C_j) \in E$, если возможно переместить подзадачи от процессора i к процесору j. Если есть такое множество подзадач C_i в C, что $l_i > l_{avg}$, одна или несколько подзадач $k \in C_i$ должны быть перемещены в другую часть графа подзадач. Часть графа подзадач, принимающая вершины может быть выбрана только среди множества смежных вершин графа $\{C_j | (C_i,C_j) \in E\}$.

Задача динамической балансировки нагрузки заключается в поиске нового размещения подзадач $C' = \{C'_1, C'_2, \ldots, C'_P\}$, которое минимизирует одновременно дисбаланс и уровень миграции. Дисбаланс можно найти по одной из следующих формул [5], [6], [3]:

$$M_1(C) = \max_{k=\overline{1,P}} |l_k| - \min_{m=\overline{1,P}} |l_m|,$$

$$M_2(C) = \frac{1}{l_{\max}} \sqrt{\frac{\sum_{i=1}^{P} \left(li - l_{avg}\right)^2}{P-1}},$$

$$M_3(C) = \max_{i=\overline{1,P}} l_i,$$

где

$$l_i = | C_i |, \; l_{\max} = \max_{i=\overline{1,P}} l_i, \; l_{avg} = \frac{1}{P} \sum_{i=1}^{P} l_i .$$

Уровень миграции вычисляется следующим образом:

$$Q(C) = \frac{\sum_{k=1}^{P} \left(|C_k| - |C_k \setminus C'_k|\right) + |C'_k \setminus C_k|}{2}.$$

Алгоритм, основанный на решении транспортной задачи. Задачу балансировки нагрузки можно рассматривать как задачу определения числа подзадач, которые необходимо переместить с каждого процессора и указания целевых процессоров для каждой перемещаемой подзадачи.

Есть несколько процессоров с нагрузкой $l_i > l_{avg}$. Каждый из этих процессоров должен передать $\lfloor l_i - l_{avg} \rfloor$ подзадач. Также есть некоторое число процессоров с нагрузкой $l_i < l_{avg}$,, каждый из которых должен получить $\lfloor l_{avg} - l_i \rfloor$ подзадач. Если определить расстояние между двумя процессорами, описанная задача может быть решена как транспортная с помощью метода потенциалов. Метод потенциалов описан в [10].

Псевдокод алгоритма балансировки нагрузки, основанного на решении транспортной задачи, представлен на рис. 1. Этот алгоритм возвращает матрицу Δ размера $P \times P$, $\Delta[i,j]$ является количеством подзадач передаваемых от процессора i процессору j. Начальные значения $\Delta[i,j]$ должны быть равны нулю. На первом шаге алгоритм вычисляет среднюю нагрузку на процессор. Затем формируются множества O и I процессоров, которые должны соответственно передавать и получать подзадачи. На шаге 4 вызывается алгоритм Флойда, результатами работы которого являются матрица D длин путей матрица W, характеризующая последовательность вершин для каждого пути.

На шагах 6-8 формируются исходные данные для метода потенциалов: длины кратчайших путей между процессорами, передающими подзадачи, и принимающими их, заносятся в матрицу T. Затем на шаге 9 вызывается метод потенциалов, чтобы решить транспортную задачу. Он использует матрицу T длин путей между процессорами, , множество $O - \lfloor l_{avg} \rfloor$ количеств подзадач, которые требуется передать каждому процессору с избыточной нагрузкой, множество $\lfloor l_{avg} \rfloor - I$ количеств подзадач, которые может получить каждый процессор с недостатком нагрузки. Результатом метода потенциалов является матрица TS размера $|O| \times |I|$, каждый элемент $TS[k,m]$ есть количество подзадач, которое следует передать от процессора $O[k]$ процессору $I[m]$.

После этого, результаты метода потенциалов должны быть интерпретированы. Для каждой пары процессоров «передающий - принимающий» (шаги 10-11) необходимо восстановить кратчайший путь (шаги 12-16). Переменная c для восстановления пути инициализируется на шаге 12 и ссылается на текущий процессор пути. Количество подзадач, передаваемых от процессора k процессору m, добавляется к каждому ребру пути в графе процессоров, как это показано на рис. 1. В реализации

алгоритма это количество равно $TS[k,m]$, и оно прибавляется к элементу $\Delta[c,W[c,m]]$ возвращаемой матрицы, где $W[c,m]$ – следующий процессор пути [см. 11]. Эти операции выполняются на шагах 13-16, пока переменная c, представляющая текущий процессор, не станет равной процессору m.

1. $l_{avg} = \left\lceil \frac{1}{P}\sum_{i=1}^{P} C_i \right\rceil$
2. $O \leftarrow \{C_i \mid |C_i| > l_{avg}\}$
3. $I \leftarrow \{C_i \mid |C_i| < l_{avg}\}$
4. $\langle D,W \rangle \leftarrow Floyd(G(C,E_N))$
5. $T \leftarrow CreateMatrix(|O|\times|I|)$
6. *foreach* $k \in O$
7. *foreach* $m \in I$
8. $T[k,m] \leftarrow D[k,m]$

   endfor

   endfor
9. $TS \leftarrow SolveTransport(T, O - \lfloor l_{avg} \rfloor, \lfloor l_{avg} \rfloor - I)$
10. *foreach* $k \in O$
11. *foreach* $m \in I$
12. $c := vs$
13. *repeat*
14. $\Delta[c,W[c,m]] \leftarrow \Delta[c,W[c,m]] + TS[k,m]$
15. $c \leftarrow W[c,m];$
16. *until* $c=m;$

    endfor

    endfor

Рис. 1. Псевдокод алгоритма балансировки нагрузки, основанного на решении транспортной задачи.

Модификация алгоритма. Описанный алгоритм позволяет всегда получать наименьшую максимальную нагрузку на процессор $M_1(C)$. Это достигается за счет того, что вся нагрузка выше средней передается другим процессорам. При этом также минимизируется количество

передаваемых подзадач – оптимальность решения достигается за счет использования метода потенциалов.

Однако минимальная нагрузка на процессор может остаться неизменной. По этой причине показатели сбалансированности нагрузки $M_2(C)$ и $M_3(C)$ оказываются большими.

Рассмотрим причины появления таких решений на примере. Пусть нагрузка на 10 процессорах распределена следующим образом: 100, 104, 96, 96, 99, 100, 103, 101, 102, 96. Средняя нагрузка на процессор l_{avg}=99.7, поэтому нагрузка свыше $\lceil l_{avg} \rceil$=100 подзадач будет передана другим процессорам.

После применения описанного алгоритма нагрузка распределена следующим образом: 100, 100, 100, 100, 100, 100, 100, 100, 100, 97. Очевидно, что сделать нагрузку меньше 100 подзадач невозможно, но возможно распределить нагрузку следующим образом: семь процессоров по 100 подзадач и три по 99.

Так как 997 подзадач не могут равномерно распределиться на 10 процессорах – необходимо еще 3 задачи – оказывается, что избыточная нагрузка на все процессоры составляет 10 подзадач, а недостаток нагрузки – 13 подзадач. В таких случаях для решения задачи методом потенциалов вводятся фиктивный процессор с фиктивной лишней нагрузкой, которая удаляется при составлении плана миграции. В данном случае вводятся 3 фиктивные подзадачи.

Метод потенциалов работает таким образом, что фиктивные подзадачи попадают на один, реже на два, процессора. Исправить этот недостаток внесением изменений в алгоритм оказалось невозможным.

Однако можно использовать алгоритм повторно, чтобы получить сбалансированную нагрузку. Для этого применим повторно алгоритм, но примем за среднюю нагрузку $\lfloor l_{avg} \rfloor$=99. Это создаст избыток лишней нагрузки.

В примере после повторного применения распределение нагрузки будет следующим: 99, 99, 100, 100, 100, 100, 100, 100, 100, 99.

Таким образом, способ применения алгоритма балансировки нагрузки, основанного на транспортной задаче, позволяющий получить оптимальный баланс состоит из двух этапов:

1. Применить для решения задачи алгоритм, основанный на транспортной задаче (см. рис 1). На шаге 1 алгоритма принять

$$l_{avg} = \left\lceil \frac{1}{P} \sum_{i=1}^{P} C_i \right\rceil$$

2. Применить для решения задачи алгоритм, основанный на транспортной задаче (см. рис 1). На шаге 1 алгоритма принять

$$l_{avg} = \left\lfloor \frac{1}{P} \sum_{i=1}^{P} C_i \right\rfloor.$$

Экспериментальные исследования. Обе версии алгоритма были программно реализованы. Также программно реализован алгоритм из [7]. Эксперименты проводились на случайных графах задач с 10000 вершин. Задачи распределялись между 10 процессорами.

В таблице 1 представлены характеристики полученных решений $M_1(C)$, $M_2(C)$, $M_3(C)$, при этом Н – для алгоритма из [7], Т – для исходного алгоритма, ТВ – для модифицированного алгоритма.

Из таблицы видно, что характеристика сбалансированности $M_1(C)$ одинакова для разработанных алгоритмов, но лучше, чем у алгоритма из [9]. Характеристики $M_2(C)$ и $M_3(C)$ существенно лучше у модифицированного алгоритма. Кроме того, $M^{TB}{}_2(C)$=1 для всех входных графов. Это означает, что на всех тестовых задачах достигнуто максимально сбалансированное решение, так как в случаях, когда количество подзадач не делится нацело на количество процессоров, разница в нагрузке на процессоры не может быть устранена полностью.

В таблице 2 представлены уровни миграции для тех же тестовых задач. Уровень миграции разработанных алгоритмов ниже, чем у алгоритма из [7] на 24.7% для исходного алгоритма и на 22.5% для модифицированного алгоритма. Таким образом, для получения оптимального баланса нагрузки па процессоры модифицированный алгоритм перемещает на 2,2% больше подзадач, чем исходный алгоритм.

Выводы. Проведен анализ недостатков работы алгоритма динамической балансировки нагрузки, основанного на решении транспортной задачи.

Разработана модификация алгоритма, позволяющая получать полностью сбалансированные решения.

Экспериментальные результаты показали, что решения, получаемые при помощи модификации алгоритма, действительно полностью сбалансированы, при этом уровень миграции по сравнению с исходным алгоритмом практически не изменяется.

Таблица 1- Характеристики полученных решений

$M^{H}_{1}(C)$	$M^{T}_{1}(C)$	$M^{TB}_{1}(C)$		$M^{H}_{2}(C)$	$M^{T}_{2}(C)$	$M^{TB}_{2}(C)$		$M^{H}_{3}(C)$	$M^{T}_{3}(C)$	$M^{TB}_{3}(C)$
502	501	501		4	19	1		0,002312	0,008265	0,000435
515	514	514		4	16	1		0,002854	0,006784	0,000778
507	506	506		4	16	1		0,001933	0,006892	0,000791
509	508	508		3	5	1		0,001506	0,002145	0,000852
486	484	484		5	8	1		0,003468	0,003942	0,001012
508	507	507		4	14	1		0,001978	0,006018	0,000904
492	489	489		5	3	1		0,002593	0,001337	0,00073
509	507	507		5	16	1		0,003204	0,006878	0,000789
509	507	507		5	9	1		0,002811	0,003869	0,000981
505	503	503		4	1	1		0,002299	0,000433	0,000433
501	500	500		4	11	1		0,002685	0,004903	0,000917
500	500	500		2	10	1		0,001183	0,004359	0,001
492	491	491		3	17	1		0,001849	0,007546	0,000727
476	474	474		5	1	1		0,002856	0,00046	0,00046
505	504	504		3	13	1		0,001907	0,005622	0,000946
509	507	507		3	2	1		0,002143	0,00086	0,000592
510	510	510		3	18	1		0,001743	0,007692	0,000588
520	519	519		3	10	1		0,001654	0,004239	0,000944
497	494	494		6	15	1		0,003237	0,006618	0,000877
501	499	499		4	16	1		0,001851	0,006988	0,000802

Таблица 2 – уровень миграции для полученных решений

№	Q^{H}	Q^{T}	Q^{TB}		№	Q^{H}	Q^{T}	Q^{TB}
1	510	386	404		11	536	419	431
2	485	353	368		12	441	324	333
3	558	414	429		13	553	419	435
4	385	273	277		14	424	308	308
5	427	319	329		15	472	347	359
6	514	383	396		16	509	382	383
7	503	417	419		17	561	444	461
8	504	370	385		18	514	396	406
9	457	359	367		19	436	300	314
10	517	391	391		20	478	369	384

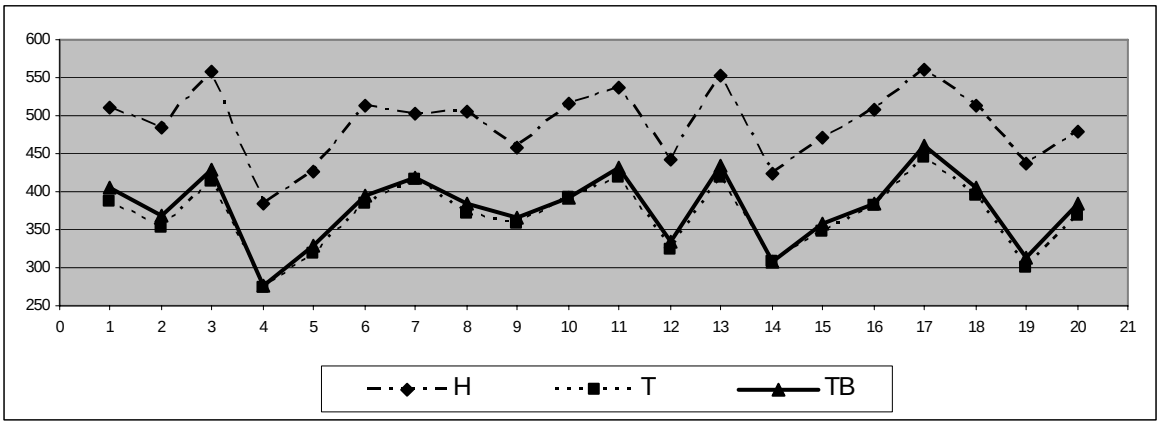


Рисунок 2 – Уровень миграции для каждой тестовой задачи

Литература

1. Ю. Ладыженский, В. Куркчи, Балансировка нагрузки в распределенном логическом моделировании//ІНТЕРНЕТ-ОСВІТА-НАУКА-2006, п'ята міжнародна конференція ІОН-2006, 10-14 жовтня, 2006, збірник матеріалів конференції. Том 2. – Вінниця: УНІВЕРСУМ-Вінниця, 2006, – 647-653.

2. K. Devine and al., A new challenges in dynamic load balancing.//Applied Numerical Mathematics, v 52, 2005, pp 133-152.

3. A. Pothen, Graph partitioning algorithms with applications to scientific computing.//Parallel Numerical Algorithms, Kluwer Academic Press, 1996.

4. G. Karypis, V. Kumar, Analysis of multilevel graph partitioning. Tech. Report 95-035, Computer Science Department, University of Minnesota, 1995.

5. H. Avril, C. Tropper, Clustered time warp and logic simulation//9th Workshop on Parallel and Distributed Simulation (PADS'95), 1995, pages 112-119.

6. H. Avril, C. Tropper, The Dynamic Load Balancing of Clustered Time Warp for Logic Simulations//Workshop on Parallel and Distributed Simulation, 1996, pages 20-27.

7. Y.F. Hu, R.J. Blake, An optimal dynamic load balancing algorithm. Preprint DL-P-95-011, Daresbury Laboratory, Warrington, WA44AD, UK. (To be published in Concurrency: Practice & Experience), 1995.

8. Y.V. Ladyzhensky, V.V. Kourktchi, Transport-Problem-Based algorithm for dynamic load balancing in logic simulation.//Information Systems Technology and its Applications: Proceedings of 6th international conference ISTA'2007, may 23-25, 2007, Kharkiv, Ukraine. – Lecture Notes in Informatics, Gesellschaft fur Informatik, Bonn 2007.

9. Ю.В. Ладыженский, В.А. Куркчи. Исследование свойств алгоритмов для динамической балансировки нагрузки в распределенном логическом моделировании.//Вісник Національного технічного університету „Харківський політехнічний інститут": Збірник наукових праць. Тематичний випуск „Системний аналіз, управління та інформаційні технології". – Харків: НТУ „ХПІ". –2007. –№6. – 24-34.

10. Венцель Е.С. "Исследование операций: задачи, принципы, методология" -М.: "Наука",1980. - 532с.

11. Дискретная математика для программистов / Ф.А. Новиков – СпБ: Питер, 2001. – 304с.: ил.

Работа поступила в редакцию 08.06.2007.

УДК 681.3

МУРАВЬИНАЯ МАРШРУТИЗАЦИЯ В СЕТЯХ НА КРИСТАЛЛАХ

Ю.В. Ладыженский, В.А. Мирецкая
Донецкий Национальный Технический Университет

В статье рассматривается применение муравьиного алгоритма AntNet для маршрутизации данных в сетях на кристалле. Приводится обобщение алгоритма для использования в однородных сетях на кристалле с любым количеством соединений между маршрутизаторами. Разработаны схемы, реализующие основные функции алгоритма.

Введение. Реализация в сверхбольших интегральных схемах (VLSI) архитектуры межкомпонентного соединения с помощью сетей на кристаллах позволяет создавать гибкие, программируемые и реконфигурируемые сети на процессоре [1]. Основные требования, предъявляемые к таким системам: малое потребление энергии и невысокая стоимость.

В работе [2] представлена схема маршрутизации в сетях на кристаллах, использующая алгоритм AntNet (см. [3]). Применение распределенного муравьиного алгоритма к маршрутизации данных в таких сетях уменьшает количество точек скученности (hot-spot), часто возникающих при так называемом «тяжелом трафике».

Алгоритм, описанный в [2], применим для сетей с однородной структурой маршрутизаторов, в которой каждый маршрутизатор соединен с соседними маршрутизаторами через четыре канала связи.

В статье приводится обобщение алгоритма для использования в сетях с любым количеством соединений между маршрутизаторами. Разработаны схемы, реализующие основные функции алгоритма.

Применение алгоритма antnet к сетям на кристаллах. Алгоритм AntNet [3] позволяет избежать перегрузок и появления точек скученности в сети, что является одной из основных проблем маршрутизации.

Агенты (муравьи), которые участвуют в исследовании сети, используют вероятностные правила выбора маршрутов и вероятностные таблицы маршрутизации [3].

Для упрощения использования вероятностей в аппаратной реализации предложено [2] кодировать значения вероятности целыми

числами в диапазоне [0, 4] (вероятность p=0.25 кодируется числом 1, вероятность 0.5 – числом 2 и т.д.), вместо привычного диапазона [0,1]. Значение 4 соответствует количеству линий связи исходящих из узлов. Однако в [2] используется только один тип сетей на кристаллах – тип Torus (см. рис 1). В [4] описаны сети, в которых количество направлений, в общем случае не равно 4. Соответственно верхняя граница диапазона вероятностей будет изменяться. Пусть *Dir* – количество направлений, по которым маршрутизатор может отправлять пакеты другим маршрутизаторам, тогда вероятность *P=[0..1]* можно закодировать целыми числами *[0..Dir]*, т.е. код $c \in [0..Dir]$ соответствует вероятности *P=c/Dir*.

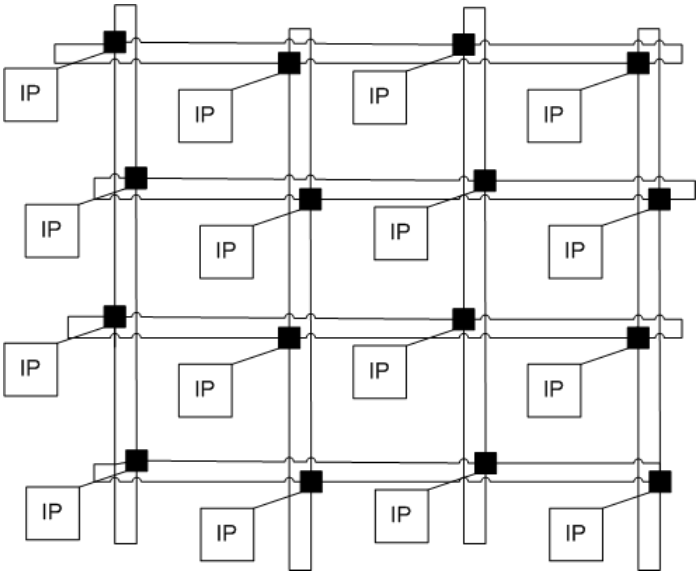


Рис. 1 – Сеть на кристаллах типа Torus

При движении муравья для выбора следующего узла используется формула [3]:

$$P'_{nd} = \frac{P_{nd} + \alpha \cdot l_n}{1 + \alpha(N_k - 1)} \qquad (1)$$

где P_{nd} – значение из таблицы маршрутизации, определяющее вероятность достижения из текущего узла k узла d через узел n; α – весовой коэффициент, $\alpha \in [0.2;0.5]$; l_n – состояние очереди пакетов между текущим узлом и узлом n; N_k – количество узлов соседних с k узлом.

Для упрощения аппаратной реализации формулы (1) в сетях на кристаллах в [2] предлагается использовать следующие значения: α = 1/3, N_k = 2.

Учитывая определенный выше параметр *Dir*, формулу (1) можно привести к виду:

$$P'_{jd} = \frac{3 \times P_{jd} + L_j}{2 + N_k} = \frac{3 \times P_{jd} + L_j}{2 + Dir} \qquad (2)$$

Для обновления таблиц маршрутизации в оригинальном алгоритме AntNet используются формулы [3]:

$$P_{fd'} := P_{fd'} + r \cdot (1 - P_{fd'}) \qquad (3)$$

$$P_{nd'} := P_{nd'} - r \cdot P_{nd'}, \quad n \in N_k, n \neq f, \qquad (4)$$

где $r \in (0,1]$ – коэффициент стабилизации, аналог феромонов. В [2] предложено вместо коэффициента стабилизации использовать состояние перегрузки узла – CS (congestion status). Значение CS определяется как сумма четырех Флагов Перегрузки (CF), т.е. CS – целое число, CS$\in$[0,4]. Каждый из флагов соответствует своему направлению в сети.

Для сетей с количеством направлений отличающихся от 4-х, значение CS будет определяться как $CS = \sum_{i=1}^{Dir} CF_i$. Коэффициент стабилизации $r = (Dir - CS)/Dir$. Таким образом, формулы (3) и (4) могут быть модифицированы:

$$P'_{fd} = P_{fd} + \frac{(Dir - CS) \cdot (Dir - P_{fd})}{Dir} = Dir - CS + CS \cdot P_{fd} \cdot Dir^{-1} \qquad (5)$$

$$P'_{nd} = P_{nd} - \frac{(Dir - CS) \cdot P_{nd}}{Dir} = CS \cdot P_{nd} \cdot Dir^{-1}, для \forall n \neq f \qquad (6)$$

Это преобразование позволяет реализовать совмещенную схему для вычисления значений двух формул. Т.е. блок вычисления P'_{nd} является составной частью блока вычисления P'_{fd} (см. рис. 2).

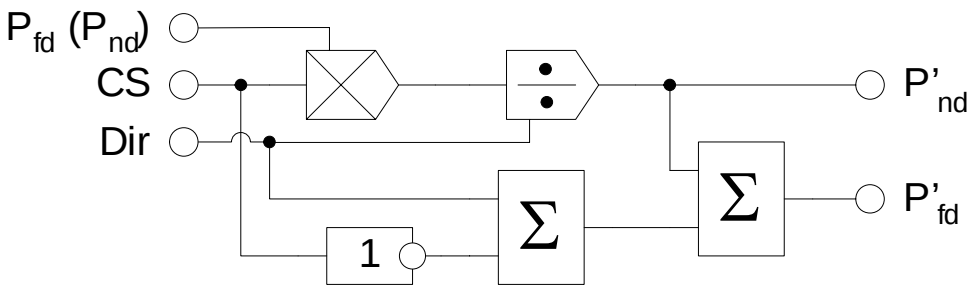


Рис. 2 – Блок обновления ячейки таблицы маршрутизации

В [2] для перехода F-муравья в следующий узел используется концепция «популярного узла», то есть такого узла, в который наиболее часто отправляются пакеты. Количество битов,

необходимых для кодирования адреса узла, зависит от количества узлов в сети.

В [2] в F-муравьях сохраняется путь их движения как список идентификаторов узлов, которые прошел муравей. Поэтому для отправки B-муравья каждый узел должен определять направление перемещения этого муравья по идентификатору узла. Это усложняет схему обработки B-муравьев. Вместо этого предлагается хранить в F-муравье последовательность закодированных направлений перемещений муравья. Направления должны быть закодированы таким образом, чтобы направление с кодом cd было противоположным направлению с кодом $\overline{cd}$. Например, для сети типа Torus (4 направления, называются Север, Юг, Запад, Восток [1]): Север – 00, Юг – 11, Запад – 01, Восток – 10. Такое решение упрощает схему выбора направления перемещения B-муравьев: для определения направления, необходимо инвертировать значение из маршрута F-муравья. Кроме того, уменьшается длина пакетов-муравьев.

Заключение. Использование сетей на кристаллах позволяет создавать гибкие, программируемые и реконфигурируемые сети на процессоре, к которым могут быть применены те же методы маршрутизации, что и к обычным компьютерным сетям. Алгоритм AntNet хорошо зарекомендовал себя как алгоритм, позволяющий равномерно распределить трафик по сети и, таким образом, избежать перегрузок [5, 6] и возникновения точек скученности данных.

Как показано в [2] использование этого алгоритма для сетей на кристаллах уменьшает среднюю задержку пакетов, и уменьшает вероятность возникновения точек скученности, особенно для модели трафика, приближенной к реальной.

Разработано обобщение алгоритма AntNet для использования в сетях с любым количеством соединений между маршрутизаторами. Разработаны схемы, реализующие основные функции алгоритма.

В будущем планируется провести исследование алгоритма AntNet, который не использует F-муравьев, т.е. функции F-муравьев выполняются B-муравьями. Предполагается, что это позволит снизить загрузку сети пакетами, не несущими полезных данных, передаваемых между ядрами процессора, и не приведет к значительной потере эффективности алгоритма.

Литература

[1] Dally W. J., Towles B. Route packets, not wires: on-chip interconnection Networks in Proc. DAC. - June 2001 - pp. 684-689.

[2] Daneshtalab M., Sobhani A., Mottaghi M. D., Kusha A.A., Navabi Z., Fatemi O. Ant Colony Based Routing Architecture for Minimizing Hot Spots in NOCs // Proceedings of the 19th annual symposium on Integrated circuits and systems design. 2006. P. 56-61

[3] Di Caro G., Dorigo M. AntNet: Distributed Stigmergetic Control for Communications Networks: Journal of Artificial Intelligence Research. 1998. №9. P. 317-365.

[4] V. Hahanov, O. Yegorov, K. Mostova. NoCs Design For Verification: Electronics and Electrical engineering. 2007. No. 3(75). P. 45-48.

[5] Ладыженский Ю.В., Мирецкая В.А. Исследование динамических алгоритмов маршрутизации в компьютерных сетях //Современные проблемы и достижения в области радиотехники, телекоммуникаций и информационных технологий: Тезисы докладов Международной научно-практичной конференции, г. Запорожье, 13-15 апреля 2006г. – Запорожье, ЗНТУ, 2006. – С.163-165.

[6] Ладыженский Ю.В., Мирецкая В.А. ИССЛЕДОВАНИЕ МУРАВЬИНЫХ АЛГОРИТМОВ МАРШРУТИЗАЦИИ В КОМПЬЮТЕРНЫХ СЕТЯХ // ІНТЕРНЕТ-ОСВІТА-НАУКА-2006, п’ята міжнародна конференція ІОН-2006, 10-14 жовтня, 2006. Збірник матеріалів конференції. Том 2. – Вінниця: УНІВЕРСУМ-Вінниця. 2006. С. 375-377.

Работа поступила в редакцию 24.05.07.

УДК 681.32

ОПТИМИСТИЧЕСКИЙ АЛГОРИТМ СИНХРОНИЗАЦИИ С ВНУТРИКОМПОНЕНТНЫМИ ОТКАТАМИ ДЛЯ РАСПРЕДЕЛЕННОГО МОДЕЛИРОВАНИЯ

Мирецкий А.В., Ладыженский Ю.В.
Донецкий Национальный Технический Университет

В статье представлен новый алгоритм отката для оптимистических алгоритмов синхронизации при распределенном дискретно-событийном моделировании систем. Описанный алгоритм основан на понятии компонентов. Он позволяет ускорить моделирование за счет уменьшения глубины выполняемых откатов, вплоть до отката состояния и отмены событий только одного компонента моделируемой системы.

Введение. Основная идея оптимистического алгоритма Time Warp заключается в том, что логические процессы обрабатывают события, «оптимистически» предполагая, что в результате этой обработки не возникнет ошибок причинности [1]. Если в процессе моделирования логический процесс принимает от другого процесса событие, которое должно произойти в прошлом, то выполняется возврат состояния моделируемой системы до этого момента в прошлом. Основные недостатки алгоритма Time Warp: 1) большие временные затраты на выполнение откатов; 2) Time Warp использует большие объемы памяти для хранения истории изменения состояния системы, необходимой для откатов.

Цель данной работы – снижение временных затрат на выполнение откатов за счет уменьшения глубины выполняемых откатов.

Локальные откаты. Одним из основных недостатков откатов в алгоритме Time Warp является отмена всех событий в рамках одного или нескольких логических процессов. По завершении отката, отмененные события должны быть выполнены заново. Чем глубже был откат, тем больше событий должно быть выполнено для достижения логическими процессами тех отметок модельного времени, которые были достигнуты до отката. Это может привести к значительным задержкам процесса моделирования.

Уже разработан ряд алгоритмов, позволяющих снизить временные издержки на выполнение откатов. Однако во всех

алгоритмах откат производится в одном или нескольких логических процессах в целом.

В [2] описан алгоритм SRADS with Local Rollbacks (Агрессивная обработка с локальными откатами), который позволяет полностью избавиться от анти-сообщений, инициализирующих откаты в соседних логических процессах. Т.е. все откаты будут локальными, внутри одного логического процесса. Каждый логический процесс выполняет агрессивную обработку событий (как в Time Warp), но результаты этой обработки (новые сообщения) не передаются другим логическим процессам до тех пор, пока не будет гарантий, что они никогда не будут отменены. Сообщение никогда не будет отменено, если его временная метка не превышает глобальное виртуальное время.

Внутрикомпонентные откаты. Предлагаемый алгоритм основан на алгоритме локальных откатов, и позволяет снизить глубину этих локальных откатов.

Известно, что многие сложные системы можно представить множеством более простых элементов (подсистем), взаимодействующих между собой. Такие элементы назовем компонентами. В терминах дискретно-событийного моделирования, компоненты способны обрабатывать и, при необходимости, порождать новые события.

Таким образом, модель представляется набором компонентов, связанных между собой определенным образом. Связанные компоненты могут обмениваться между собой событиями. Формально, модель представляется графом $M=(C, L)$, где $C=\{c\}$ – множество компонентов модели; $L=\{l \mid l=(c_k, c_m), c_k, c_m \in C, c_k \neq c_m\}$ – множество связей между компонентами.

Физическая система представляется множеством физических процессов, взаимодействующих между собой. Каждый физический процесс моделируется логическим процессом, а взаимодействие между физическими процессами моделируется обменом сообщениями, имеющими временные метки, между соответствующими логическими процессами. [1]

Каждый логический процесс выполняется на отдельном процессоре.

Пусть $LP = \{LP_i\}$, $|LP|=N$ – множество логических процессов, выполняемых на N процессорах. Модель М разделена на подмодели, моделируемые логическими процессами:

$$S=\left\{s_i \mid s_i=(C_i,L_i), s_i \subseteq M, s_i \cap s_j=\varnothing\ \forall i,j: i\neq j, \bigcup_i s_i \cup(\varnothing, L_{bound})=M\right\},$$

где s_i – подмодель, моделируемая логическим процессом LP_i; $C_i \subseteq C$ – множество компонентов подмодели s_i; $L_i \subseteq L$ – множество связей между компонентами подмодели s_i; $L_{bound} \subseteq L$, $L_{bound}=\{l_{bound} \mid l_{bound} = (c_k, c_m),\ c_k \in s_i,\ c_m \in s_i,\ i\neq j\}$ – множество связей между компонентами, находящимися в разных подмоделях. |S|=N.

Каждый логический процесс LP_i характеризуется локальным модельным временем T_i, которое представляет собой модельное время, достигнутое LP_i в процессе моделирования. В общем случае локальные модельные времена процессов отличаются друг от друга: $T_i \neq T_j$, $i\neq j$.

Пусть $B = \{b \mid b \in C,\ \exists\ l=(c_k, c_m): (b\equiv c_k \text{ или } b\equiv c_m) \text{ и } l \in L_{bound}\}$ – множество пограничных компонентов, т.е. таких компонентов, которые способны обмениваться сообщениями с другими пограничными компонентами, моделируемыми другими логическими процессами.

Пусть e – событие; $e = (\tau_e, \sigma_e, c_e, h_e)$, где τ_e – временная метка, показывающая, когда должно произойти событие; σ_e – временная метка, показывающая, когда было порождено событие; c_e – компонент, породивший событие; h_e – компонент, который должен обработать событие. $\sigma_e \leq \tau_e\ \forall\ e$, т.е. событие не может быть порождено позже, чем будет обработано.

Любое событие может быть следствием только обработки другого события. Событие не может быть порождено самостоятельно, т.е. между событиями имеются причинно-следственные связи. Если $e_1 = (\tau_1, \sigma_1, c_1, h_1)$ – событие-причина, а $e_2 = (\tau_2, \sigma_2, c_2, h_2)$ – событие-следствие, то $h_1=c_2 \cup \tau_1 = \sigma_2$, т.е. событие-следствие порождается в тот же момент времени, когда происходит обработка события-причины, и событие-следствие порождается тем же компонентом, который обрабатывает событие-причину.

Каждый логический процесс моделируется последовательно на одном процессоре. Пусть LP_1, LP_2 – некоторые логические процессы, C_1, C_2 – множества компонентов моделируемых, соответствующими процессами, B_1, B_2 – пограничные компоненты соответствующих логических процессов. Пусть компонент $c' \in B_1$ порождает некоторое событие $e=(\tau, \sigma, c, h)$, тогда $\sigma=T_1$, $c=c'$, $\tau\geq\sigma \rightarrow \tau\geq T_1$. Если $h\in B_2$, то метка τ может быть любой по отношению к T_2. Если $h\in B_2$ и $\tau<T_2$, то событие e называется отставшим событием.

Если логический процесс принимает отставшее событие, то это событие нарушает ограничение причинности [1]. Логический процесс должен восстановить причинно-следственные связи между событиями.

Рассмотрим процесс отката при получении процессом LP_2 отставшего события из LP_1 (см. рис. 1). Пусть $C_2 = \{c_1, c_2, c_3, c_4\}$, $B_2=\{c_1\}$. Компонент c_1 получает отставшее событие $e_s=(\tau_s, \sigma_s, c_s, h_s)$, где $h_s = c_1$ и $\tau_s < T_2$. Алгоритм Time Warp предусматривает отмену всех событий $e_i=(\tau_i, \sigma_i, c_i, h_i) \mid \forall i: \tau_i > \tau_s$, и откат состояния всего логического процесса до отметки τ_s.

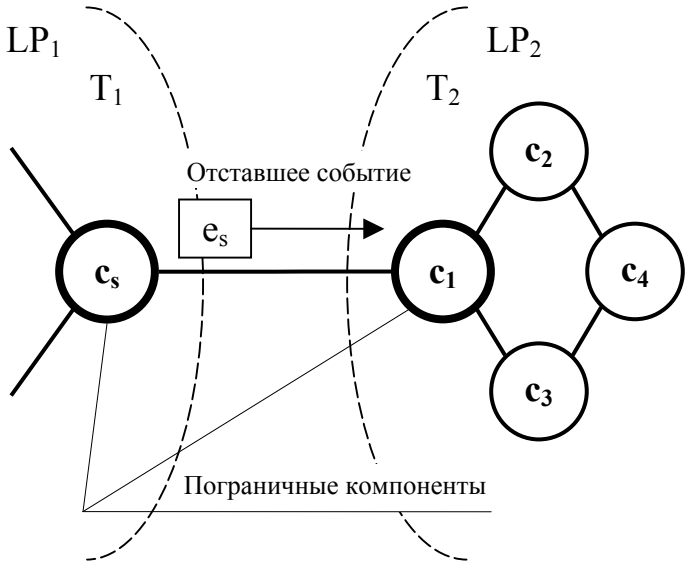


Рис. 1 – Взаимодействие двух логических процессов

Назовем внутрикомпонентным откатом для компонента c откат состояния и отмену событий $e_i=(\tau_i, \sigma_i, c_i, h_i)$ таких, что $\forall i: h_i = c$ или $c_i=c$. Пусть $E_{ROLLBACK}(c, \tau) = \{e_i=(\tau_i, \sigma_i, c_i, h_i) \mid h_i = c$ и $\tau_i > \tau\}$ – множество событий, отмененных для компонента c, т.е. множество событий, отмененных в процессе внутрикомпонентного отката.

Если не существует такого события $e_i=(\tau_i, \sigma_i, c_i, h_i)$, $e_i \in E_{ROLLBACK}(c_1, \tau_s)$, что $h_i \neq c_1$ и $h_i \in C_2$, то компонент c_1 гарантированно не оказывал влияния ни на один компонент множества C_2, кроме самого себя. Как следствие, в алгоритме Time Warp все отмененные события, не принадлежащие множеству $E_{ROLLBACK}(c_1, \tau_s)$, будут в точности повторены. Это означает, что откат этих событий не имеет смысла. В этом заключается ключевая идея предлагаемого алгоритма внутрикомпонентных откатов.

Рассмотрим случай, когда компонент c_1 оказывает влияние на компонент c_2. Это означает, что $\exists\ e_i=(\tau_i, \sigma_i, c_i, h_i), e_i \in E_{ROLLBACK}(c_1, \tau_s)$: $ci=c_1$, $hi=c_2$. Такие события могут быть причиной изменения состояния компонента c_2 или причиной порождения этим компонентом новых событий. Следовательно, в компоненте c_2 также необходимо выполнить откат.

Если учесть, что $\forall e_i=(\tau_i, \sigma_i, c_i, h_i)\ \tau_i \geq \sigma_i$, и учесть связь между событием-причиной и событием-следствием, то можно сказать, что отмененные в c_1 события не могли быть причиной событий, порожденных ранее в компоненте c_2. В общем случае, $\tau_i > \sigma_i$, что дает возможность уменьшить глубину отката для компонента c_2. События $e_i=(\tau_i, \sigma_i, c_i, h_i), e_i \in E_{ROLLBACK}(c_1, \tau_s)$ могут быть причиной только таких событий $e_j=(\tau_j, \sigma_j, c_j, h_j)$, что $\sigma_j \geq \min_i \tau_i\ \forall j$. Следовательно, откат в компоненте c_2 должен быть выполнен до отметки $\min_i \tau_i$. Согласно условию причинно-следственных связей событий, $\min_i \tau_i \geq \tau_s$. В общем случае, выполняется условие неравенства, что является определяющим фактором уменьшения глубины отката в отдельных компонентах.

Аналогичным образом возможно и влияние между другими компонентами. Если факт влияния установлен, то необходимо выполнить те же действия, что описаны выше. Если имеет место обратное влияние, т.е., например, c_1 повлиял на c_2, а c_2 в свою очередь повлиял на c_1, то его рассматривать нет необходимости. Это связано с тем, что откат компонента c_1 уже выполнен, и рассмотрение обратного влияния уже не уменьшит глубину отката.

Чем глубже будет уходить процесс отката от компонента, получившего отставшее событие, тем меньше будет глубина внутрикомпонентного отката (для общего случая). В худшем случае будут отменены события для всех компонентов до момента времени τ_s, что эквивалентно откату в алгоритме Time Warp.

После выполнения отката необходимо повторить моделирование всех отмененных событий $e_i=(\tau_i, \sigma_i, c_i, h_i)$ таких, что $\sigma_i < \tau_s$. Эти события были созданы до того, как произошло событие e_s, Следовательно, событие e_s не может стать причиной отмены в прошлом этих событий.

Следует учесть, что в процессе обработки отставшего события или в результате повторной обработки отмененных событий могут быть порождены новые события, предназначенные для других компонентов. Если откат этих компонентов не был выполнен, то

новые события станут для них отставшими. В этом случае процесс отката следует повторить.

Алгоритм внутрикомпонентных откатов используется в системе распределенного моделирования алгоритмов маршрутизации [3, 4] для повышения скорости обработки дискретных событий.

Заключение. В статье разработан новый алгоритм внутрикомпонентных откатов. Он позволяет ускорить распределенное моделирование за счет уменьшения глубины откатов. Если отставшее событие с временной меткой τ_s должно быть выполнено компонентом, который не оказывал влияния на другие компоненты, то это будет лучший случай. Откат будет выполнен только для одного компонента. Если же компонент был причиной изменения состояния или создания новых событий всех компонентов логического процесса в момент времени τ_s, то результат работы алгоритма будет эквивалентен откату в алгоритме SRADS with Local Rollbacks.

Литература

[1] R. M. Fujimoto, Parallel and Distributed Simulation Systems, Wiley Interscience, 2000.

[2] P. Dickens and P. Reynolds, Jr. SRADS with local rollback. In Distributed Simulation, volume 22, pages 161--164. SCS Simulation Series, Jan. 1990.

[3] Ладыженский Ю.В., Мирецкий А.В. МОДЕЛИРОВАНИЕ АЛГОРИТМОВ ДИНАМИЧЕСКОЙ МАРШРУТИЗАЦИИ // Сучасні проблеми і досягнення в галузі радіотехніки, телекомунікацій та інформаційних технологій: Тези доповідей Міжнародної науково-практичної конференції м. Запоріжжя, 13-15 квітня 2006 року / Під заг. ред. Д.М. Пізи. – Запоріжжя: ЗНТУ. 2006. С. 165-167.

[4] Ладыженский Ю.В., Мирецкий А.В. ПРОГРАММНАЯ СИСТЕМА ДЛЯ МОДЕЛИРОВАНИЯ АЛГОРИТМОВ ДИНАМИЧЕСКОЙ МАРШРУТИЗАЦИИ // ІНТЕРНЕТ-ОСВІТА-НАУКА-2006, п’ята міжнародна конференція ІОН-2006, 10-14 жовтня, 2006. Збірник матеріалів конференції. Том 2. – Вінниця: УНІВЕРСУМ-Вінниця. 2006. С. 372-374.

Получено 01.06.07

УДК 681.3

МУРАВЬИНЫЙ АЛГОРИТМ ДЛЯ РАЗБИЕНИЯ ГРАФОВ

Ю.В. Ладыженский , Р.А. Родригес Залепинос
Донецкий национальный технический университет

Реализован алгоритм с использованием парадигмы колонии муравьёв для поиска и стягивания подмножеств вершин (кластеров) при укрупнении графа в многоуровневой схеме разбиения графов. Описана методика проведения экспериментов и представлены их результаты.

Проблема разбиения графов возникает в различных формах в data mining, робототехнике, искусственном интеллекте, переупорядочивании разреженных матриц, обработке изображений, компьютерных сетях, параллельных вычислениях, оптимизации кэширования, проектировании СБИС, базах данных, операционных системах и других областях.

Пусть дан неориентированный граф $G = G(V, E)$, где V − множество вершин, имеющие веса, $|V'|$ сумма весов вершин из $V' \subseteq V$, а E − множество рёбер (v, w) с весами $w(v, w) \in \Re$, $w(V_1, V_2) = \sum_{v_1 \in V_1, v_2 \in V_2} w(v_1, v_2)$, $V_1, V_2 \in V$. Для $(v, w) \notin E$ $w(v, w) = 0$, Проблема разбиения графа на k ($k \in N$, $k \geq 2$) частей и заданного баланса $\Theta \in \Re$, $\Theta > 0$ состоит в поиске набора попарно непересекающихся подмножеств $V_1 \cup V_2 \cup ... \cup V_k = V$ таких, что $\max(k|V_i|/|V|,1) - 1 \leq \Theta$, и минимизации $w(V_i, V_j)$ $\forall i, j$.

В практических задачах возникают графы с большим числом вершин ($\approx 2 \cdot 10^3 \div 7 \cdot 10^6$), поэтому для их разбиения успешно применяется многоуровневая парадигма [1-5]. Она позволяет получать «хорошие» решения за «разумное» время. Вначале исходный граф G обрабатывается рекурсивной функцией, результатом работы которой является последовательность графов $G, G_1, G_2, ..., G_n, G'$, где каждый последующий граф получается укрупнением предыдущего графа, путём попарного стягивания вершин. Граф G' разбивается на требуемое число подмножеств. Затем выполняются обратные действия, назначая вершинам v и w, стянутым в вершину u то же подмножество разбиения, что и для u. При укрупнении стягиваются вершины, вероятность которых быть помещёнными в разные подмножества разбиения мала. Процесс попарного стягивания имеет некоторую природу случайности и локальность, поэтому зависит от нумерации вершин. Предлагается вместо этого стягивать сразу

несколько вершин (кластеры), в надежде, что это устранит зависимость от нумерации вершин и повысит качество решения.

Реализованный алгоритм состоит из двух этапов. Этап 1 является муравьиным алгоритмом, цель которого изменить веса рёбер так, чтобы потом легко отличить кластеры. Этап 2, используя новые веса рёбер, выделяет кластеры вершин и стягивает их.

Муравьи являются социальными насекомыми, живущими в высокоорганизованных колониях. Муравьи могут находить кратчайшие пути между источниками пищи и своим гнездом. Когда муравьи перемещаются от источника пищи к гнезду и в обратном направлении, они выделяют химическое вещество феромон, которое остаётся на земле, формируя след. Двигаясь, муравьи выбирают, с некоторой вероятностью, пути с сильными концентрациями феромона. След феромона помогает муравьям находить путь обратно к гнезду и достигать источников пищи, найденных своими собратьями. Феромон имеет свойство накапливаться и его концентрация на пути пропорциональна числу прошедших по нему муравьёв. Со временем феромон испаряется с редко посещаемых путей. Искусственные муравьи обладают некоторым объёмом памяти, функционируют в дискретном времени, перемещаются по рёбрам графа и способны анализировать списки смежности вершин графа [6-8].

Пусть на вход Этапа 1 подан граф $G = G(V, E)$. Допустим, муравей выбирает вершину для своего следующего шага случайно, причём вероятность выбрать $w \in V$ пропорциональна весу $w(v, w)$, если муравей находится в $v \in V$. Можно предположить, что если v принадлежит кластеру $C \subseteq V$, тогда вероятность из v попасть в $w \in C$ должна быть высока $p = w(v, C \cap Adj[v]) / w(v, Adj[v])$, где $Adj[v]$ множество вершин, смежных с v. Следовательно, C захватывает муравьёв: попав в кластер, они будут долго циркулировать в нём; за продолжительное время число покинувших кластер муравьёв много меньше прибывших и находившихся в нём за это время.

Будем использовать новую функцию $w'(v, w) = w(v, w) * pher(v, w)$, определяющую вес ребра, где $pher(v, w) \in \Re, pher(v, w) \geq 1$ − концентрация феромона на ребре (v, w). Муравей помнит последние Δt_2 вершин своего пути, для чего сохраняет их в очереди $q = < v_1, v_2, ..., v_{\Delta t_2 - \Delta t_1}, v_{\Delta t_2 - \Delta t_1 + 1}, ..., v_{\Delta t_2} >$, и ему запрещено возвращаться в вершины, посещённые за последние Δt_1 шагов, $\Delta t_1 < \Delta t_2$. Вероятность перехода муравья в вершину, смежную

с $v_{||q||}$ пропорциональна её связности с ранее посещёнными $\sum_{w \in Adj[v_{||q||}], v \in q \cap Adj[w]} w'(v,w)$, где $||q||$ число вершин в q. Когда q заполнена Δt_2 вершинами, то при занесении новой вершины её индекс равен Δt_2, и $v_i = v_{i+1}, i = 1,...,\Delta t_2 - 1$. Когда муравей переходит в вершину $v_t \in q$, то фиксируется факт обнаружения кластера. На основании вершин $v_t,...,v_{||q||}$ выделяется подграф $G' = G'(V',E')$, $V' \subseteq V$, $v_t,...,v_{||q||} \in V'$, $E' = \{(v,w) : v,w \in V', (v,w) \in E\}$. Очевидно, в общем случае $||E|| > ||V'||$, т.е. E' состоит не только из рёбер цикла.

Когда все муравьи сделают шаг, происходит изменение значения феромона во всех найденных кластерах на этом шаг: новое значение $pher(v,w)$ полагается равным старому значению $pher(v,w) + \varepsilon \quad \forall (v,w) \in E'$, что является косвенной мерой качества найденного решения. Из очереди удаляются вершины с индексом большим t, при этом муравей остаётся в вершине v_t. За всё время работы алгоритма феромон не испаряется.

Этап 2 рассматриваемого алгоритма в модифицированном графе $G = G(V,E)$ классифицирует вершины на граничные (B), внутренние (I) и не принадлежащие какому-либо кластеру (C). Множества $B, I, C \subseteq V$, попарно не пересекаются. Если

$$\max_k w'(v, Adj[v][k]) / w'(v, Adj[v][k+1]) \geq \varphi_{\max} \qquad (1)$$

то $v \in B$ где $\varphi_{\max}$ – некоторый параметр, обычно $\varphi_{\max} \geq 2$, а $Adj[v][i]$ индекс i-ой смежной вершины с v, причём $w'(v, Adj[v][i]) \leq w'(v, Adj[v][i+1])$ для $\forall v \in V$. Предполагается, что индекс k, найденный из неравенства (1) для вершины v отделит особо тяжёлые от резко отличающихся от них лёгких рёбер. Считается, что если $v \in B$, то v принадлежит пока что неизвестному кластеру Q, и $Adj[v][i] \in Q$, $i = \overline{0,k}$. Если для v неравенство (1) не может быть удовлетворено ни при каком k, то $v \in I \cup C$. Пусть для $v \in B$ $v^+ = \{Adj[v][i] : i = \overline{0,k}\}$, а $v^- = Adj[v] \setminus v^+$. Для $v \in I \cup C$ $v^+ = Adj[v]$.

Путь $< x_1, x_2, ..., x_n >$ в графе G положителен ($x_i \in V, i = \overline{1,n}$), если $x_{i+1} \in x_i^+$, $i = \overline{1, n-1}$. Множество $Q \subseteq V$ максимально, если для $\forall v \in V \setminus Q$ не существует положительного пути от v к w, если $v \in B$ либо от z к w через v, если $v \notin B$, где $w \in Q$, а $z \in B$ (т.е. $\forall v \notin Q$ нельзя присоединить к Q). Также, если $v \in Q$ и $w \in v^+$, то $w \in Q$. Если

$\exists Q' \subset Q$, $Q' \neq \varnothing$ такое, что Q' максимально, то Q избыточно. Подмножество $Q \subseteq V$ называется кластером (с точки зрения Этапа 2), если оно максимально и не избыточно. Этап 2 выделяет кластеры из V (можно доказать, если Q_1 и Q_2 – различные кластеры, то $Q_1 \cap Q_2 = \varnothing$). В одном кластере не может быть рёбер, сильно отличающихся друг от друга по весу. Пусть величина me равна весу максимального ребра в найденном кластере Q. Если $v \in Q$ и $me / Adj[v][0] \geq \varphi_{\max}$, то v удаляется из Q. В общем случае, подграф из вершин Q графа G становится несвязным и распадается на части $Q_1, Q_2, ..., Q_n$, которые стягиваются по отдельности.

Реализованы оба этапа описанного выше алгоритма. Для задания графа в пакетах разбиения графов используются текстовые файлы. В них построчно помещаются списки смежности вершин графа. Если для изображённого на бумаге графа выполнять это вручную, то легко допустить ошибку. Существуют утилиты, позволяющие задать граф визуально, однако они хранят описание графа в файлах, форматы которых специфичны для каждой утилиты.

Для проведения экспериментов, визуального наблюдения за процессом работы алгоритма и снятия результатов в наглядной форме используется PowerPoint. Он является COM-сервером, которым можно управлять непосредственно из собственного приложения и получать доступ к изображённому на слайде графу (с помощью эллипсов и соединительных линий) через соответствующие классы (как если бы граф был уже загружен из файла). При работе Этапа 1 после выполнения заданного числа итераций, можно визуализировать интенсивность феромона на рёбрах через толщину соединительных линий. Если граф обрабатывает один муравей, то можно проследить пошагово его перемещение из вершины в вершину (текущая вершина и вершины очереди выделяются особыми цветами).

Проведенные эксперименты на небольших графах (в PowerPoint) показывают, что муравьи ошибаются (в силу своей стохастической природы), наращивая феромон на рёбрах, не входящих в кластеры: некоторые вершины становятся граничными, а некоторые перестают быть таковыми. При этом вершины кластеров становятся неотличимыми от соседних, сливаясь на Этапе 2 в огромный кластер, вызывая большой разброс в весах попавших в него рёбер, приводя к удалению практически всех вершин из кластера. Находится несколько огромных кластеров, поэтому стягивается ничтожное число вершин по сравнению с их общим количеством, а укрупнённый граф по числу вершин получается не намного меньше

исходного. Для уменьшения числа ошибок возможна модификация правил движения муравья. Также следует разработать более чувствительные схемы построения кластеров.

Проведены эксперименты с использованием METIS [9] на графах из архива C. Walshaw [10]. После Этапа 1 обработки графа он посылался в METIS для нахождения бисекции. Находилась также другая бисекция, последовательно обрабатывая тот же граф Этапом 1, Этапом 2 и METIS. Лучшая из этих двух бисекций сравнивалась с бисекцией, полученной непосредственным применением METIS к исходному графу. Для графа «data» получено разбиение на 5% лучше, но с ухудшением баланса на 2%. Для графа «add32» разбиение улучшено на 16%, а баланс на 2.78%. Это разбиение совпало с лучшим из известных на сегодняшний день.

Предложенный метод по качеству получаемого решения сравним с существующими. Он может применяться для негарантированного улучшения качества разбиений, находимых METIS, дополнительно затрачивая несколько минут на предварительную обработку графа описанным алгоритмом.

Литература

1. Shloegel K., Karypis G., Kumar V., Parallel multilevel algorithms for multi-constraint graph partitioning, 1999
2. Soper A.J, Walshaw C., Cross M., A combined evolutionary search and multilevel optimization approach to graph partitioining, 2004
3. Karypis G., Kumar V., A fast and high quality multilevel scheme for partitioning irregular graphs, 1998
4. Karypis G., Kumar V., Multilevel k-way partitioning scheme for irregular graphs, 1998
5. C. Walshaw, Multilevel refinement for combinatorial optimization problems, 2004
6. Dorigo M., Real ants inspire ant algorithms, 2004
7. Dorigo M., Maniezzo V., Colorni A., The ant system: optimization by a colony of cooperative agents, 1996
8. Dorigo M., Caro G., Gambardella L., Ant algorithms for discrete optimization, 1999
9. http://www.cs.umn.edu/~karypis/metis
10. http://staffweb.cms.gre.ac.uk/~c.walshaw

Получено 01.06.07

ПРОГРАММНАЯ СИСТЕМА ДЛЯ РЕШЕНИЯ ЗАДАЧ ОПТИМИЗАЦИИ НА ГРАФАХ С ИСПОЛЬЗОВАНИЕМ ГЕНЕТИЧЕСКИХ АЛГОРИТМОВ

О.А.Старосоцкая, Ю.В.Ладыженский
Донецкий национальный технический университет

Разработана система, покрывающая широкий класс задач оптимизации на графах. В основе решения лежат теоретические основы генетических алгоритмов. Вычисления осуществляются с заданными пользователем параметрами. Ход решения визуализируется с определенной пользователем частотой.

Введение. Графы являются удобным инструментом для наглядного представления различных объектов и процессов. Их применение способствует простоте визуализации задач различной сложности и доступности для широкого понимания методов их решения. Поэтому задачи оптимизации на графах и поиск эффективных методов их решения представляют особый интерес.

Граф задается множеством вершин и множеством ребер, соединяющих некоторые пары вершин. Для решения задач графы могут быть представлены матричным способом, с помощью списка, графическим способом.

Задачи оптимизации на графах относят к классу NP-полных задач, поэтому классические методы их решения не всегда дают желаемый результат. Для их решения предлагается использовать генетические алгоритмы, которые представляют собой адаптивный поиск, основанный на селекции лучших элементов в популяции, подобно эволюции Ч.Дарвина.

Оптимизационные задачи и генетические алгоритмы. Генетические алгоритмы работают с набором некоторой закодированной информации – популяцией. Каждый элемент популяции представляет собой хромосому, а часть хромосомы - ген.

Для каждой задачи заранее задан вид хромосомы, способ кодирования. Для задачи о коммивояжере могут использоваться 2 вида кодировки: матричный и в виде списка последовательности посещаемых вершин. Для кодирования решений задачи о раскраске также используется битовая строка, сообщающая о смежности вершин. Кодирование в задаче определения планарности может основываться на матрице линейно независимых циклов.

Оптимизации производится на основе целевой функции , путем последовательного применения различных генетических операторов (селекции, кроссовера, мутации), как классических, так и модифицированных. Особый интерес представляет «жадный» оператор кроссинговера: он позволяет создавать новые решения на основе частичного выбора на каждом шаге преобразования локально оптимального значения.

Целевая функция для задачи о коммивояжере имеет вид:

$$F(\varphi) = \{R(\varphi(1), \varphi(n)) + \sum_i \{R(\varphi(i), \varphi(i+1))\} \ , \qquad (1)$$

где Rij – стоимость переезда из i-ой вершины в j-ую.

В задаче определения планарности оптимизация производится на основе аддитивной функции двух переменных :

$$f(H) = (\delta(m); L_0), \qquad (2)$$

где L_0 - длина незначащей части хромосомы, $\delta(m)$- функция пригодности.

Организация программной системы. Эти сведения легли в основу разработки программного комплекса, позволяющего решать задачи оптимизации на графах различной сложности с помощью генетических алгоритмов. Целью разработки является предоставление вычислительных преимуществ генетических алгоритмов, которые более эффективны по сравнению с другими методами решения подобных задач. Система не только показывает результат решения, но и предоставляет возможность частичной или полной визуализации хода решения.

Функциями системы предусмотрено изменение различных параметров решения задачи. Существует возможность указания вероятностей применения различных генетических операторов: селекции, кроссинговера, мутации. Для некоторых задач возможен выбор конкретных операторов из списка предложенных, например, в качестве основного оператора селекции используется селекция на основе рулетки, которая может быть изменена пользователем на элитную или турнирную. Размер популяции и количество поколений могут определяться пользователем или ограничениями решаемой задачи. Пользователь может задавать исходный граф всеми описанными способами.

Система позволяет загружать графы из файла и сохранять их. Файлы графов имеют расширение *.grf. Файлы *.opt содержат трассировки хода и результаты решения задач оптимизации.

Для достижения лучших результатов при решении используются специфичные для данной предметной области генетические операторы. В задаче о коммивояжере используется «жадный» оператор кроссинговера, а планарность определяется при помощи двухточечного суммирующего оператора кроссинговера.

Применение того или иного генетического оператора определяется вероятностным методом. Вероятности задаются системой или пользователем.

При решении задач используются основные математические формулы и теоремы. Например, планарность определяется на основе критерия Мак-Лейна, согласно которому для планарного графа в матрице линейно независимых циклов можно выделить подматрицу размерностью m-n+2 на m, где m- число ребер, а n – вершин, в столбцах которой содержится строго две единицы.

Структура системы представляет собой совокупность взаимодействующих модулей и обусловлена логикой постановки задачи и требованиями удобного и наглядного представления (рис. 1).

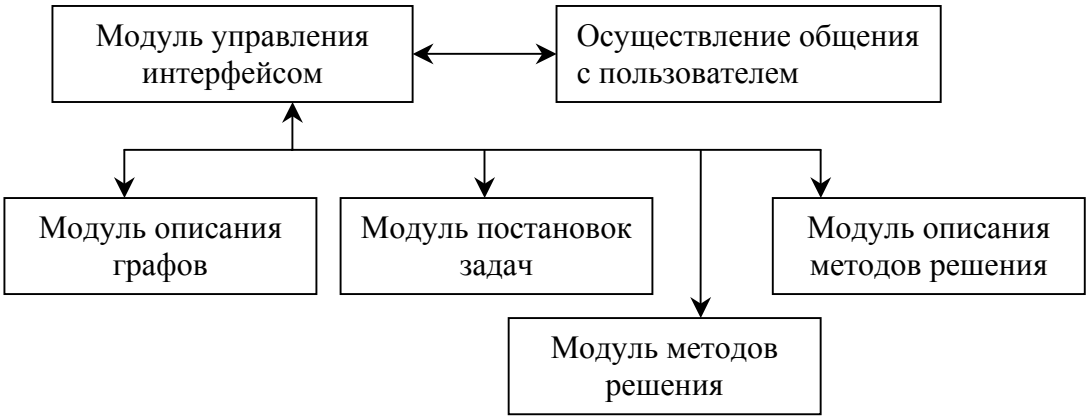


Рис. 1-Взаимодействие модулей системы

Пользовательский интерфейс представлен на рисунке 2.

Меню содержит как совсем привычные пункты меню (Файл, Помощь, Опции), так и специфичные для данного программного продукта. Для удобства работы используются следующие панели:

- списка созданных и загруженных графов;
- информации о текущем графе;
- панель параметров решаемой задачи.

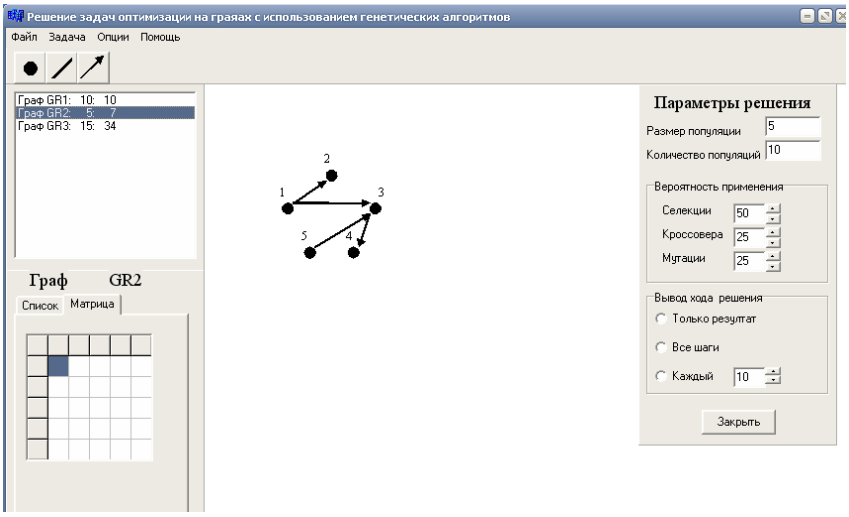


Рис. 2 – Пользовательский интерфейс приложения

Выводы. Разрабатываемая программная система позволяет решать задачи оптимизации на графах с использованием генетических алгоритмов. С ее помощью можно найти оптимальные решения для задачи о коммивояжере, определения планарности, раскраски графов.

К достоинствам разработанного продукта относятся:

- возможность визуализации хода решения;
- задание пользователем параметров решения;
- различные способы задания графов;
- возможность создания библиотеки графов;
- использование актуальных и перспективных генетических методов для решения задач;
- возможность расширения библиотеки методов решения;

Литература:

1. Гладков Л.А, Курейчик В.М., Курейчик В.В. Генетические алгоритмы.- М:Физмалит,2006.
2. Кристофидес Н. Теория графов. Алгоритмический подход. – М:Мир,1978.

Получено 01.06.07

УДК 681.5

АЛГОРИТМИЧЕСКИЕ МЕТОДЫ ПОВЫШЕНИЯ ЭФФЕКТИВНОСТИ ПАРАЛЛЕЛЬНИХ ВС ПРИ ЧИСЛЕННОМ РЕШЕНИИ СОДУ С КОНТРОЛЕМ ПОГРЕШНОСТИ НА ШАГЕ

И.А. Назарова
Донецкий национальный технический университет

В статті розглянуті програмно-алгоритмічні засоби підвищення ефективності паралельних ОС при чисельному вирішенні систем звичайних диференційних рівнянь з контролем локальної похибки. Розроблені ефективні обчислювальні схеми відображення методів на паралельні структури різної топології. Отримані порівняльні характеристики потенційного та реального паралелізму, проведені чисельні експерименти.

Применение параллельных вычислительных систем является стратегическим направлением развития современной компьютерной индустрии. Это обстоятельство вызвано не только принципиальным ограничением максимально возможного быстродействия обычных последовательных машин, но и практически постоянным существованием вычислительных задач, для решения которых возможностей существующих средств вычислительной техники всегда оказывается недостаточно. К таким задачам относится решение систем обыкновенных дифференциальных уравнений большой размерности, используемое при моделировании различных динамических процессов с сосредоточенными параметрами.

В данной статье приводятся параллельные алгоритмы одношаговых методов численного решения задачи Коши, которые могут применяться для решения, как жестких, так и нежестких задач, а также исследуется эффективность их отображения на параллельные структуры различной архитектуры и топологии.

Для системы из m ОДУ:

$$\begin{cases} \bar{y}' = F(x, \bar{y}); \\ \bar{y}(x_0) = \bar{y}_0. \end{cases},$$

где $\bar{y} = (y_1, y_2, ..., y_m)^T$, $F = (f_1, f_2, ..., f_m)^T$, правая часть системы есть в общем случае нелинейная функция $F = \bar{f}$, задающая отображение: $F : R \times R^m \to R^m$.

Одношаговый метод типа Рунге-Кутты для решения нелинейной задачи Коши, описываемой СОДУ, имеет вид:

$$\begin{cases} \bar{y}_{n+1} = \bar{y}_n + h \cdot \sum_{i=1}^{s} b_i \cdot \bar{k}_i, \\ \bar{k}_i = F(x_n + c_i h, \bar{y}_n + h \sum_{j=1}^{s} a_{ij} \bar{k}_j), \ i = \overline{1,s} \end{cases},$$

s-размерные вектора $c = (c_1, c_2, ..., c_s)^T$, $b = (b_1, b_2, ..., b_s)^T$ и $s \times s -$матрица $A = (a_{ij}), i, j = \overline{1,s}$ описывают уникальный вариант метода и выбираются из соображений точности. Вид матрицы A задает тип численной схемы, положенной в основу метода. Если $a_{ij} = 0$ при $i \le j$, то метод является явным и при наличии большой степени параллелизма, обладает условной устойчивостью. При полностью заполненной матрице имеем полностью неявную схему, применяемую для решения жестких задач. К достоинствам полностью неявных, одношаговых методов общего вида (на основе квадратурных формул Радо и Лобатто [1]) следует отнести хорошие характеристики устойчивости и точности, достаточные для решения жестких задач. Так например, $s-$ стадийный метод РадоIA имеет порядок практически в 2 раза больше, чем число стадий и обладает А-устойчивостью. Недостатками этих методов является высокая вычислительная сложность, обусловленная итерационным процессом определения стадийных коэффициентов, которая не позволяет эффективно применять эти методы на последовательных машинах. При решении СОДУ с использованием полностью неявных методов Рунге-Кутта все $m \cdot s$ неизвестных: $K = \|k_{ij}\|, i = 1,..,s;\ j = 1,..,m$ должны определяться одновременно, что существенно усложняет задачу (всего шаговых коэффициентов s и размерность каждого вектора $\bar{k}_i$ равна m). Заметим, что характеристики параллелизма для всех одношаговых методов не зависят от того, какой конкретно метод используется, поэтому исследования проводятся в общем виде.

С целью минимизировать вычислительные затраты для достижения некоторой заданной точности приближенного решения, численный алгоритм должен иметь механизм управления шагом интегрирования. Это возможно на основе информации об апостериорной локальной погрешности. Для реализации поставленной цели использовались альтернативные способа оценки погрешности на шаге:

1) дублирование шага по правилу Рунге;
2) вложенные пары схем;
3) технология локальной экстраполяции Ричардсона.

Учет специфики задачи позволяет во многих случаях строить

более эффективные параллельные вычислительные алгоритмы, чем в случае применения стандартных численных схем. Поэтому отдельно рассматривается параллельное решение линейной задачи, в частности, предложены специальные методы решения однородных и неоднородных СЛОДУ с постоянными коэффициентами на основе экспоненциального метода [2] и эффективных алгоритмов умножения матриц.

Разработка параллельных алгоритмов осуществлялась по иерархической декомпозиционной методике с использованием аппарата графов влияния. Определение теоретических характеристик параллелизма выполнялось с помощью пакета *Mathematica® (Wolfram Research Inc.),* программная реализация осуществлялась при использовании стандарта передачи сообщений *MPI* (*Message Passing Interface*) на языке *C*++. Тестирование алгоритмов в данной работе производилось на базе *Argonne National Library MPICH-1.2.5,* одной из наиболее известных реализаций стандарта MPI для OS Windows. Все временные характеристики алгоритмов определялись с использованием функции *MPI_Barrier.* Для сокращения времени выполнения коллективных операций обмена использовались алгоритмы покоординатной маршрутизации.

На основе теоретического анализа вычислительной сложности параллельных алгоритмов с контролем локальной погрешности и численного эксперимента для тестовых задач можно сделать следующие выводы:

– в общем случае последовательные неявные методы Рунге-Кутты имеют достаточно большую вычислительную сложность по сравнению со своими явными аналогами;

– для высокоточных приложений ($10^{-15} - 10^{-20}$) и для сложных правых частей доминируют схемы локальной экстраполяции Ричардсона;

– для решения линейных однородных СОДУ с постоянными коэффициентами применение параллельных алгоритмов вложенных схем экспоненциального метода дает значительное сокращение накладных расходов.

Литература

1.Хайрер Э., Нёрсетт С., Ваннер Г. Решение обыкновенных дифференциальных уравнений. Нежесткие задачи: Пер. с англ. – М.: Мир, 1990. – 512с.

2.Арушунян О.Б., Залеткин С.Ф., Калиткин Н.Н. Тесты для вычислительного практикума по обыкновенным дифференциальным уравнениям. // Вычислительные методы и программирование, 2002, т.3, с.11-19.

Получено 01.06.07

УДК 519.6

МЕТОД *CR* – ФАКТОРИЗАЦИИ МАТРИЦ

С.Е. Саух

Институт проблем моделирования в энергетике им. Г.Е. Пухова НАН Украины

Запропоновано метод стовпцево-рядкової CR – факторизації матриць, який відрізняється від методу LU – факторизації адаптивністю до множини динамічно обираємих провідних елементів, що дозволяє відмовитися від виконання дій з упорядкування рядків та стовпчиків та суттєво пришвидшити розв'язок систем алгебраїчних рівнянь великої розмірності.

Одним из наиболее часто используемых вариантов метода Гаусса является метод треугольного разложения или LU – факторизации [1, 2]. В этом случае матрица $\boldsymbol{A}$ представляется в виде произведения нижнетреугольной матрицы $\boldsymbol{L}$ и верхнетреугольной матрицы $\boldsymbol{U}$. Необходимыми условиями осуществимости LU – факторизации являются невырожденность матрицы $\boldsymbol{A}$ и неравенство нулю элементов u_{kk}. Осуществимости треугольного разложения невырожденной матрицы $\boldsymbol{A}$ всегда можно добиться переупорядочив ее строки и столбцы так, чтобы выполнялись условия $u_{kk} \neq 0$. Для этого используют матрицы перестановок $\boldsymbol{P}$ и $\boldsymbol{Q}$ такие, что $\boldsymbol{P} \cdot \boldsymbol{A} \cdot \boldsymbol{Q} = \boldsymbol{L} \cdot \boldsymbol{U}$. Особенности реализации метода Гаусса в условиях конечно-разрядных вычислений требуют такого выбора ведущих элементов a_{ij}, при котором ограничивается рост абсолютных значений элементов матриц $\boldsymbol{L}$ и $\boldsymbol{U}$ [3-8]. В случае разреженных матриц большой размерности во внимание принимается влияние перестановок на уровень заполненности ненулевыми элементами матриц $\boldsymbol{L}$ и $\boldsymbol{U}$, который может существенно превысить уровень заполненности ненулевыми элементами исходной матрицы $\boldsymbol{A}$. Поэтому выбор ведущих элементов основывается на сочетании требований как обеспечения численной устойчивости метода Гаусса, так и минимизации уровеня заполненности ненулевыми элементами матриц $\boldsymbol{L}$ и $\boldsymbol{U}$. В соответствии с этими требованиями в современных программных средствах реализуются различные стратегии выбора ведущих элементов [3-6].

Хранение ненулевых элементов разреженных матриц большой размерности в строчных или связанных списках существенно затрудняет перестановки строк и столбцов [3-6]. Заметим, что

реализация перестановок не связана с выполнением фактических пересылок значений ненулевых элементов в памяти вычислительного устройства, а связана с выполнением операций поиска и изменения значений строчных и столбцовых индексов. Кроме того, в общем случае списки столбцовых индексов в строках и списки строчных индексов в столбцах подлежащих перестановкам не совпадют по длине, что усложняет реализацию изменений их индексов. Наши исследования показывают соизмеримость объемов операций необходимых для реализации перестановок и осуществления собственно $LU-$факторизации матриц. Мы предлагаем новый метод столбцово-строчной факторизации или, кратко, $CR-$факторизации, который принципиально отличается от метода $LU-$факторизации адаптивностью к выбору ведущих элементов, что позволяет вовсе отказаться от перестановок строк и столбцов в процессе формирования факторных матриц и ускорить процесс их вычисления.

Метод CR – факторизации матриц. Базовым соотношением метода является выражение

$$\boldsymbol{A} = \boldsymbol{C}_j \boldsymbol{R}_i + \boldsymbol{A}\langle(i, j)\rangle, \tag{1}$$

в котором исходная $n \times n$ матрица $\boldsymbol{A}$ представляется суммой произведения $n \times 1$ вектор-столбца $\boldsymbol{C}_j$ на $1 \times n$ вектор-строку $\boldsymbol{R}_i$ и $n \times n$ матрицы $\boldsymbol{A}\langle(i, j)\rangle$.

Если элемент a_{ij} матрицы $\boldsymbol{A}$ не равен нулю, то, полагая равными нулю элементы $i-$й строки и $j-$го столбца матрицы $\boldsymbol{A}\langle(i, j)\rangle$, из соотношения (1) с точностью до некоторого множителя можно определить все элементы столбца $\boldsymbol{C}_j$ и строки $\boldsymbol{R}_i$. Неоднозначность «с точностью до некоторого множителя» в определении элементов столбца $\boldsymbol{C}_j$ и строки $\boldsymbol{R}_i$ обусловлена невозможностью найти из одного уравнения вида $c_{ij} \cdot r_{ij} = a_{ij}$ две искомые величины c_{ij} и r_{ij}.

Для устранения неоднозначности в определении величин c_{ij} и r_{ij} следует принять одно из возможных допущений, например, вида

$$c_{ij} = 1, \tag{2}$$

$$r_{ij} = 1, \tag{3}$$

$$\left|c_{ij}\right| = \left|r_{ij}\right|. \tag{4}$$

Заметим, что допущения (2) и (3) аналогичны допущениям принимаемым в отношении диагональных элементов одной из

факторных матриц $\boldsymbol{L}$ и $\boldsymbol{U}$ в методе $LU-$факторизации. Допущение вида (4) также имеет место в специальном варианте метода $LU-$факторизации симметричных положительно определенных матриц известного как метод LL^T – факторизации Холесского или метод квадратного корня [7, 8].

Соотношение (1) определяет частичную одношаговую столбцово-строчную факторизацию матрицы $\boldsymbol{A}$ относительно ведущего элемента a_{ij}. Здесь столбец $\boldsymbol{C}_j$ и строка $\boldsymbol{R}_i$ являются множителями или факторами, а матрица $\boldsymbol{A}\langle(i,j)\rangle$ является остаточной матрицей, элементы которой, кроме тех, что расположены в $i-$й строке и $j-$м столбце, образуют активную $(n-1)\times(n-1)$ подматрицу.

Анализируя структуру матрицы $\boldsymbol{A}\langle(i,j)\rangle$ в соотношении (1), замечаем, что найденные элементы столбца $\boldsymbol{C}_j$ и строки $\boldsymbol{R}_i$ естественно расположить в матрице $\boldsymbol{A}\langle(i,j)\rangle$ на месте нулевых элементов ее $i-$й строки и $j-$го столбца. Такое совмещение элементов столбца $\boldsymbol{C}_j$ и строки $\boldsymbol{R}_i$ с элементами матрицы $\boldsymbol{A}\langle(i,j)\rangle$ приводит к формированию совмещенной матрицы вида

$$\boldsymbol{A}_{CR}\langle(i,j)\rangle = \boldsymbol{A}\langle(i,j)\rangle \oplus (\boldsymbol{C}_j, \boldsymbol{R}_i), \tag{5}$$

где символ $\oplus$ обозначает операцию совмещения.

Таким образом соотношения (1) – (5) определяют первый шаг столбцово-строчной факторизации матрицы $\boldsymbol{A}$ относительно ведущего элемента a_{ij} без перестановок строк и столбцов. Следующий аналогичный шаг факторизации осуществляется в отношении ведущего элемента, выбираемого в активной части матрицы $\boldsymbol{A}\langle(i,j)\rangle$, совпадающей с аналогичной частью матрицы $\boldsymbol{A}_{CR}\langle(i,j)\rangle$.

Если в качестве ведущего выбран элемент $a_{pq} - c_{pi}r_{jq} \neq 0$ находящийся на пересечении $p-$й строки и $q-$го столбца матрицы $\boldsymbol{A}\langle(i,j)\rangle$, то, приняв во внимание тождество (1), можем записать

$$\boldsymbol{A}\langle(i,j)\rangle = \boldsymbol{C}_q\boldsymbol{R}_p + \boldsymbol{A}\langle(i,j),(p,q)\rangle, \tag{6}$$

где $n\times 1$ вектор-столбц $\boldsymbol{C}_q$ и $1\times n$ вектор-строка $\boldsymbol{R}_p$ содержат по нулевому элементу $c_{iq}=0$ и $r_{pj}=0$, а $n\times n$ матрица $\boldsymbol{A}\langle(i,j),(p,q)\rangle$ имеет нулевыми две строки и два столбца с номерами i, p и j, q соотвественно.

После подстановки (6) в (1) получаем

$$\boldsymbol{A} = \boldsymbol{C}_j \boldsymbol{R}_i + \boldsymbol{C}_q \boldsymbol{R}_p + \boldsymbol{A}\langle (i,j),(p,q)\rangle \tag{7}$$

или в совмещенной форме

$$\boldsymbol{A}_{CR}\langle (i,j),(p,q)\rangle = \boldsymbol{A}\langle (i,j),(p,q)\rangle \oplus (\boldsymbol{C}_j, \boldsymbol{R}_i, \boldsymbol{C}_q, \boldsymbol{R}_p). \tag{8}$$

Основываясь на выражениях (1) и (6) определяющих частичную одно- и двухшаговую солбцово-строчную факторизацию, получаем формулу полной $CR-$факторизации невырожденной $n \times n$ матрицы $\boldsymbol{A}$ для случая, когда координаты (i, j) динамически выбираемых в процессе факторизации ведущих элементов образуют некоторое упорядоченное множество $\boldsymbol{P}$:

$$\boldsymbol{A} = \sum_{(i,j)\in P} \boldsymbol{C}_j \cdot \boldsymbol{R}_i \,. \tag{9}$$

Здесь суммирование осуществляется в порядке следования элементов в множестве $\boldsymbol{P}$. Заметим, что в отличие от (1) и (7) правая часть этого выражения не содержит матричных слагаемых вида $\boldsymbol{A}\langle \boldsymbol{P}\rangle$, поскольку в процессе их формирования за n шагов устанавливались равными нулю значения элементов тех n строк и n столбцов, на пересечении которых выбирались ведущие элементы. Именно на этих местах размещались вектор-столбцы $\boldsymbol{C}_j$ и вектор-строки $\boldsymbol{R}_i$ формируя совмещенную матрицу $\boldsymbol{A}_{CR}\langle \boldsymbol{P}\rangle$. Поэтому $\boldsymbol{A}\langle \boldsymbol{P}\rangle \equiv 0$.

Очевидно выражение (9) может быть представлено в матричном виде

$$\boldsymbol{A} = \boldsymbol{C}\langle \boldsymbol{P}\rangle \cdot \boldsymbol{R}\langle \boldsymbol{P}\rangle, \tag{10}$$

где матрицы $\boldsymbol{C}\langle \boldsymbol{P}\rangle$ и $\boldsymbol{R}\langle \boldsymbol{P}\rangle$ составлены соответственно из вектор-столбцов $\boldsymbol{C}_j$ и вектор-строк $\boldsymbol{R}_i$, расположенных в том порядке, в котором их индексы встречаются в множестве $\boldsymbol{P}$. Однако, порядок размещения вектор-столбцов $\boldsymbol{C}_j$ и вектор-строк $\boldsymbol{R}_i$ в совмещенной матрице вида

$$\boldsymbol{A}_{CR}\langle \boldsymbol{P}\rangle = \boldsymbol{C}_{CR} \oplus \boldsymbol{R}_{CR}. \tag{11}$$

соответствует номерам их индексов j и i, т.е. вместо матриц $\boldsymbol{C}\langle \boldsymbol{P}\rangle$ и $\boldsymbol{R}\langle \boldsymbol{P}\rangle$ в матрице $\boldsymbol{A}_{CR}\langle \boldsymbol{P}\rangle$ фактически совмещаются две матрицы $\boldsymbol{C}_{CR}\langle \boldsymbol{P}\rangle$ и $\boldsymbol{R}_{CR}\langle \boldsymbol{P}\rangle$.

Матрицы $\boldsymbol{C}\langle \boldsymbol{P}\rangle$ и $\boldsymbol{R}\langle \boldsymbol{P}\rangle$ в общем случае не приобретают треугольных форм. Только, когда в процессе $CR-$факторизации ведущие элементы выбираются последовательно в координатах $\boldsymbol{P}\langle (1,1),(2,2),\ldots,(n,n)\rangle$, тогда формируемые вектор-столбцы $\boldsymbol{C}_j$ и вектор-строки $\boldsymbol{R}_i$ образуют нижнетреугольную и верхнетреугольную

матрицы $\boldsymbol{C}\langle\boldsymbol{P}\rangle = \boldsymbol{L}$ и $\boldsymbol{R}\langle\boldsymbol{P}\rangle = \boldsymbol{U}$, которые располагаются в совмещенной матрице $\boldsymbol{A}_{CR}\langle\boldsymbol{P}\rangle$ так, что $\boldsymbol{C}_{CR}\langle\boldsymbol{P}\rangle = \boldsymbol{C}\langle\boldsymbol{P}\rangle = \boldsymbol{L}$ и $\boldsymbol{R}_{CR}\langle\boldsymbol{P}\rangle = \boldsymbol{R}\langle\boldsymbol{P}\rangle = \boldsymbol{U}$.

Основная структурная особенность матриц $\boldsymbol{C}\langle\boldsymbol{P}\rangle$ и $\boldsymbol{R}\langle\boldsymbol{P}\rangle$ общего вида заключается в том, что они могут быть в принципе преобразованы к треугольному виду путем перестановок строк и столбцов.

Результаты экспериментальных исследований. Очевидно выполняемые в процессе перестановок строк и столбцов операции выборки-сравнения индексов строчных и столбцовых списков ненулевых матричных элементов не могут быть непосредственно соотнесены с арифметическими операциями выполняемыми в процессе факторизации над ненулевыми элементами. Поэтому количество операций выборки-сравнения следует оценивать экспериментально по затратам времени на выполнение таких операций и оценивать их удельный вес в общих затратах времени на факторизацию матриц. Описание результатов и условий проведения экспериментов представлены ниже.

Вычисления выполнялись на компьютере Intel P4 (Chipset Intel 865 PE, FSB 800 MHz, CPU 3.0GHz with HT, Dual Channel Memory 1 GB: 2x512MB, DDR400), работающем под управлением операционной системы Microsoft Windows XP.

В основу экспериментальных исследований был положен программный код, написанный автором на языке C++ в среде Microsoft Visual Studio.Net. В нем реализованы последовательно три алгоритма решения системы линейных алгебраических уравнений вида $\boldsymbol{A}\cdot\boldsymbol{X} = \boldsymbol{B}$ с вектором правой части $\boldsymbol{B} = \boldsymbol{A}\cdot\boldsymbol{1}$, где $\boldsymbol{1}-$ единичный вектор. Первый алгоритм реализует метод $CR-$факторизации матрицы $\boldsymbol{A}$ с улучшенной обобщенной стратегией Марковица выбора ведущих элементов, описанной в работе [4]. Второй алгоритм реализует метод $CR-$факторизации той же матрицы $\boldsymbol{A}$, использующий уже сформированное первым алгоритмом множество $\boldsymbol{P}$ последовательности выбора и координат ведущих элементов. Третий алгоритм отличается от второго только тем, что он реализует метод $LU-$факторизации. Отметим, что в обоих реализациях метода $CR-$факторизации матрицы $\boldsymbol{A}$ используется предположение (3).

Исполнение написанного программного кода позволяет оценить суммарные затраты времени вычислений на поиск решения $\boldsymbol{X}$ по заданному вектору $\boldsymbol{B}$ и дифференцировать затраты времени вычислений на $CR-$ и $LU-$факторизацию. При этом обеспечивается

выполнение равенства $nnz(\boldsymbol{C}_{CR} \oplus \boldsymbol{R}_{CR}) = nnz(\boldsymbol{L} \oplus \boldsymbol{U})$, т.е. совпадение количеств ненулевых элементов в факторных матрицах $\boldsymbol{C}$, $\boldsymbol{R}$ и $\boldsymbol{L}$, $\boldsymbol{U}$ полученных методами $CR-$ и $LU-$факторизации.

Для обеспечения корректности экспериментальных исследований было максимально ослаблено влияние процедур управления расположением ненулевых матричных элементов в рабочих массивах. Прежде всего размеры массивов ненулевых элементов, их строчных и столбцовых индексов были выбраны равными $45 \cdot 10^6$, т.е. настолько большими, чтобы не возникало потребности инициировать работу специальных процедур «уборки мусора» в них [4, 5]. К тому же во всех тестах применялась улучшенная обобщенная стратегия Марковица, где поиск ведущего элемента ограничивался множеством S_p ненулевых элементов, расположенных в p строках активной подматрицы с минимальным количеством таких элеменов, а ведущим выбирался элемент a_{ij} с наименьшей ценой Марковица, но тот, абсолютное значение которого не более чем в τ раз меньше максимального по абсолютной величине элемента $a_{\boldsymbol{max}} \in S_p$.

Таблица 1. Тестовые матрицы, их размерность n, количество ненулевых элементов в исходных матрицах $nnz(\boldsymbol{A})$ и в совмещенных факторных матрицах $nnz(\boldsymbol{C} \oplus \boldsymbol{R})$, ошибки ε и время решения $T_{\boldsymbol{CR}}$ тестовых систем.

название	n	$nnz(\boldsymbol{A})$	$nnz(\boldsymbol{C} \oplus \boldsymbol{R})$	ε	$T_{\boldsymbol{CR}}, c$
ASIC_680k	682862	3871773	8220972	4.57E-08	1024.531
ASIC_680ks	682712	2329176	6170719	9.51E-09	47.953
ASIC_320ks	321671	1827807	5082048	2.89E-12	54.875
scircuit	170998	958936	2408645	8.39E-11	2.328
circuit_4	80209	307604	707803	1.92E-09	1.359
bayer01	57735	277774	7283567	4.33E-08	40.453
mark3jac060	27449	170695	28775871	3.42E-11	2345.125
ex35	19716	228208	2474829	6.82E-08	7.360
poisson3Da	13514	352762	10697886	1.10E-13	162.281
circuit_3	12127	48137	76708	2.07E-12	0.141
cry10000	10000	49699	501518	4.04E-06	0.828
gemat11	4929	33185	77616	2.21E-13	0.047
lns_3937	3937	25407	568884	1.60E-12	1.719
psmigr_1	3140	543162	6534812	9.40E-13	122.531
orani678	2529	90158	336551	3.07E-15	0.469
adder_trans_01	1814	14579	20541	2.11E-16	0.016
orsirr_1	1030	6858	57892	1.42E-13	0.063
orsirr_2	886	5970	45146	1.37E-13	0.031

Вычислительные эксперименты выполнялись с матрицами, взятыми с веб-сайтов [9, 10]. Общие характеристики тестовых матриц и результаты экспериментирования с ними представлены в таблицах 1-2, где

$$\varepsilon = \sqrt{\frac{\sum_{z=1}^{n}(x_z - 1)^2}{n}}.$$

В зависимости от тестируемой матрицы параметры p и τ устанавливались равными: в тесте ASIC_680k – $p = 100$ и $\tau = 0.0005$, в тесте mark3jac060 – $p = n$ и $\tau = 0.001$. Во всех остальных тестах значения параметров были $p = 1$ и $\tau = 1$.

Таблица 2. Затраты времени на факторизацию (T_{CR-F}, T_{LU-F}) и решение (T_{CR-S}, T_{LU-S}) тестовых систем уравнений методами $CR-$ и $LU-$факторизации; относительное замедление процесса $LU-$факторизации матриц $\delta T_F = \left(\frac{T_{LU-F}}{T_{CR-F}} - 1\right) \cdot 100\%$.

название	T_{CR-F}, c	T_{LU-F}, c	$\delta T_F, \%$	T_{CR-S}, c	T_{LU-S}, c
ASIC_680k	1004.328	1539.047	53.2	0.172	0.156
ASIC_680ks	46.656	49.204	5.5	0.188	0.140
ASIC_320ks	54.875	59.921	9.2	0.171	0.141
scircuit	2.000	3.907	95.4	0.125	0.078
circuit_4	1.172	1.562	33.3	0.016	0.032
bayer01	39.062	51.563	32.0	0.094	0.094
mark3jac060	903.953	1106.750	22.4	0.297	0.281
ex35	7.016	10.578	50.8	0.047	0.031
poisson3Da	159.109	204.328	28.4	0.110	0.110
circuit_3	0.093	0.109	17.2	0.000	0.016
cry10000	0.781	1.171	49.9	0.000	0.016
gemat11	0.031	0.047	51.6	0.000	0.000
lns_3937	1.625	2.313	42.3	0.015	0.000
psmigr_1	120.437	147.562	22.5	0.063	0.063
orani678	0.437	0.656	50.1	0.016	0.000
adder_trans_01	0.015	0.016	6.7	0.000	0.000
orsirr_1	0.047	0.078	66.0	0.000	0.000
orsirr_2	0.047	0.063	34.0	0.000	0.000

Описанные условия проведения экспериментов позволили избежать инициализации процедур «уборки мусора» в массивах. Однако, в алгоритме реализующем метод $LU-$факторизации мы

применили одну из известных процедур перестановки строк и столбцов, минимизирующую объем требуемых для этого операций выборки-сравнения [11]. Такая процедура приводит не только к изменению значений индексов переставляемых строк и столбцов в массивах строчных и столбцовых индексов, но и сопровождается перестановками отдельных элементов в тех же массивах, а таже в массиве ненулевых элементов. В результате перестановок элементов в массивах их размещение на текущем шаге реализации метода $LU-$факторизации становится отличным от того размещения, которое можно наблюдать в случае реализации метода $CR-$факторизации. Такие различия в размещении элементов усиливаются необходимостью размещения вновь возникающих ненулевых элементов, что порождает различные динамические процессы переразмещения элементов строк и столбцов. Таким образом нам не удалось достичь полного равенста условий проведения экспериментов с программными реализациями методов $CR-$ и $LU-$факторизации. Однако мы минимизировали расхождения в условиях настолько насколько это было возможно.

Из полученных результатов следуют такие выводы:

1. Затраты времени на осуществление перестановок строк и столбцов в активных подматрицах оказываются сопоставимыми с затратами времени на факторизацию матриц. Реализация перестановок в методе $LU-$факторизации приводит к существенному замедлению вычислительного процесса в среднем на величину $\delta T_F = 37.3\%$ для данного множества тестов, что следует из таблицы 2, где представлены значения $\delta T_F = \left(\frac{T_{LU-F}}{T_{CR-F}} - 1 \right) \cdot 100\%$.
2. Затраты времени на вычисление решений систем уравнений вида $\boldsymbol{C}\langle \boldsymbol{P} \rangle \cdot \boldsymbol{R}\langle \boldsymbol{P} \rangle \cdot \boldsymbol{X} = \boldsymbol{B}$, полученных методом $CR-$факторизации, больше затрат времени на вычисление решений систем уравнений вида $\boldsymbol{L} \cdot \boldsymbol{U} \cdot \boldsymbol{X} = \boldsymbol{B}$, формируемых методом $LU-$факторизации. Увеличение вычислительных затрат обусловлено особенностями алгоритмов решения систем уравнений вида $\boldsymbol{C}\langle \boldsymbol{P} \rangle \cdot \boldsymbol{V} = \boldsymbol{B}$ и $\boldsymbol{R}\langle \boldsymbol{P} \rangle \cdot \boldsymbol{X} = \boldsymbol{V}$ с нетреугольными матрицами $\boldsymbol{C}\langle \boldsymbol{P} \rangle$ и $\boldsymbol{R}\langle \boldsymbol{P} \rangle$. Однако, в целом суммарное время решения исходных систем уравнений составляет $T_F + T_S$ и является существенно большим в случае использования метода $LU-$факторизации.

Заключение. В отличие от метода $LU-$ факторизации матриц с выбором ведущих элементов, где в общем случае обязательно

выполняются операции перестановки строк и столбцов, метод CR – факторизации матриц не требует выполнения таких операций, что ускоряет процесс факторизации. Ускорение оказывается особенно существенным в случае затрудненного доступа к матричным элементам, который обусловлен использованием специальных форматов размещения в массивах ненулевых элементов разреженных матриц.

Очевидно метод CR – факторизации матриц может использоваться не только, как эффективный прямой метод решения больших систем уравнений, но и как метод быстрого построения предобусловливателей для итерационного решения систем [12-14]. В этом случае построение предобусловливателей посредством неполной CR – факторизации матриц может осуществляться без какого-либо перемещения ненулевых элементов в массивах. Таким образом предлагаемый метод CR – факторизации матриц может быть положен в основу алгоритмов и программ ускоренного решения систем линейных алгебраических уравнений большой размерности.

1. www.cise.ufl.edu/research/sparse/codes/
2. www.srcc.msu.su/num_anal/lib_na/cat/cat552.htm
3. Писанецки С. Технология разреженных матриц. – М.: Мир, – 1988. – 410 с.
4. Эстербю О., Златев З. Прямые методы для разреженных матриц. – М.: Мир, – 1987. – 120 с.
5. Икрамов Х.Д. Вычислительные методы линейной алгебры. – Знание, 1989. – 48 с.
6. Davis T.A. Direct Methods for Sparse Linear Systems. – Philadelphia.: SIAM, – 2006. – 226 p.
7. Самарский А.А., Гулин А.В. Численные методы. – М.: Наука, – 1989. – 432 с.
8. Джордж А., Лю Дж. Численное решение больших разреженных систем уравнений. – М.: Мир, – 1984. – 334 с.
9. www.cise.ufl.edu/research/sparse/matrices/index.html
10. http://math.nist.gov/MatrixMarket/matrices.html
11. www.netlib.org/y12m/index.html
12. Saad Y. Iterative Methods for Sparse Linear Systems, University of Minnesota, Department of Computer Science and Engineering, Minneapolis, MN, 2000. – 448 p.
13. Ортега Дж. Введение в параллельные и векторные методы решения линейных систем. – М.: Мир, – 1991. – 386 с.
14. Saukh S.Ye. Incomplete Cholesky factorization in fixed memory // Parallel Processing and Applied Mathematics: 5th International Conference, PPAM 2003, Czestochowa, Poland, September 7-10, 2003. Revised Papers. – Springer-Verlag Heidelberg: Lecture Notes in Computer Science. –Volume 3019 / 2004, pp. 1042 - 1051.

Материал поступил в редколлегию 10.05.2007

УДК 681.5

ПАРАЛЛЕЛЬНЫЕ АЛГОРИТМЫ ЧИСЛЕННОГО РЕШЕНИЯ СИСТЕМ ЛИНЕЙНЫХ ОБЫКНОВЕННЫХ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ В СИСТЕМАХ С РАСПРЕДЕЛЁННОЙ ПАМЯТЬЮ С ТОПОЛОГИЕЙ РЕШЁТКА-ТОР

Фельдман Л.П., Назарова И.А., Кожухов А.Е.
Донецкий Национальный Технический Университет

Рассматриваются параллельные алгоритмы численного решения задачи Коши для систем обыкновенных линейных дифференциальных уравнений в системах с распределённой памятью и топологией типа решётка–тор. Получены оценки времени выполнения, ускорения и эффективности распараллеливания метода Рунге-Кутта с учётом коммуникационных затрат на обмены данными. Показаны зависимости характеристик параллельных алгоритмов от размерности решётки, сложности исходной задачи и числа шагов метода Рунге-Кутта.

В настоящее время имеют место бурное развитие науки и, особенно, рост сложности решаемых человеком практических задач. Во многих случаях процесс решения задач такого типа сопряжен с большими материальными и вычислительными затратами, если вообще возможен. В таких случаях оправдано и необходимо применение наиболее эффективных вычислительных методов и их эффективная реализация в высокопроизводительных вычислительных системах [1].

Многие сложные задачи (наиболее часто – моделирование сложных вычислительных систем) сводятся к обыкновенным дифференциальным уравнениям и их системам. Огромный практический интерес связан с задачами моделирования сложных динамических систем с сосредоточенными параметрами.

Решение вышеописанных задач моделирования имеет особый интерес в случае систем обыкновенных линейных дифференциальных уравнений [2]. Этот интерес обусловлен возможностью сведения алгоритма решения к последовательности матрично–векторных арифметических операций, которые могут быть эффективно распараллелены в системах с распределённой памятью.

Численное решение задачи Коши вида (1) для системы обыкновенных линейных дифференциальных уравнений [2]:

$$\frac{dx}{dt} = A \cdot x + f(t)$$
$$x(t_0) = x^{(0)} = (x_1^0, x_2^0, \ldots, x_m^0)^T \qquad (1)$$

Решение задачи вида (1) может быть получено по правилу Рунге-Кутта и определяется формулой (2).

$$x^{(n+1)} = x^{(n)} + \frac{1}{6} \cdot (k_1^{(n)} + 2k_2^{(n)} + 2k_3^{(n)} + k_4^{(n)})$$
$$k_1^{(n)} = \tau(A \cdot x^{(n)} + f^{(n)})$$
$$k_2^{(n)} = \tau(A \cdot (x^{(n)} + 0.5k_1^{(n)}) + f^{(n+0.5)}) \qquad (2)$$
$$k_3^{(n)} = \tau(A \cdot (x^{(n)} + 0.5k_2^{(n)}) + f^{(n+0.5)})$$
$$k_4^{(n)} = \tau(A \cdot (x^{(n)} + 0.5k_3^{(n)}) + f^{(n+1)})$$

Формула (2) определяет итерационный процесс, который сводится к операциям матрично-векторной арифметики. На каждом шаге процесса вычислений происходит расчёт промежуточных векторов k_i, порядок вычисления которых строго последователен. Как результат, параллелизм может быть реализован только при расчёте каждого из промежуточных векторов в процессе вычисления результатов матрично–векторных операций.

Альтернативный метод решения задачи Коши по правилу Рунге-Кутта – определение полного оператора (матрицы) перехода. Эта матрица рассчитывается предварительно и единожды (до выполнения итераций метода) по формуле (3). Таким образом, весь итерационный процесс сводится к серии последовательных умножений матрицы на вектор [2].

$$x^{(n+1)} = A' \cdot x^{(n)}$$
$$A' = (E + \tau A(E + \frac{\tau A}{2}(E + \frac{\tau A}{3}(E + \frac{\tau A}{4})))) \qquad (3)$$

Преимущество подхода с расчётом матрицы перехода проявляется при большом количестве итераций метода Рунге–Кутта, вычислительная сложность последовательного алгоритма характеризуется формулой (4). Соответственно, для малого количества итераций лучшие показатели будет демонстрировать подход с расчётом промежуточных величин, сложность последовательного алгоритма которого представлена формулой (5).

В формулах величина m – размерность исходной задачи, N – число шагов метода Рунге–Кутта, τ – длительность модельного такта времени.

$$T_{RKStepSolveS}[m,N] = N \cdot \left(15 \cdot m\tau + 8 \cdot m^2\tau\right) \quad (4)$$

$$T_{RKTransMatrixS}[m,N] = 4 \cdot m\tau + 4 \cdot m^2\tau + 6 \cdot m^3\tau + 2N \cdot m^2\tau \quad (5)$$

В распределённых параллельных вычислительных системах наиболее трудоёмкими с вычислительной и коммуникационной точек зрения являются операции вычисления матричного и матрично–векторного произведений. Наиболее подходящими в плане коммуникационных затрат (при одинаковых вычислительных затратах) для таких операций являются топологии решётка (решётка-тор и её модификации) и гиперкуб.

Для топологии типа решётка вычисление матрично–векторных арифметических операций можно свести к выполнению операций аналогичного типа над блоками матриц. Сложение матриц сводится к однократному матричному сложению соответствующих блоков.

Умножение матриц является более сложным и может быть эффективно выполнено с помощью ряда алгоритмов семейств Кэннона и Фокса [3]. Алгоритмы семейства Фокса не меняют отображения блоков матриц-операндов и матрицы-результата на узлы вычислительной решётки. Особенностью алгоритмов Кэннона является смена отображения блоков одной из матриц–операндов или матрицы–результата на узлы вычислительной решётки. Каждый из алгоритмов Кэннона сохраняет отображение блоков соответствующей матрицы на узлы вычислительной решётки [4].

Наиболее эффективным среди рассматриваемых алгоритмов вычисления матричного произведения является алгоритм Кэннона, сохраняющий отображение блоков матрицы–результата.

Умножение матрицы на вектор может быть сведено к умножению матрицы на матрицу. При выполнении определённых условий эта операция может быть осуществлена по модифицированному алгоритму вычисления произведения матриц.

В системах с распределённой памятью время передачи данных пропорционально их объёму (справедливо для больших объёмов передаваемых данных) [1]. При уменьшении объёма передаваемых данных происходит снижение времени выполнения параллельного алгоритма. Таким образом, параллельный алгоритм умножения матрицы на вектор в общем случае будет эффективнее умножения матрицы на матрицу. Такое соотношение времён выполнения

алгоритмов может нарушаться, если в алгоритме умножения матрицы на вектор будут передаваться только блоки матрицы.

Реализация метода Рунге–Кутта с вычислением промежуточных коэффициентов распараллеливается на этапах выполнения матрично–векторных операций. Реализация метода Рунге–Кутта с вычислением матрицы перехода распараллеливается аналогичным образом.

Пусть в дополнение к введённым ранее обозначениям будут использованы: k – размерность блока вектора, p – число процессоров, t_s – временные затраты подготовки сообщения к отправке (латентность), β - пропускная способность канала передачи данных, t_w – время передачи слова данных, γ – относительная доля служебных данных (которыми обрамляется передаваемое сообщение) по отношению к размеру передаваемого сообщения.

Величины времени выполнения параллельных алгоритмов с вычислением промежуточных коэффициентов и матрицы перехода определяются формулами (6) и (7) соответственно.

$$T_{RKStepSolve}[N,k] = N \cdot \left(12 \cdot k\tau + 4 \cdot \left[p \cdot (t_s + kt_w + \gamma kt_w) + 3 \cdot p \cdot k^2\tau + 4 \cdot (t_s + kt_w + \gamma kt_w \cdot \left\lfloor \frac{p}{2} \right\rfloor) \right] \right) \quad (6)$$

$$\begin{aligned} T_{RKTransMatrix}[N,k] = N \cdot & \left[p \cdot (t_s + kt_w + \gamma kt_w) + 3 \cdot p \cdot k^2\tau + 4 \cdot (t_s + kt_w + \gamma kt_w \cdot \left\lfloor \frac{p}{2} \right\rfloor) \right] + \\ & +4 \cdot k\tau + 4 \cdot k^2\tau + 3 \cdot \left(2 \cdot t_s + 2 \cdot k^2 t_w + p \cdot (t_s + k^2 t_w + \gamma k^2 t_w) + 2 \cdot p \cdot k^3\tau + 2 \cdot \gamma k^2 t_w \cdot \left\lfloor \frac{p}{2} \right\rfloor \right) \end{aligned} \quad (7)$$

Величины ускорений по сравнению с последовательными вариантами параллельных алгоритмов задаются формулами (8) и (9) соответственно.

$$S_m(N,m,p) \underset{p\to\infty}{\cong} \frac{m \cdot (15 + 8 \cdot m) \cdot \tau}{(4 \cdot t_s) \cdot p + 4 \cdot (4 \cdot t_s + mt_w \cdot (1+\gamma))} \quad (8)$$

$$S_m(N,m,p) \underset{p\to\infty}{\cong} \frac{2 \cdot m \cdot (2 + 3 \cdot m^2 + m \cdot (2+N)) \cdot \tau}{p \cdot (3+N) \cdot t_s + (6 + 4 \cdot N) \cdot t_s + mNt_w \cdot (1+\gamma)} \quad (9)$$

Величины эффективности использования процессоров параллельными алгоритмами описываются формулами (10) и (11) соответственно.

$$E_m(N,m,p) \underset{p\to\infty}{\cong} \frac{m \cdot (15 + 8 \cdot m) \cdot \tau}{(4 \cdot t_s) \cdot p^2 + 4 \cdot (4 \cdot t_s + mt_w \cdot (1+\gamma)) \cdot p} \quad (10)$$

$$E_m(N,m,p) \underset{p\to\infty}{\cong} \frac{2\cdot m\cdot\left(2+3\cdot m^2+m\cdot(2+N)\right)\cdot\tau}{p^2\cdot(3+N)\cdot t_s+p\cdot(6+4\cdot N)\cdot t_s+p\cdot mNt_w\cdot(1+\gamma)} \qquad (11)$$

Пусть алгоритм решения задачи Коши для системы линейных ОДУ с вычислением промежуточных данных по методу Рунге–Кутта будет обозначен через RKStepSolve. Тогда альтернативный алгоритм с предварительным вычислением матрицы перехода, реализующий метод Рунге–Кутта, будет обозначен через RKTransMatrix.

С учётом вышеприведённых формул могут быть сделаны сравнительные выводы по поводу параллельных алгоритмов решения систем линейных ОДУ:

1) для больших значений p преимущество алгоритма RKTransMatrix над алгоритмом RKStepSolve реализуется за счёт меньшего времени выполнения итерации метода Рунге–Кутта;

2) для относительно малых значений p время выполнения алгоритма RKTransMatrix будет хуже аналогичной величины для алгоритма RKStepSolve в связи с затратами на предварительное вычисление матрицы перехода;

3) для больших значений m преимущество алгоритма RKTransMatrix над алгоритмом RKStepSolve реализуется за счёт меньшего времени выполнения итерации метода Рунге–Кутта;

4) для относительно малых значений m время выполнения алгоритма RKTransMatrix будет хуже аналогичной величины для алгоритма RKStepSolve в связи с затратами на предварительное вычисление матрицы перехода;

5) для малого числа шагов метода Рунге–Кутта параллельный алгоритм RKTransMatrix будет показывать худшие показатели, чем параллельный алгоритм RKStepSolve;

6) алгоритм RKTransMatrix имеет меньшие временные затраты на выполнение итерации по сравнению с RKStepSolve.

Литература

1. Гергель В.П. Основы параллельных вычислений для многопроцессорных вычислительных систем. – Н.Новгород, ННГУ, 2001.
2. Фельдман Л.П. Параллельные алгоритмы численного решения систем линейных обыкновенных дифференциальных уравнений.
3. Gupta A., Kumar V. Scalability of Parallel Algorithms for Matrix Multiplication. Department of Computer Science, University of Minnesota Minneapolis, 1991.
4. Li J. A Poly-Algorithm for Parallel Dense Matrix Multiplication on Two-Dimensional Process Grid Topologies. Department of Computer Science, Mississippi State University, 1996.

Получено 01.06.07

УДК 681.3

МОДЕЛЬ ПРОИЗВОДИТЕЛЬНОСТИ РЕПЛИЦИРОВАННОЙ БАЗЫ ДАННЫХ 2rc

Афонов И.В., Ладыженский Ю.В.
Донецкий Национальный Технический Университет

В данном докладе рассматривается характеристика модели производительности распределённой и реплицированной базы данных (БД) 2rc. Данная модель позволяет получить характеристики производительности системы – среднее время отклика и пропускную способность системы. Модель позволяет анализировать производительность системы в зависимости, от многих факторов включая качество выбора данных для репликации и выбора узлов для размещения копий.

Репликация данных – механизм синхронизации содержимого нескольких копий базы данных. Репликация может быть синхронной или асинхронной. В большинстве реальных систем применяется асинхронная репликация, когда данные после изменения на одном из узлов, распространяются на другие узлы в течение некоторого времени, а не сразу после изменения. Репликация также может быть полной либо частичной. При полной репликации – все данные размещены во всех базах данных. При частичной – часть данных размещена в части баз данных, а к остальным, не реплицированным данным, доступ производится удалённо.

Математическая модель 2rc представляет собой комплекс моделей основных компонентов реплицированной базы данных. Критерием производительности данной модели является среднее время отклика и пропускная способность системы.

В таблице 1 представлены основные компоненты распределённой системы хранения данных (СХД) и методы их моделирования.

Таблица 1 – Компоненты модели 2rc и способы моделирования

Компонент	Метод моделирования
Узлы	Сеть из m M/G/1
Сеть	M/G/1
Репликация	Частичная, 2 измерения
Качество репликации	Выбор информации для репликации и узлов для хранения копий

Таблица 1(Продолжение) – Компоненты модели 2rc и способы моделирования

Доступ к данным	Locality (*b* % запросов могут быть выполнены локально)
Обработка транзакций	Более 2 типов транзакций
Разрешение конфликтов	Игнорируется

В модели используется стратегия управления обновлениям «primary copy». Данная стратегия подразумевает выполнения обновлений только на одной копии данных, которая считается главной, в то время как запросы на выборку производятся с наиболее удобных узлов. Обновления распространяются на ведомые копии асинхронно. К преимуществам данной стратегии можно отнести низкую вероятность возникновения конфликтов и относительно небольшую нагрузку на сеть при частых обновлениях. Недостатком данной стратегии является то, что клиент должен быть всегда подключен к главному серверу.

Отличительной характеристикой данной модели является введение в модель показателей качества выбора данных для реплицирования и выбора узлов для размещения копий. Выбор данных для репликации и решение где разместить копии оказывает огромное влияние на производительность системы в целом.

Показатель качества выбора данных для репликации оказывает большое влияние на выполнение запросов на обновление данных, в то время как показатель качества размещения копий оказывает большое влияние на обработку запросов на выборку.

Выбор данных для репликации, которые должны часто обновляться вызывает повышенную загрузку сети во время распространения изменений на ведомые копии базы дынных. Пример графика зависимости пропускной способности системы от показателя качества выбора данных для репликации показан на рисунке 1.

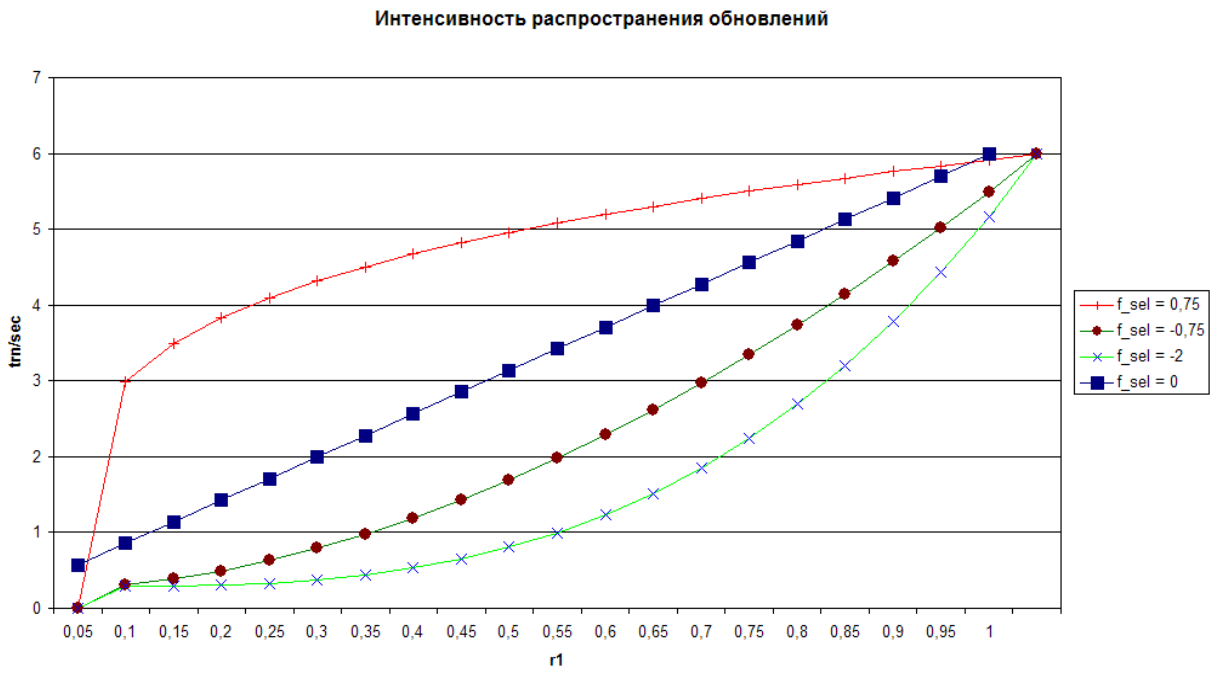


Рисунок 1 – График зависимости пропускной способности системы от показателя качества выбора репликации.

Неудачный выбор данных для репликации приводит к нелинейному и быстрому возрастанию нагрузки с увеличением доли реплицируемых данных. Удачный выбор данных не приводит к значительному увеличению нагрузки на систему, и функция возрастает медленнее линейной.

Кроме того, что выбор данных для репликации оказывает влияние на производительность обработки запросов на выборку, гораздо большее влияние на скорость их обработки оказывает размещение копий данных. Даже если все данные, к которым наиболее вероятен доступ, реплицированы, производительность всей системы не будет значительно увеличиваться, пока эти данные не будут доступны для запросов на локальных узлах.

Литература

[1] Jim Gray, Pat Helland, Patrick O'Neil, Dennis Shasha. *The Dangers of Replication and a Solution*. Microsoft, 2002.

[2] Matthias Nicola, Matthias Jarke. *Performance Modeling of Distributed and Replicated Databases.* Technical University of Aachen, 2000

Дата поступления в редакцию 16.05.2007.

УДК 681

ОЦЕНКА ЭФФЕКТИВНОСТИ ВЫСОКОПРОИЗВОДИТЕЛЬНЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ С ИСПОЛЬЗОВАНИЕМ АНАЛИТИЧЕСКИХ МЕТОДОВ

Т.В. Михайлова
Донецкий национальный технический университет

Предлагаются модифицированные методы анализа и синтеза высокопроизводительных вычислительных ресурсов различной топологии с помощью вероятностных моделей, позволяющие анализировать и проектировать более широкий класс высокопроизводительных параллельных вычислительных сред.

В настоящее время широкое распространение получили кластеры различной архитектуры. По совместному использованию дискового пространства их можно классифицировать следующим образом: с совместным использованием дискового пространства и без предоставления доступа к ресурсам [1].

Одним из способов исследования различных структур является использование непрерывных [2] или дискретных аналитических моделей [3]. Дискретные модели Маркова, отражающие модель вычислительной среды более точно [4], имеют большую размерность, что затрудняет иследования. Однако они эффективно распараллеливаются на параллельные вычислительные структуры [5].

Непрерывные модели менее трудоемкие, поэтому более широко применяются для исследований высокопроизводительных вычисительных систем. На основании методик [6,7] можно построить модели кластеров с совместным использованием дискового пространства и без предоставления доступа к ресурсам для их исследования.

Оптимальный коэффициент мультипрограммирования подбирается с использованием методики [8], в которой предлагается критерий сбалансированности, составляющие которого цена простоя оборудования и штраф за задержку выполнения запроса.

Для оптимизации состава и структуры вычислительных систем можно использовать методы [9], позволяющие определить структуру вычислительной среды минимальной стоимости при заданном времени отклика или, наоборот, с минимальным временем отклика заданной стоимости.

Алгоритм с использованием теоремы Джексона имеет комбинаторный порядок, а алгоритмы с использованием теоремы о среднем [9] – полиномиальный, что позволяет решать задачи, которые

вообще не решаются аналитическим методом на современных ЭВМ в течение реального времени.

Использование модифицированных вероятностных моделей и методов при проектировании, эксплуатации и оптимизации высокопроизводительных вычислительных систем позволяет вырабатывать рекомендации по их рациональному использованию.

1. Спортак М., Франк Ч., Паппас Ч. и др. Высокопроизводительные сети. Энциклопедия пользователя.-К.:"ДиаСофт", 1998.-432С.
2. Авен О. И. и др. Оценка качества и оптимизация вычислительных систем. – М.: Наука, 1982, 464С.
3. Михайлова Т.В., Фельдман Л.П. Параллельный алгоритм построения дискретной модели Маркова // Материалы Международной научно-технической конференции «Искусственный интеллект. Интеллектуаьные и многопроцесорные системы.», 25-30 сентября 2006г., Таганрог-Донецк-Минск, 2006.– С.. 275–279.
4. Михайлова Т.В. Оценка точности непрерывной и дискретной моделей Маркова // Научные труды Донецкого государственного технического университета. Серия «Информатика, кибернетика и вычислительная техника»(ИКВТ-2005), выпуск 93.- Донецк: ДонГТУ.- 2005. -С. 114-122.
5. Фельдман Л.П., Михайлова Т.В. Эффективность параллельного алгоритма построения дискретных марковских моделей // Материалы III Международной конференции «Параллельные вычисления и задачи управления» , 2-4 октября 2006г., Москва.-С.160-165.
6.Основы теории вычислительных систем/С.А.Майоров, Г.И.Новиков, Т.И.Алиев и др.М.: Высшая школа, 1978, 408С.
7. Михайлова Т.В., Фельдман Л.П. Оценка эффективности кластерных систем с использованием аналитических методов // Сборник трудов конференции "Моделирование 2006" (Simulation-2006), 16-18 мая 2006, г.Киев, 2006. – С.465-468.
8. Фельдман Л.П., Михайлова Т.В. Оценка эффективности кластерных систем с использованием моделей Маркова. //Известия ТРТУ. Тематический выпуск: Материалы Всероссийской научно-технической конференции с международным участием «Компьютерные технологии в инженерной и управленческой деятельности». – Таганрог: ТРТУ, 2002. – №2 (25). – С. 50–53.
9. Фельдман Л.П., Михайлова Т.В. Способы оптимизации состава и структуры высокопроизводительных вычислительных систем //Научные труды Донецкого государственного технического университета. Серия «Информатика, кибернетика и вычислительная техника»(ИКВТ-2001).- Донецк: ДонГТУ.- 2000. С. 80-85.

Получено 01.06.07

Удк 681.32

MODEL CASCADE FOR FLASH FLOOD PREDICTION

Kussul N.M., Kravchenko O.M.
Space Research Institute of NAS and NSA of Ukraine

Розглянуто питання створення каскаду гідрометеорологічних моделей для розв'язання задачі прогнозування паводків. Даний підхід розглядається як альтернативний до повного зв'язування моделей. Розглянуто питання паралельного виконання обчислень з використанням сучасної високопродуктивної техніки і підходу Grid до розподілення обчислень.

Introduction. Flooding is generally considered as devastating natural hazard. Ukraine is vulnerable to floods, in particular in the Carpathian region where it occurs almost every year. For last 10 years Ukrainian part of Carpathian was affected by two extreme flood events in 1998 and 2001. These events results in human dies and infrastructure losses. For instance during the floods in 2001 9 people were killed and 12000 citizens were evacuated, more than 1500 buildings were destroyed, and more than 30000 buildings were flooded. That is why, flood monitoring and forecasting is a task of great importance.

Modern flood forecasting and monitoring involves complex hydromeorological simulations. To model the system as a whole one has to manage several models from different application domains such as meteorology, hydrology, and hydraulics. Two general approaches exists to integrate several models: the first is full model coupling that require substantial work and even complete redesign of existing models, the second is model cascading approach, which is more straightforward. In latter output from one model is used as input to the next model in cascade. In this paper we explore both approaches using coupled models when they are available and cascading them in other cases.

To perform real-world simulations using computationally complex hydrometeorological models require using High Performance Computing (HPC). Therefore special attention is devoted to parallel simulations using HPC clusters and Grid approach to distributed computations.

Simulation cascade. Most flood events are driven by rapid development of runoff caused by intense precipitation or/and snowmelt. Therefore quantitative estimation of precipitation and snowmelt is crucial

point in flood forecasting. To assess these phenomena hydrologists traditionally rely on conventional in-situ observations that were later expanded by meteorological radars and satellite-based precipitation estimates. The use of meteorological models was very limited due to their coarse resolution that allowed it to be used only for macroscale hydrological studies. However, with the progress in regional weather modeling and rapid increase of computation power it is become possible to utilize such models as additional source of precipitation.

To predict flooding parameters such as rivers stage/discharge and extents of flooded areas we use cascade of simulation models: regional numerical weather prediction (NWP) model, hydrological model and hydraulic model (Fig. 1). This approach was modeled after successful work of *Hluchy et. al.* [1] considering Slovakian watersheds.

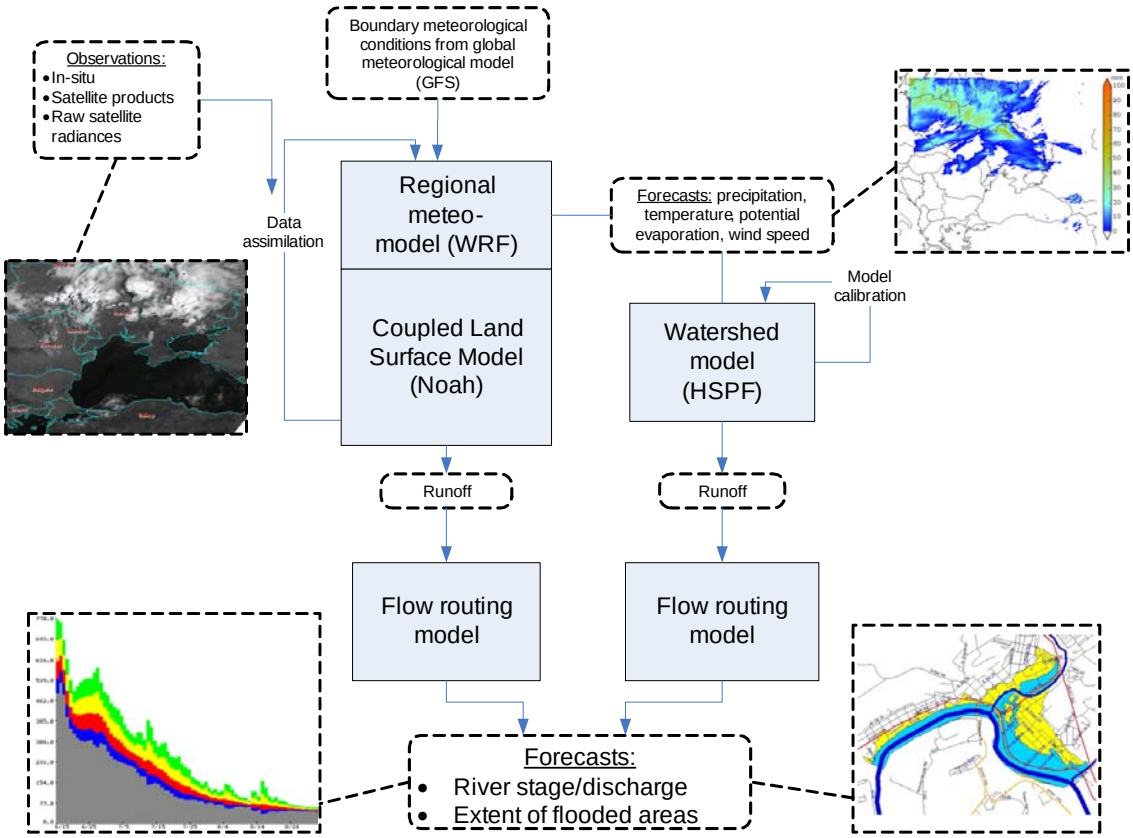


Fig. 1. Simulation cascade to predict flood events

To obtain quantitative estimates of precipitation and other meteorological forcing in the Space Research Institute WRF (Weather

Research&Forecasting) regional NWP model is used. WRF model is a joint development of a number of USA agencies and universities (http://wrf-model.org). This model was configured and adapted to the territory of Ukraine to run with spatial resolution of 10 km. Currently we routinely produce 72-hours weather forecasts every 6 hours [2]. To drive regional model the additional weather forecasts from global NWP model are used. These data are required to specify external meteorological forcing as boundary conditions for regional weather model. Currently we use forecast frames produced by GFS (Global Forecast System) model operated by NCEP.

10 km ground spatial resolution is generally enough to support regional applications while it is clearly insufficient to represent small-scale features like precipitation or temperature distributions in mountainous of the Carpathian region. Ground resolution of about 1 km is required to cope with such life-threaten events such as flash floods. Regional models such as WRF are designed to operate at such resolutions, however direct increase of spatial resolution results in enormous increase of required computation performance. To solve this antinome problem nesting approach is used. Within this method a set of nested model domains is used, each subsequent domain "zooms in" parent domain thus providing better resolution. Example of such approach applied to the Carpathians is shown in Fig. 2.

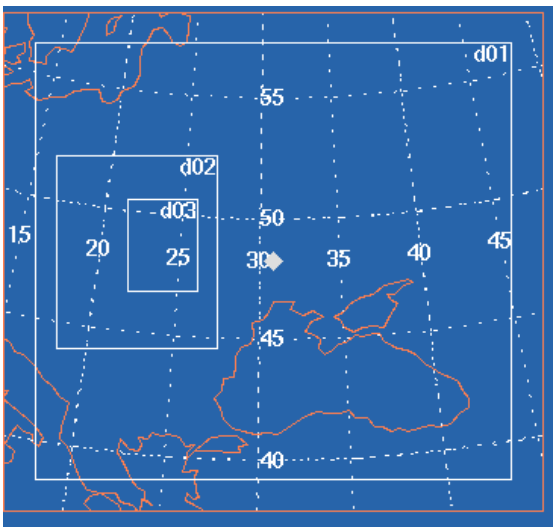


Fig. 2. Nested domains for Carpathian region to support high-resolution WRF forecasts

Two approaches are used to simulate hydrological processes in the Carpathians watersheds: more traditional one using dedicated hydrological

model and the second using coupled modeling. As dedicated hydrological model we use robust industry proven HSPF (Hydrological Simulation Program – Fortran, http://water.usgs.gov/software/hspf.html). HSPF is lumped model, i.e. within this type of models the whole watershed is divided into a number of sub-catchments that are considered hydrologically homogeneous. To run HSPF model several meteorological forcing parameters are required (at least precipitation and estimation of potential evapotranspiration). These data are obtained from high-resolution weather forecasts. Another approach we utilize is a coupled modeling. Within this approach we use Noah Land Surface Model (LSM) that is fully coupled with WRF model. Noah LSM is a distributed physically-based model that works with the same resolution as WRF model does.

Both hydrological model (HSPF and Noah) are able to compute quick response runoff to precipitation events. To route computed runoff and derive river stages and extent of flooded areas appropriate hydraulics models are involved.

Implementation. Modern hydrometerological models require High Performance Computations (HPC) to run at appropriate resolution. This is the most obvious for numerical weather prediction models, for instance it takes 4 hours on 4-processors AMD Opteron system to produce 72 hours forecast using WRF model. Therefore proper implementation of simulation cascade has to handle high performance computations. To run simulation models a Grid system of Space Research Institute [3] was used. This system is designed to distribute high performance computations over several HPC clusters. Now it includes two clusters of Institute of Cybernetics of NASU (SCIT-1 and SCIT-3) and cluster of Space Research Institute.

Use of clusters allows decreasing simulation time significantly while observing almost linear productivity growth within increasing number of computation nodes. To show this we used WRF model version 2.2 with model domain of 200x200x31 gridpoints with horizontal spatial resolution 10 km. Within this experiment 8 nodes of SCIT-3 cluster give performance increase in 7.09 times (of 8.0 theoretically possible) when compared to one node. 64 nodes increase performance in 43.6 times (fig. 3).

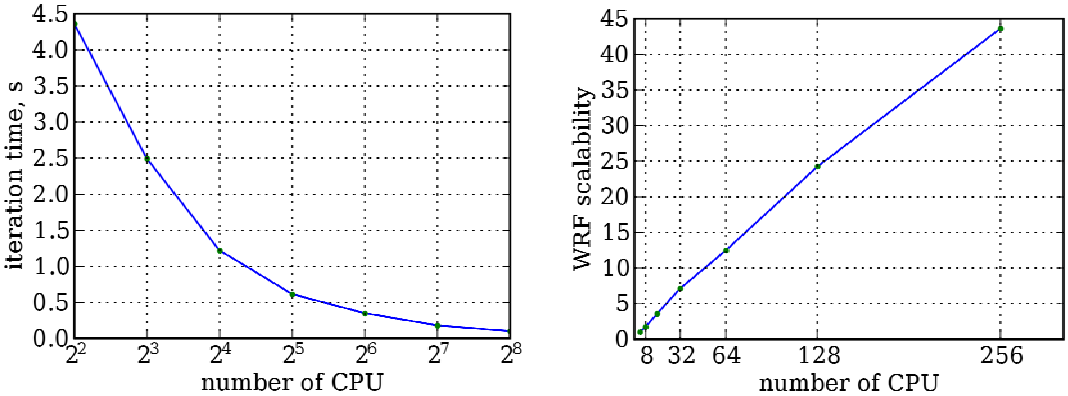


Fig. 3. WRF performance on SCIT-3 cluster: a) one iteration's computation time, b) acceleration of WRF model with respect to number of nodes

Conclusion. Creating simulation cascade is promising and straightforward approach to cope with complexity and interdisciplinary of modern models. This approach was used to integrate numerical weather prediction model, hydrological model and hydraulics model to predict flood events in Ukrainian part of Carpathian basins. Modern hydrometeorological models require a lot of computational recourses to solve real tasks and designed to run in massive parallel environment. Using High Performance Computing clusters, which are available for scientific community in Ukraine, to run models for real-sized problems showed their potential in case of hydrometerological models.

References

1. Hluchy L., *et al.* Collaborative environment for Grid-based flood prediction // Computing and Informatics, Vol. 24, 2005, 1001-1022.

2. *Shelestov A., Kravchenko O., Ilin M.* Geospatial data visualisation in Grid system on Ukrainian segment of GEOSS/GMES// Proc. of the V-th International Conference “Information Research&Applications”. — Varna (Bulgaria). — June 26-30, 2007. — Vol. 2. — P. 422-428.

3. *Шелестов А.Ю., Кравченко А.Н., Корбаков М.Б., Куссуль Н.Н., Скакун С.В., Рудакова А.И., Ильин Н.И., Тютюнник Л.И.* Grid-технология реализации украинского сегмента GEOSS // Загальногалузевий науково-виробничий журнал «Зв’язок» (спеціалізований додаток). — 2006. — С. 106-125.

Получено 01.06.07

УДК 681.3.068

МОДЕЛИРОВАНИЕ ФУНКЦИОНИРОВАНИЯ РЕЛЕЙНЫХ КЛЕТОК ЗРИТЕЛЬНОЙ СИСТЕМЫ В ЗАДАЧАХ ИНТЕРПРЕТАЦИИ ИЗОБРАЖЕНИЙ

Каргин А.А.[1)], Доценко И.А. Пятикоп Е.Е.[2)]
[1)]Донецкий национальный государственный университет
[2)] Приазовский государственный технический университет

Описуються етапи розпізнавання зображень методами когнітивних наук. Розглядається перший етап – представлення зображення на клітинному рівні. Пропонується нечітка модель релейної клітки.

Введение. Компьютерное зрение требуется в самых различных сферах человеческой жизни. Такие системы необходимы на производстве для отслеживания развития технологических процессов, в медицине, где получение, обработка и расшифровка изображений позволяет более точно поставить диагноз и контролировать ход хирургических операций, в робототехнике для ориентации в пространстве и навигации. Помимо этого анализ изображений выполняется в системах видеонаблюдения, а также для обеспечения взаимодействия человека и компьютера при помощи голоса, жестов и движений.

Одной из наиболее важных проблем разработки систем компьютерного зрения является задача распознавания изображения. Актуальной задачей на сегодняшний день остается задача распознавания лица человека на изображении, но помимо этого решаются также задачи распознавания дорожных знаков [1], контуров губ [2], символов и другие. Процесс распознавания традиционно включает такие этапы: преобразование исходного изображения в начальное представление (может включать в себя как предобработку, так и математические преобразования); выделение ключевых характеристик; классификацию объекта.

Анализ литературных источников показал, что для решения этой задачи применяются такие средства как линейный дискриминантный анализ, факторный анализ, фильтры, метод главных компонент, скрытые Марковские модели, нейронные сети и другие [3-5]. Параллельно с ними исследуется возможность применения методов когнитивных наук [6-8].

В когнитивной психологии на основе наблюдений за человеком за длительный период предложны и апробированы экспериментально модели организации практически всех когнитивных функций

человека: восприятие зрительной, звуковой информации, организация памяти разных уровней [8-9].

Эти знания дают возможность моделировать строения восприятия глаза человека и применять полученные модели для обработки изображения. Использование такого подхода позволяет представить изображение в виде моделей отдельных элементов. Тогда можно говорить не только о решении задачи обнаружения или локализации объекта, но и об определении его элементов. Результат структурирования изображения в виде моделей элементов служит основой для понимания объекта в общем, поскольку определение (классификация) элементов изображения создает дополнительный уровень знаний, на основе которых можно судить об объекте в целом. Применение этого способа можно разбить на такие этапы обработки изображения:

1. представление изображения на клеточном уровне;
2. структурирование изображения на уровне ячеек;
3. интерпретация изображения с помощью модели ситуационного логогена.

В настоящей статье рассматривается первый этап. Целью моделирования является получение нечеткой модели представления изображения на клеточном уровне.

Модель релейной клетки. В основу модели положена работа ганглиозные клеток зрительной системы, которые отвечают за определение контура. Как привило, мы видим контуры тогда, когда примыкающие друг к другу поверхности освещены по-разному. Для некоторых ганглиозных клеток, если свет попадает на маленькую область сетчатки, будет отмечаться увеличение частоты генерации разрядов (ON -центр). Но если свет падает на область вокруг этого чувствительного центра, частота генерации разрядов снижается. Свет, падающий в отдалении от центра, вообще не вызывает никакой реакции (OFF- центр). Эти клетки известны как релейные [8-9].

По аналогии работы описанных клеток представим исходное пиксельное поле P в виде множества клеток K_j. Каждая клетка сформирована на множестве пикселей $\{p_i\}$ начального изображения и состоит из ядра и полости, модель клетки показана на рисунке 1.

Каждый пиксель p_i характеризуется величиной яркости f, которая принимает значение из промежутка [0,1].

Характеристикой ядра является радиус Rk, а характеристикой полости – Rc. Размер этих величин определяется в зависимости от

размера исходного изображения или задается перед процессом моделирования.

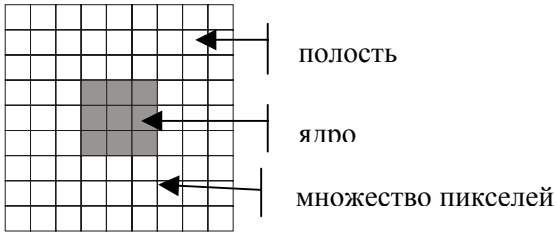


Рис. 1 – Модель релейной клетки и ее характеристики

Для каждой клеток K_j необходимо определить принадлежность к каждому из типов, показанных на рисунке 2.

а) б)

Рисунок 2 – Модели релейных клеток разных типов:
а) тип «включено» (ON), б) тип «выключено» (OFF)

Поскольку фрагменты изображения не будут однозначно соответствовать приведенным типам, то принадлежность к конкретному типу будет иметь нечеткий характер. Тогда характеристиками каждой клетки являются функция уверенности $\theta(K_j)$, которая показывает величину принадлежности K_j клетки к типу «включено» (ON) и функция уверенности $\varphi(K_j)$, которая показывает величину принадлежности K_j клетки к типу «выключено» (OFF). При этом значение величин определяется в промежутке [0, 1].

Таким образом, основными параметрами модели клетки являются:

- множество функций яркостей $f(p_i)$ пикселей;
- радиус ядра *Rk*;
- радиус полости *Rc*;
- множество пикселей $Pk \in \{pk_i\}$ принадлежащих ядру радиусом *Rk;*
- множество пикселей $Pc \in \{pc_i\}$ принадлежащих полости радиусом *Rc;*
- θ – значение активности клетки типа «ON»;

- φ – значение активности клетки типа «OFF»;
- функция принадлежности $\mu_{\underset{\sim}{ON}}(\theta_j)$ клетки к типу «ON»;
- функция принадлежности $\mu_{\underset{\sim}{OFF}}(\varphi_j)$ клетки к типу «OFF»;

Определение функций принадлежностей $\mu_{\underset{\sim}{ON}}(\theta_j)$ и $\mu_{\underset{\sim}{OFF}}(\varphi_j)$ показано на рисунке 3.

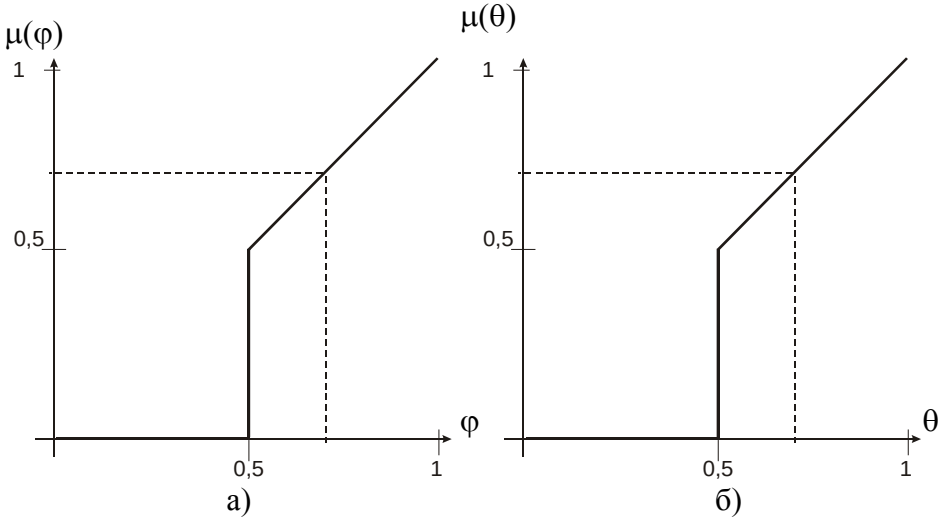


Рисунок 3 – Графики функций принадлежности
а) для клетки типа «OFF», б) для клетки типа «ON»

Величина θ_j рассчитывается по формуле (1), а φ_j по формуле (2)

$$\theta_j = \frac{1-\chi(Pk)+\chi(Pc)}{2} \quad (1)$$

$$\varphi_j = \frac{1-\chi(Pc)+\chi(Pk)}{2} \quad (2)$$

где величина $\chi(Pk)$ показывает насколько белое ядро, а $\chi(Pc)$ – насколько белая полость и рассчитывается по формуле (3).

$$\chi(R) = \frac{\sum_{p\in R} f(p_i) - n\cdot\min(f(p_i))}{\delta\cdot n}, \quad (3)$$

где $R \in \{Pk, Pc\}$, n- количество пикселей области R,

для $R \in Pc \quad n = (2\cdot Rc)^2 - Rk^2$, $R \in Pk \quad n = Rk^2$, $\delta = \max(f(p_i)) - \min(f(p_i))$

$\min(f(p_i))$ - минимальное значение функции яркости;

Определение наличия фрагмента контура в ячейке возможно в случае активизации детектора края или полосы, которые формируются из групп клеток противоположных типов. Так детектор края активизируется, если группа клеток типа «ON», расположенных в некотором направлении, граничит с клетками типа «OFF» с любой из сторон. Детектор полосы имеет место, если группа клеток типа

«ON», расположенных в некотором направлении, обрамлена с обеих сторон клетками типа «OFF». [9-10]. В связи с этим для каждой клетки K_j имеются также характеристики:

- α – величина участия клетки K_j в активизации детектора полосы;
- β – величина участия клетки K_j в активизации детектора края.

Значение α_i и β_i засчитываются по формулам (4) и (5) соответственно:

$$\alpha_i = \frac{\mu(\varphi_{i-1}) + \mu(\theta_i) + \mu(\varphi_{i+1})}{3} \quad (4)$$

$$\beta_i = \frac{\mu(\theta_i) + \mu(\varphi_{i+1})}{2} \quad (5)$$

Выводы. Для проведения экспериментов разработан инструментальный комплекс, позволяющий обрабатывать изображения с целью получения нечетких характеристик направлений фрагментов контура в ячейках. В ходе экспериментов были использовались изображения с разрешением 72 точки на дюйм, размером от 100x100 пикселей и выше. Изображения представляли собой различные фигуры в черно-белой палитре, а также градации серого до 8 бит.

Результаты положительной проверки описанной модели позволяют перейти к следующему этапу моделирования.

Литература

1. Применение поведенческой модели зрения для инвариантного распознавания лиц и дорожных знаков / Шапошников Д.Г. и др.; Режим доступа: http://nisms.krinc.ru/papers/neurocomp2002_vision.htm - название экрана.
2. Чтение по губам: распознавание контуров губ / Солдатов С. А.; Лаборатория компьютерной графики, ВМК МГУ, Режим доступа: graphics.cs.msu.su/ru/publications/text/ioi2002sss – название экрана.
3. Распознавание человека по изображению лица и нейросетевые методы / Брилюк Д., Старовойтов В.; Публикации Daily.Sec.ru, Режим доступа: http://daily.sec.ru/dailypblshow.cfm?rid=5&pid=4425&pos=1&stp=50 – название экрана.
4. Доклады IX всероссийской конференции Математические методы распознавания образов // Российская академия наук вычислительный центр при поддержке Российского фонда фундаментальных исследований, «АЛЕВ-В», Москва 1999, стр. 271
5. Обнаружение и локализация лица на изображении/ Вежневец В., Дегтярева А.; Научно-образовательный сетевой журнал «Графика и Мультимедиа», Режим доступа: http://cgm.graphicon.ru/content/view/40/28/ – название экрана.
6. A Biological Neural Network of Visual Cell Responses: Static and Motion Processing / Journal of the Brazilian Computer Society; Режим доступа http://www.scielo.br/scielo.php?pid=S0104-65001997000200002&script=sci_arttext&tlng=en – название экрана.
7. Learning to See: Genetic and Environmental Influences on Visual Development Bednar, J. A. / The University of Texas at Austin Режим доступа: www.iovs.org/cgi/reprint/18/5/524.pdf –название экрана
8. Андерсон Дж. «Когнитивная психология. 5-е изд.» – СПб.: Питер, 2002. – 496с. Серия «Мастера психологии»
9. Солсо «Когнитивная психология» – СПб.: Питер, 2002. – 592с.

Работа поступила в редакцию 05.06.07.

УДК 0004.04:576.08

ИССЛЕДОВАНИЕ И ВЫБОР МЕТОДА ВЫДЕЛЕНИЯ ПРИЗНАКОВ ИЗОБРАЖЕНИЯ КЕРАТИНОЦИТОВ ПРИ ОЦЕНКЕ ЖИЗНЕСПОСОБНОСТИ И КОЛИЧЕСТВЕННОГО РОСТА КЛЕТОК

Каира В.В.

Донецкий национальный технический Университет, кафедра Автоматизированных систем управления

Рассмотрены фрактальный, спектральный, вейвлет и статистический подходы получения признаков изображения. Проведен анализ результатов классификации признаков различных классов изображений кератиноцитов. Предложен спектральный метод получения признаков изображений для использования в компьютерной системе оценки качества и прогнозирования сроков созревания кератиноцитов.

Введение. В работе [1] была поставлена задача разработки компьютерной системы прогнозирования сроков созревания кератиноцитов, клеток кожи, выращиваемых в специализированных медицинских лабораториях для пересадки пациентам на пораженные ожогами или болезнями участки кожных покровов. Согласно [1] компьютерная система должна включать подсистему контроля качества выращиваемой культуры. Входом данной подсистемы являются снимки участков культурального матраса с клетками, выходом – процентное заполнение матраса жизнеспособными клетками. Правильный выбор наиболее информативных признаков изображения способен увеличить точность классификации снимков в подсистеме контроля качества выращиваемой культуры.

Постановка задачи. После съемки изображения матраса снимок поступает на вход подсистемы контроля качества выращиваемой культуры (рис. 1). С выхода блока предварительной обработки изображение подается на блок выделения признаков изображения.

В этом блоке необходимо выработать множество информативных признаков изображения, по которым с помощью

классификатора возможно сформировать значение, соответствующее визуальному процентному заполнению снимка живыми клетками.

Получение такого признака с помощью анализа гистограммы яркостей изображения либо контурного препарата изображения [2] не представляется возможным, т.к. яркость внутреннего заполнения клеток на изображении часто соответствует яркости фона и невозможно утверждать о принадлежности оконтуренного участка фону или клетке.

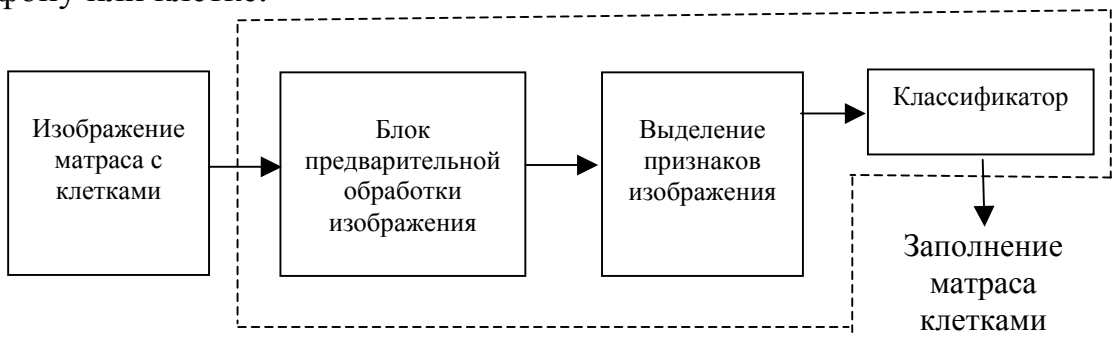


Рис. 1 - Структура подсистемы контроля качества выращиваемой культуры

С помощью детекторов границ изображения возможно получить контуры клеток, однако они неотличимы от ярких участков снимков (соответствующих изначально посеянным группам клеток, прикрепившимся к культуральному матрасу и начинающим делиться и расти). Поэтому основной задачей при реализации подсистемы контроля качества выращиваемой культуры является выбор математического метода формирования признака заполнения изображения живыми клетками и его классификации. Анализируемые снимки произведены с помощью микроскопа на увеличении ×100 с разрешением 512×512 пикселей в 256 градациях серого.

Решение задачи. Признаки изображения должны обладать инвариантностью к повороту и незначительным изменениям масштаба. В работе [3,4] предложен подход к классификации текстуры изображения, основанный на анализе коэффициентов вейвлет разложения. Для обеспечения инвариантности к повороту предварительно предлагается использовать логарифмически-полярное преобразование изображения. После логарифмически-полярного преобразования изображение подлежит вейвлет декомпозиции C и является входным изображением на уровне разложения q=0, т.е. C^0_0.

Для проведения эксперимента в работе был использован вейвлет Добеши 10-го порядка. После декомпозиции для отдельного уровня разложения каждого изображения вычислялась мера энергии:

(2)

Признаком изображения принимался упорядоченный по убыванию вектор значений e_m, m=1,…,96 – вейвлет сигнатура.

В литературе встречается метод классификации текстур и медицинских изображений с помощью фрактальной размерности [5]. Для вычисления фрактальной размерности необходимо положить, что наблюдаемое серое изображение I(x,y), x=0,..,N, y=0,..,M, представляет собой фрактальную броуновскую поверхность; x и y – координаты пикселей изображения, N и M – размеры изображения. Тогда согласно закону дисперсии и стационарности [5], вытекающему из гауссовости приращений, должно выполняться отношение:

$$E(|I(x2,y2)-I(x1,y1)|)\propto \qquad , \qquad (3)$$

где E(|I(x2,y2)-I(x1,y1)|) – математическое ожидание изменения яркости, H – фрактальный параметр.

Другой часто используемый подход извлечения статистических характеристик текстуры базируется на вычислении матрицы совместной встречаемости (матрицы смежностей, GLCM) [2]. Обозначив эту матрицу S(i, j, d, θ), где i и j – яркости соседних точек на изображении, расположенных на расстоянии d друг от друга, при угловом направлении θ. Матрицы содержат информацию характеризующую текстуру. По матрице совместной встречаемости вычисляется около двадцати различных признаков, в данной работе использована энергия:

. (4)

В [1] предлагается судить о заполнении изображения матраса изображениями клеток, анализируя двумерный спектр снимка [2]. Такой подход инвариантен к изменению средней яркости изображения. Для изображения I(x,y) размером N×M спектр определяется как

, (5)

где x и y – координаты пикселей изображения, $F(w_x,w_y)$ – спектр изображения, и , $n_x = 0,\ldots,N-1$, $n_y = 0,\ldots,M-1$.

На практике при анализе комплексной функции $F(w_x,w_y)$ целесообразно принимать во внимание модуль её значения:

. (6)

Функция модуля спектра инвариантна к сдвигу [2], для обеспечения инвариантности к вращению применяется радиальная функция признаков:

, (7)

где $\theta = \mathrm{arctg}(w_y/w_x)$ и $\rho^2 = w_x^2+w_y^2$.

Результаты. С целью проверки возможностей предложенных подходов анализа изображения матраса по классификации процентного заполнения клетками были реализованы программы в среде программирования Borland C++ Builder 6. Для классификации радиальных функций признаков и вейвлет сигнатур использовался простой и достаточно точный метод классификации – классификатор на основе Евклидовых расстояний [6]. Экспертами было определено 3 класса изображений культурального матраса (рис.2):

1) изображения матраса, с низким уровнем заполнения клетками;
2) изображения матраса, с средним уровнем заполнения клетками;
3) изображения матраса, с полным заполнением клетками.

Для формирования эталонных точек классов были выбраны десять эталонных снимков каждого из классов. Статистическая обработка результатов работы методов была выполнена в программе MS Excel 2003. Для каждого из снимков были получены следующие характеристики: фрактальная размерность, энергия текстуры, радиальная функция признаков Фурье, 96-значный вектор признаков вейвлет разложения. При выбранном уровне значимости α=0,001 для каждого из классов были сформированы доверительные интервалы. Для Евклидова расстояния вейвлет сигнатур невозможно сформировать доверительные интервалы для второго и третьего эталонных классов, различия между ними не значимы даже при уровне значимости α=0,05. Согласно [6], при использовании классификатора на основе Евклидовых расстояний целесообразно использовать следующий алгоритм классификации без построения доверительных интервалов:

1. Сформировать точки, соответствующие эталонным классам.
2. Проверить Евклидово расстояние тестируемой точки до каждой из точек, представляющей эталонный класс.

3. Отнести тестируемую точку к классу, Евклидово расстояние до эталонной которого минимально.

Была выполнена классификация 30 снимков, по 10 снимков в каждом из классов. Обобщенные результаты точности классификации снимков кератиноцитов с использованием предложенных методов выделения признаков изображений приведены в табл. 2.

Выводы. В исследовании был выполнен анализ возможного применения в компьютерных системах прогнозирования сроков созревания кератиноцитов нескольких методов выделения признаков изображения, таких как фрактальный (использующий фрактальную размерность), статистический (использующий признаки матрицы смежности), частотный (использующий радиальный признак Фурье-спектра), вейвлет (использующий предварительное полярное преобразование).

Таблица 2

Показатели точности классификации для различных методов

	Точность при построении доверительных интервалов с α=0,001	Точность при использовании классификатора на основе Евклидовых расстояний
Энергия	77%	-
Фрактальная размерность	80%	-
Радиальный Фурье признак	83%	87%
Вейвлет сигнатура	56%	80%

Сравнительный анализ методов показал, что для компьютерных систем прогнозирования сроков созревания кератиноцитов наиболее целесообразным является использование метода радиального признака Фурье. Показано, что использование классификатора способно увеличить точность распознавания снимков кератиноцитов.

В направлении дальнейшего развития исследования предполагается разработка классификатора, что позволит использовать большее число классов изображений без значительного снижения точности, например классификатор на основе аппарата нейронных сетей.

Литература

1. Адамов В.Г., Каира В.В. Компьютерная система прогнозирования сроков созревания кератиноцитов // Моделювання та керування станом еколого-економічних систем регіону. – Київ: Міжнародний науково-навчальний центр

інформаційних технологій та систем НАН України та МОН України, 3 вип., 2006.
2. Pratt William K. Digital Image Processing: PIKS Inside, Third Edition. — John Wiley & Sons Inc., 2001, 735 p.
3. Chi-Man Pun, Moon-Chuen Lee. Log-polar wavelet energy signatures for rotation and scale invariant texture classification // IEEE Transactions on pattern analysis and machine intelligence, vol.25, no.5, May 2003, pp. 590-603.
4. Меркулова Е.В. Создание модели процесса определения жизнеспособности культивируемых фибробластов для автоматизированной системы // Вестник Херсонского Государственного технического университета. - Херсон: ХГТУ.- 2004 р. -№ 1(19).
5. Кроновер Р. М. Фракталы и хаос в динамических системах. Основы теории. Москва: Постмаркет, 2000, — 352 с.
6. Bow Sing-Tze. Pattern Recognition and Image Preprocessing, Second Edition, Revised and Expanded. — Marcel Dekker Inc., New York-Basel. 2002, 698p.

Получено 22.05.2007

УДК 621.3

ОСОБЕННОСТИ ИННОВАЦИОННОЙ ПОДГОТОВКИ СПЕЦИАЛИСТОВ В ОБЛАСТИ МИКРОКОМПЬЮТЕРНЫХ СИСТЕМ НА ПЕРВЫХ КУРСАХ ОБУЧЕНИЯ

А.Б. Клевцова
Технологический институт
Южного федерального университета в г. Таганроге

В статье рассматриваются некоторые аспекты построения системы подготовки специалистов в Технологическом институте Южного федерального университета на кафедре "Микропроцессорные системы" (МПС) на первых курсах обучения.

Создание современных локальных и распределенных систем мониторинга, диагностики и управления сложными техническими объектами, которые обеспечат повышение эффективности функционирования этих объектов, улучшение их эксплуатационных и надежностных характеристик, зависит от наличия высоких технологий и наукоемких производств.

Такие системы располагаются в непосредственной близости от объектов наблюдения и управления, в темпе реального времени с помощью датчиков физических величин снимают с них информацию, обрабатывают данные и выдают результаты в компьютерные системы более высокого уровня для дальнейшей обработки или на органы управления [1,2]. В них реализуются функции многоуровневого интегрального контроля состояния в результате применения измерительных микропроцессорных интеллектуальных систем и организации интеллектуальной распределенной обработки информации [3].

На успешное внедрение современных микрокомпьютерных систем определяющее влияние оказывает наличие высококвалифицированных специалистов-разработчиков.

Как показывает анализ реализаций микрокомпьютерных систем мониторинга, диагностики и управления в машиностроительной и приборостроительной промышленности, в металлургии, в топливно-энергетическом комплексе, химико-лесном комплексе, текстильной и легкой промышленности, телекоммуникациях и связи, на транспорте [1,2,3] современный специалист должен:

– владеть методикой анализа особенностей функционирования технических объектов и протекания технологических процессов с

целью создания для них микрокомпьютерных систем контроля и управления;

– уметь синтезировать схемы организации процессов наблюдения и управления, разрабатывать их математические описания;

– знать методы обработки сигналов и данных, снимаемых с датчиков технических объектов и технологических процессов;

– обладать теоретическими знаниями и практическими навыками проектирования и эксплуатации алгоритмического и программного обеспечения создаваемых микрокомпьютерных систем;

– обладать знаниями и владеть методами проектирования аппаратуры локальных и распределенных микрокомпьютерных систем;

– владеть основами управления проектом создания системы, последующего ее внедрения и эксплуатации;

– владеть современными инструментальными средствами построения и отладки аппаратного и программного обеспечения микрокомпьютерных систем.

Этим, конечно, не исчерпывается список требований, которые предъявляются к высококвалифицированному инженеру. Однако он в достаточной степени определяет облик современного инженера рассматриваемого профиля и поэтому положен в основу построения системы подготовки специалистов в Технологическом институте Южного федерального университета на кафедре "Микропроцессорные системы" (МПС).

Рассмотрим особенности этой системы подготовки на первых курсах обучения.

Одним из существенных недостатков существующей подготовки специалистов технического профиля является практическое отсутствие на ранних курсах обучения связи студентов с будущей специальностью.

Абитуриенты при поступлении в вуз надеются в течение всего обучения быть неразрывно связанными с выбранной специальностью. Однако, первые годы учебы в этом отношении вызывают разочарование, поскольку обучение проводиться по общим для всего института программам, в них часто отсутствует какая-либо практическая привязка не только к специальности, но и к технической направленности подготовки.

Это обстоятельство снижает мотивацию студентов к обучению, и, как следствие, негативно влияет на успеваемость и желание успешного усвоения материала.

Принятый на первых курсах обезличенный подход к обучению противоречит общемировым тенденциям развития инновационного образования и лишает привлекательности как вуз, так и конкретную специальность в глазах абитуриентов и работодателей.

Особенностью учебного процесса, проводимого на кафедре МПС, является подготовка студентов с постоянной привязкой к будущей специальности, начиная с первого курса.

Основной задачей инновационной подготовки на ранних курсах обучения служит обязательная демонстрация необходимости и важности получаемых в рамках каждой конкретной дисциплины знаний и практических навыков для успешного освоения специальности. Особенно важно при этом использовать не абстрактные, теоретические примеры, а реальные программно-аппаратные и технические устройства, стенды и модели.

Когда студент, уже на первом курсе, может применить свои знания в практическом плане, даже если результатом будет простейшая процедура, например, управляемое мигание светодиода на стенде, он начинает понимать и главное, получает этому подтверждение, что обучение, это не последовательность слабо связанных между собой дисциплин, а единая технология, каждый этап которой – необходимый шаг к специальности.

Задачи, решаемые студентом, при этом должны быть интересными и достаточно простыми, доступными без подготовки и излишних усилий всем студентам. Ни в коем случае их нельзя делать подобными лабораторным или практическим работам в их классическом виде. Цель этих заданий – поддержать недавнего абитуриента в желании учиться, заставить его поверить в свои силы, заинтересовать его в освоении предмета. Задания на следующих курсах могут развиваться, однако критерии простоты и максимальной доступности должны соблюдаться без исключения.

Выполняемые студентом задания должны быть построены таким образом, чтобы он смог воспользоваться знаниями и навыками, полученными в рамках данного предмета, то есть должна быть очевидная и ясная связь с читаемыми вопросами.

Известно, что в стандартных программах дисциплин не предусмотрено место для таких «отвлечений». Тем не менее, наиболее оптимальным является реализация задачи на практических занятиях, если они предусмотрены.

Важной составляющей инновационного образования является реализация вертикальной связи между дисциплинами, преподаваемыми в различных семестрах. Эта связь может быть реализована с помощью сквозных проектов, когда на одном и том же оборудовании или в программно-аппаратной среде формируется сложная задача, решаемая студентом последовательно в течение прохождения ряда дисциплин, начиная с первого курса.

В качестве примера применения данного подхода рассмотрим особенности преподавания дисциплины «Информатика» на первом курсе специальности 210106 «Промышленная электроника». Поскольку кафедра готовит инженеров в области микрокомпьютерных систем автоматики в различных отраслях промышленности, науки, техники и т.п., то специализация на ней направлена на изучение и освоение дисциплин проектирования математического, алгоритмического, программного и аппаратного обеспечения таких систем.

Одним из базовых направлений подготовки является программно-алгоритмическое. Программно-алгоритмическое направление ориентируется на обучение студентов методам и средствам исследования и описания организации объектов и процессов, выявления их основных характеристик, определения базовых функций и процедур проектируемой микрокомпьютерной системы, разработки и моделирования реализаций машинных алгоритмов с требуемыми параметрами точности при допустимой вычислительной сложности и т.д.

В рамках дисциплины «Информатика», учитывая микрокомпьютерную направленность специальности, преподаются основы построения алгоритмов выполнения типовых процедур (процессов), языки Паскаль и СИ. Технология обучения поставлена таким образом, что при изучении того или иного алгоритма параллельно приводятся программы его реализации на обоих языках (на языках СИ и Паскаль). Такой подход обеспечивает наряду с освоением сразу двух языков, еще и параллельный анализ этих языков, определение эффективности, выявление их достоинств и недостатков при реализации различных алгоритмов, особенности и предпочтения при использовании языков. Однако, преподавание алгоритмов и языков на первом курсе, как правило, носит абстрактный характер, рассматриваемые примеры и алгоритмы реализуются только в программной среде и не связаны напрямую с алгоритмами и программами, реализуемыми при разработке и

эксплуатации микропроцессорных систем мониторинга и управления техническими объектами и процессами, что является одной из задач инженера данной специальности.

Чтобы установить эту связь и определить важность этой дисциплины для специальности, в рамках практических занятий студенты выполняют специализированные задания с использованием аппаратно-программных стендов разработки систем на контроллерах AVR, стенда промышленной электроники Omron, аппаратно-программных комплексов для обучения разработке систем на основе технологии "Система на кристалле". Конечно, студенты первого курса еще не готовы к полномасштабному освоению указанного учебно-исследовательского оборудования, они не знакомы с структурой и построением контроллера, особенностями разработки микропроцессорных алгоритмов и т.д. Тем не менее, они обладают достаточными знаниями, чтобы реализовать некоторые практические алгоритмы управления. С этой целью задания сформированы, а программно-аппаратная среда сконфигурирована таким образом, чтобы студент мог в рамках стендов реализовать логику процесса управления и получить видимый и осязаемый результат на реальном техническом устройстве или стенде. Это, например, имитация работы светофора на стенде Omron, съема информации с датчика температуры и сигнализация о нештатной ситуации на базе отладочного набора CY3210 - PSoCEval1 («Система на кристалле»), отображение на индикаторе вводимых с клавиатуры цифр с помощью стенда AVR и др.

При этом каждый из студентов, сформировав с помощью визуальных средств проектирования алгоритм управления, может увидеть свой алгоритм в действии и посмотреть программу на СИ, которая автоматически генерируется интегрированными в стенды программными средствами.

Таким образом, студенты непосредственно в практическом плане видят, как их знания и навыки реализуются в технических устройствах, что данная дисциплина действительно необходима инженеру и ее изучение в будущем даст возможность создавать алгоритмы и программы для настоящих микропроцессорных систем и промышленных средств автоматизации.

Таким образом, в результате принятой схемы преподавания дисциплин создается система подготовки специалистов со встроенной сквозной системой практических заданий, адаптированных к

текущему уровню подготовки и реализующих связь преподаваемых дисциплин со специальностью, начиная с первых курсов обучения студентов.

Литература

1. Васильев А., Заречнев В. Система контроля температуры металлургический печи.// СТА.- №2. -1998. - С.18-22.
2. Бодяжин А., Трофанюк В. Автоматизированный эколого-аналитический мониторинг источников загрязнения поверхностных вод.// СТА. - №2. - 2002. - С.68-74
3. Токарев А., Придачин В., Стороженко П., Кривошеев А., Еникеев А., Покровский А. Информационно-измерительная система реакторной установки СМ-3.// СТА. - № 3. - 2002. - С.46-54.

Работа поступила в редакцию 04.06.2007

УДК 004.048+004.932

ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА ОБРАБОТКИ И РАСПОЗНАВАНИЯ ИЗОБРАЖЕНИЙ ГИСТОЛОГИЧЕСКИХ СРЕЗОВ НА ОСНОВЕ ЭВОЛЮЦИОННЫХ МОДЕЛЕЙ

Мартыненко Т.В.
Донецкий национальный технический университет, г. Донецк
кафедра автоматизированных систем управления

У статті представлена структура програмної реалізації системи інструментальних засобів обробки й розпізнавання зображень гістологічних зрізів на базі еволюційних моделей. В якості цільової функції для добору кращих рішень запропоновано використовувати оцінку систематичної помилки пошуку залежно від чутливості й від специфічності, що враховує глибину програмного графа. Зазначена система інструментальних засобів підтвердила свою ефективність у процесі експериментів на базі патолого-анатомічного відділення ДОКТМО при проведенні гістологічних досліджень.

Введение. Актуальность задачи выявления опухолевых клеток определяется возрастанием числа раковых заболеваний, которые являются второй по частоте причиной смертности в мире. Для диагностирования и прогнозирования развития новообразований обязательным является гистологическое исследование атипичных клеточных структур. Различные по тканевой принадлежности опухоли строго специфичны на тканевом, клеточном и на структурном уровне. Однако имеется ряд морфологических и цитологических методов, позволяющих определить доброкачественность или злокачественность исследуемой опухоли: размеры клеток, размеры ядра клеток, ядерно-цитоплазматическое соотношение и т. д

Применение автоматизированных систем обработки и анализа изображений (АСОАИ) позволяют существенно повысить достоверность обнаружения изменений внутренних органов при онкологических заболеваниях, ускорить процесс обработки препаратов, а также расширяет возможности профилактики и предупреждения злокачественных новообразований за счет повышения уровня автоматизации обработки изображений гистологических срезов и качества диагностики. Основной функцией

АСОАИ является извлечение требуемой, значимой информации, определяющей эффективность работы систем обработки и распознавания медицинских изображений (СОРМИ).

Масштабы применения методов и средств цифрового анализа изображений в СОРМИ в настоящее время ограничены из-за низкой точности классификации, отсутствия эффективного автоматического подбора параметров сегментации и определения эффективных последовательностей операторов обработки изображений.

Для преодоления указанных недостатков разработана модификация эволюционного подхода по построению эффективных алгоритмов обработки и распознавания изображений гистологических срезов [1]. В котором потенциальное решение (хромосомы) представлены в виде направленных ациклических графов, где нетеминальные узлы являются стандартными операторами обработки изображений, а конечные - выходными плоскостями изображений. Разработаны функциональное и терминальное множества, проблемно-ориентированные операторы кроссинговера и мутации.

Краткая постановка задачи разработки инструментальных средств обработки и распознавания изображений. Задача инструментальных средств обработки и распознавания изображений на основе эволюционных моделей заключается в том, чтобы по изображениям обучающего множества G определить такую последовательность операций обработки изображений (ООИ) с соответствующим набором параметров, при которой критерий оценки качества распознавания F(G) принимает экстремальное значение ($f_Ц \rightarrow 1$).

Качество процесса обработки и распознавания изображений (ОРИ) определяется с помощью оценки целевой функции (ЦФ) $f_Ц$. Для задач обработки и распознавания изображения применяются ЦФ [1-2], которые не учитывают подстройку систематической ошибки поиска в зависимости от значений чувствительности (*SV*) и специфичности (*SP*):

$$SV = \frac{TP}{TP + FN}, \quad (1)$$

$$SP = \frac{TN}{TN + FP}, \quad (2)$$

где *TP* - количество правильно найденных значений позитива,
FP – количество неправильно найденных значений позитива,
TN - количество правильно найденных значений негатива,
FN - количество неправильно найденных значений негатива.

Для поставленной задачи в отличии от [1-3] предлагается целевая функция, использующая понятия чувствительности (*SV*) и специфичности (*SP*):

$$f_d = \beta \cdot SP \cdot SV\,, \qquad (3)$$

где β - нормализирующий коэффициент, который определяется по формуле:

$$\beta = \frac{(md - 0{,}5 \cdot d - 0{,}5)}{md - 1}, \qquad (4)$$

где d – количество узлов текущей особи,

md – параметр ГП соответствующий максимальной длине дерева.

Структура программной реализации системы инструментальных средств. Система инструментальных средств имеет модульную структуру и состоит из следующих основных подсистем (рис. 1):

— Подсистема обработки изображений;

— Подсистема подбора эффективных параметров сегментации;

— Подсистема построения эффективных алгоритмов обработки и распознавания изображений гистологических срезов.

Система инструментальных средств реализована на объектно-ориентированном языке высокого уровня Delphi 7.0. Разработана в формате СУБД Paradox 7.0 база данных (БД), в которой предусмотрено хранение полученных параметров сегментации и алгоритмов обработки и распознавания изображений гистологических срезов.

«Подсистема обработки изображений» включает в себя набор операторов обработки изображений и морфологические операторы. Также подсистема включает в себя процедуры вычисления значений целевых функций. В случае, если в системе уже существует построенный алгоритм для данного вида гистологического исследования, то «Подсистема обработки изображений» производит чтение соответствующей информации из БД. Далее происходит последовательное применение операторов обработки и распознавания изображений с рассчитанными параметрами.

«Подсистема подбора эффективных параметров сегментации» осуществляет генетический поиск рациональных параметров сегментации изображений. Подсистема позволяет получить значения коэффициентов для отсечения по низким и высоким частотам выбранной плоскости данных. Результаты представляются в виде числовых значений параметров выходной плоскости изображения. По

желанию лица, принимающего решение (ЛПР) значения коэффициентов могут быть занесены в БД, а выходная плоскость сохранена на жесткий диск. В «Подсистему подбора эффективных параметров сегментации» входит модуль стандартного генетического алгоритма, позволяющий осуществлять механизмы селекции, мутации и репродукции и стратегию элитарного отбора.

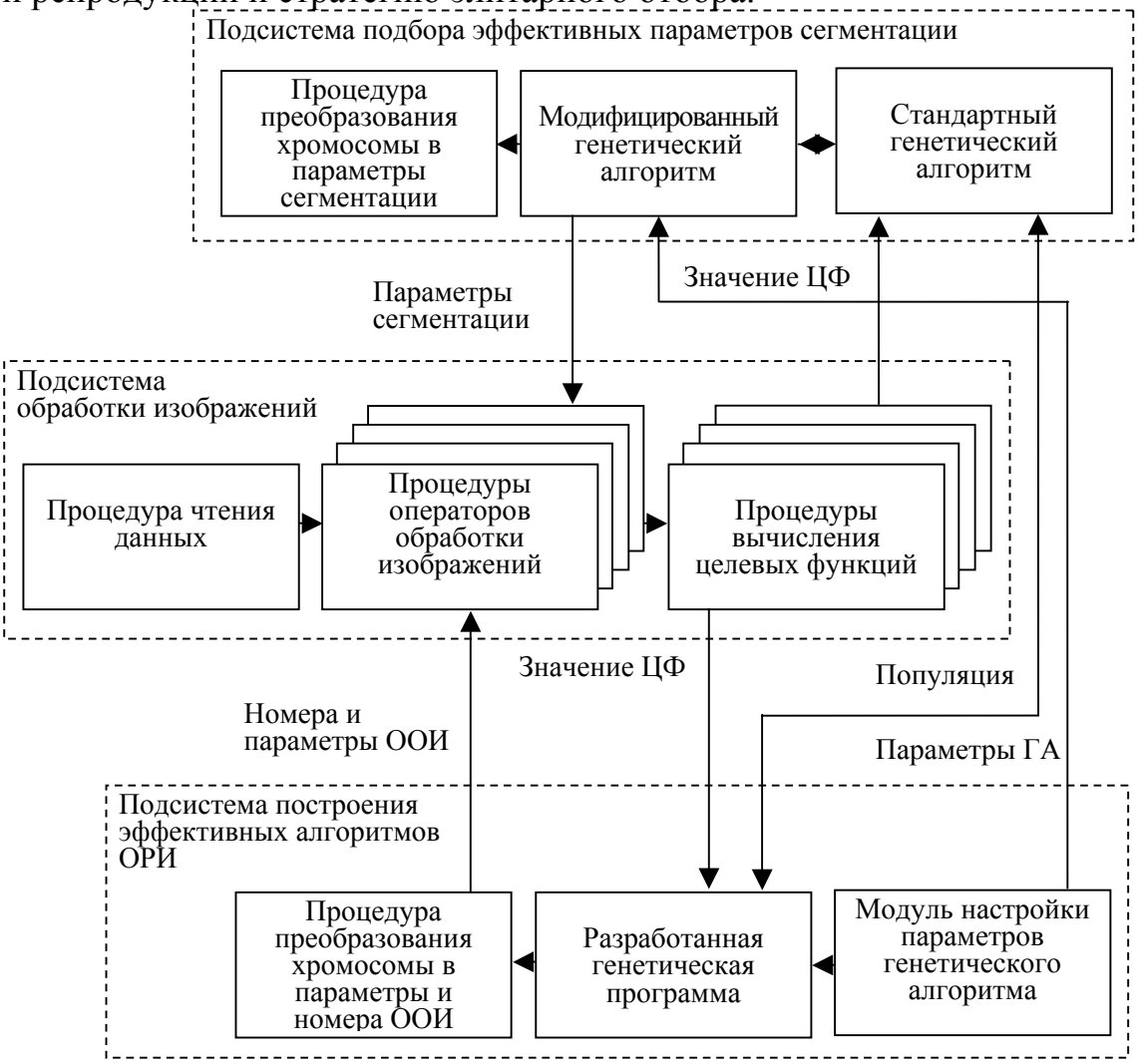


Рис. 1 Структура программной реализации системы инструментальных средств

«Подсистема построения эффективных алгоритмов обработки и распознавания изображений гистологических срезов» позволяет определить эффективную последовательность операций обработки и распознавания изображений с соответствующими рациональными параметрами. Результатом работы подсистемы являются номера операторов обработки изображений, из заранее определенного набора, значения их параметров, а так же результирующая плоскость. В случае, если достоверность результатов, представленных на выходной плоскости

изображения, обработанной с помощью построенного алгоритма, утверждена ЛПР, то полученные значения записываются в БД.

Модифицированная эволюционная программа осуществляет операторы мутации узлов программного графа и кроссинговера на основе значения коэффициента связности.

Разработан модуль, в котором имеется возможность задавать параметры ГА для повышения эффективности алгоритмов обработки и распознавания изображений гистологических срезов:

— выбор типа операторов мутации и кроссинговера, а также значения вероятности их выполнения;
— критерий останова алгоритма;
— выбор целевой функции (f_A, f_B, f_C или f_D);
— начальные параметры генетического алгоритма (мощность популяции, количество поколений, стратегия отбора, длина хромосомы).

База данных предназначена для хранения информации о последовательности операторов обработки изображений с соответствующими параметрами для каждого вида гистологического исследования и состоит из 4 таблиц формата Paradox 7.0. База данных включает следующие таблицы в виде файлов с расширением *.DB:

— справочник видов гистологических исследований (HistKind);
— данные по операциям обработки и распознавания изображений для вида гистологического исследования (IPO_Alg);
— набор входных плоскостей для каждого алгоритма (R_Planes);
— набор значений параметров операции обработки изображений (IPO_Param);
— виды параметров операций обработки изображений (Param);

Предложенный способ реализации имеет модульную структуру, состоящую из подсистем «Подсистемы подбора эффективных параметров сегментации» и «Подсистемы построения эффективных алгоритмов обработки и распознавания изображений гистологических срезов», которые могут использоваться независимо друг от друга, а так же могут применяться независимо от объекта и быть применены к различным изображениям без перестройки внутренней структуры.

Экспериментальные исследования разработанной системы инструментальных средств. Для проведения экспериментальных исследований и апробации, системы инструментальных средств обработки и распознавания цветных изображений гистологических срезов в качестве объекта выбран набор изображений срезов,

полученный на базе патолого-анатомического отделения «Донецкого областного клинического территориального медицинского отделения (ДОКТМО)».

Проведена серия экспериментов для определения эффективности целевых функций (рис. 2) и рациональных параметров эволюционного процесса (таблица 1).

Таблица 1

Основные результаты экспериментальных исследований

Параметр		Целевая функция			
Наименование	Обозначение	f_A	f_B	f_C	f_D
Глубина программного графа	md	17	15	7	25
Мощность популяции	S_{POP}	75	75	100	50
Количество поколений	C_G	7	7	7	8
Вероятность мутации	P_M	0,07	0,07	0,07	0,09
Вероятность кроссинговера	P_C	0,25	0,5	1	0,75

Также была проведена серия экспериментов, которые подтверждают эффективность модификации целевой функции (рис. 2)

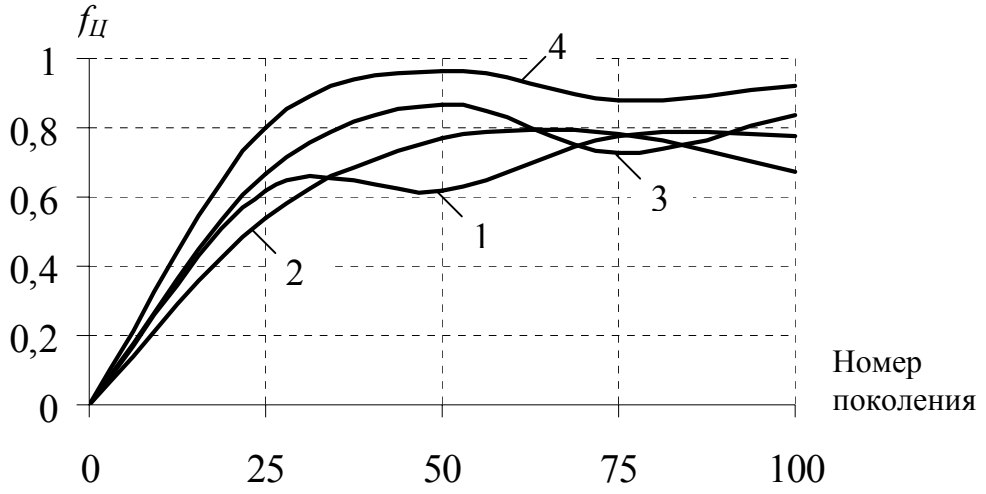
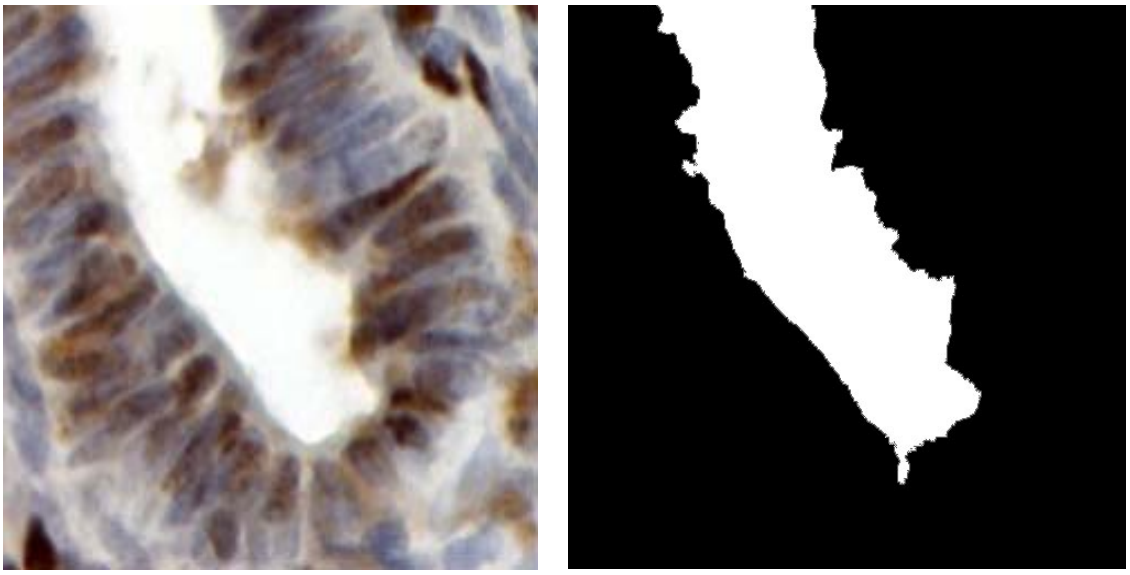


Рис. 2 Сравнительный анализ эффективности целевых функций: 1 – f_A; 2- f_B; 3 – f_C; 4 – f_D.

Сравнительный анализ целевых функций показал, что разработанная функция повышает эффективность построенных алгоритмов обработки и распознавания изображений гистологических срезов на 9%.

Результаты работы системы инструментальных средств обработки и распознавания изображений гистологических срезов приведены на рис. 3.

Выводы. Разработан и реализован новый подход в построении системы инструментальных средств обработки и распознавания изображений гистологических срезов на основе эволюционных моделей. В качестве целевой функции для отбора лучших решений предложено использовать оценку систематической ошибки поиска в зависимости от чувствительности и от специфичности, учитывающей глубину программного графа. Экспериментально доказана эффективность разработанных инструментальных средств в диагностике раковых заболеваний.



а) б)

Рис. 3 Результаты эволюционных вычислений (f_C=0,9214)
а) исходное изображение железы
б) обработанное изображение

Список использованной литературы

1. Скобцов Ю.А., Мартыненко Т.В. Построение эффективных алгоритмов обработки и распознавания изображений гистологических срезов на основе эволюционных моделей (статья в печати).

2. Poli R. Genetic programming for image analysis. In J. R. Koza, D. E. Goldberg, D. B. Fogel, and R. L. Riolo, editors, Genetic Programming 1996: Proceedings of the First Annual Conference, Stanford University, CA, USA, MIT Press .1996. – C.363–368.

3. Yoda I., Yamamoto K. and Yamada H. Automatic acquisition of hierarchical mathematical morphology procedures by genetic algorithms. Image and Vision Computing, 17(10):749–760, 1999.

Получено 01.06.07

УДК 622.822: 681.3

ЛОГИКО–ЛИНГВИСТИЧЕСКОЕ МОДЕЛИРОВАНИЕ АВАРИЙНЫХ ПРОЦЕССОВ В ВЕНТИЛЯЦИОННЫХ СИСТЕМАХ ШАХТ

В.В. Слесарев, А.Н. Коваленко, А.М. Алексеев, НГУ,
Днепропетровск, Украина

Візуалізація аварійних процесів на моделях вентиляційних систем шахт. Ці явища описуються реченнями російської мови, які формуються і зберігаються в спеціальних базах знань. Рішення при ліквідації наслідків аварії визначаються шляхом перетворення речень по визначеним правилам для кожного проблемного середовища існування автоматизованої системи.

В настоящее время на шахтах при составлении и вводе в действие планов ликвидации аварий (ПЛА) необоснованно мало используется процесс визуализации аварийных процессов на сетевых моделях вентиляционных систем шахт. В работах [1,2,3] была впервые научно обоснована возможность эффективного использования лингвистических переменных в сочетании с нечёткой логикой для описания аварийных процессов и выработки решений при управлении вентиляционными системами шахт. Это дало возможность применить методы ситуационного анализа и управления при разработке и вводе в действие ПЛА, а также работать в направлении создания экспертных систем управления безопасностью на шахтах и рудниках. Такая система в настоящее время разрабатывается на шахте «Западно – Донбасская» объединения «Павлоградуголь». Она включает в свой состав три основные автоматизированные компьютерные подсистемы: управления вентиляционной системой в нормальных и аварийных условиях эксплуатации шахты; составления и ввода в действие ПЛА; принятия решений при ликвидации последствий аварий на шахте.

Автоматизированная подсистема составления ПЛА производит выбор мероприятий в соответствии с заданными позициями и выдаёт распечатку оперативной части ПЛА в установленной форме [4]. Несмотря на то, что вышеуказанная подсистема даёт большой выигрыш времени при определении списка мероприятий по ликвидации аварий, следует отметить, что она формирует оперативную часть ПЛА только для заранее заданных позиций. Выбор

соответствующих оперативных мероприятий для ликвидации последствий аварий произвольного вида в непредусмотренном месте шахты, указанной выше подсистемой в настоящее время, не осуществляется. Сложность формализации процессов составления ПЛА и принятия решений при оперативной ликвидации возникших аварий в шахте заключается в необходимости отображения действий человека – оператора (руководителя аварийных работ), который должен принимать как количественные, так и качественные (на уровне лингвистических переменных) решения по заранее установленным правилам. Эта модель знаний хранит: описания аварийных ситуаций; процедуры конкретизации или обобщения этих ситуаций; процедуры их адаптации реальным условиям, сложившимся на объекте управления; схемы принятия решений по ликвидации последствий аварий. Классы ситуаций строятся экспертно на основе базы правил по составлению ПЛА [4,5]. После того как классы построены, каждая конфликтная ситуация должна быть распознана, т. е. признаки конфликтных ситуаций сравниваются с признаками классов, пока не совпадут с признаками одного из них. Если такой класс найден и ему соответствует единственное решение, то оно запоминается и переводится с реляционного языка на обычный разговорный. После этого генерируется текст позиций ПЛА или готовые решения по ликвидации последствий аварий. Все эти, выше рассмотренные, подсистемы связаны единой базой данных и знаний о шахте. Они функционируют в едином информационном пространстве и включают в свой состав сведения обо всех технологических процессах на шахте. В частности, речь идёт о системе вентиляции горных выработок, системе добычи угля, транспорта, энергообеспечения, противопожарной системе и т. п. Эта база данных (БД) и знаний формируется в процессе эксплуатации шахты, а также на основании проигрывании аварийных ситуаций на гибкой сетевой модели её вентиляционной системы и логико – лингвистических моделях (ЛЛМ) других технологических процессов. Составление ПЛА и принятие решений при ликвидации последствий аварий производится с наиболее большим возможным использованием наряду с количественной и качественной (на уровне лингвистических переменных) информации. Семантический анализ языка описания оперативной части ПЛА и соответствующих нормативных документов позволил установить 532 понятия и около 50 отношений между ними. Были установлены и основные правила преобразования первоначальной лингвистической информации с целью получения новых знаний о шахте в аварийной обстановке. Это даёт основание

для утверждения о разработке первого варианта языка представления знаний (ЯПЗ) о шахте в аварийной обстановке.

Основные ЛЛМ, входящие в состав разрабатываемой системы.

1. Семантическая модель распространения продуктов горения, газов и температурных полей в подземных выработках шахты.

Основные понятия: «газы», «дым», «концентрация», «интенсивность», «температура», «большая», «средняя», «малая», «горная выработка». Понятия, обладающие качеством, при их визуализации имеют свою окраску.

Основные отношения: «обладания качеством», «места», «временные», «пространства», «наименование», «И», «ИЛИ», «ЕСЛИ – ТО».

2. Маршруты эвакуации людей в безопасные места или на поверхность.

Основные понятия: «горная выработка», «наименование», «от», «до», «в», «на».

Основные отношения: «места», «временные», «пространства», «И», «ИЛИ», «ЕСЛИ – ТО».

3. Маршруты движения отделений ВГСЧ.

Основные понятия: «горная выработка», «наименование», «от», «до», «в», «на».

Основные отношения: «места», «временные», «пространства», «И», «ИЛИ», «ЕСЛИ – ТО».

4. Управляемость расходами воздуха в каждой горной выработке (или её участке) в нормальных и аварийных условиях эксплуатации.

Основные понятия: «расход воздуха», «значительный отрицательный»,…,«значительный положительный», «открыт», «закрыт», «остановлен», «реверсирован».

В зависимости от величины расхода воздуха участок горной выработки окрашивается в различные оттенки синего или красного цвета.

Основные отношения: «имеет значение», «состояния», «горная выработка», «наименование», «И», «ИЛИ», «ЕСЛИ – ТО».

Все эти процессы представляются на сетевых моделях вентиляционных систем шахт. Они имеют свою строго определённую окраску. Динамика этих процессов отображается на экранах дисплеев в реальном масштабе времени. В случаях использования системы в экстремальных ситуациях моделирование аварийных процессов производится в ускоренном варианте. Ключевым моментом данного проекта является создание структурной (подчеркивающей

организацию системы) модели вентиляционной системы шахты. Среди всех способов практической реализации наглядного моделирования были выбраны графы, поскольку они являются удобным языком для визуализации и эффективным инструментом решения задач, относящихся к весьма широкому кругу проблем. Наглядное моделирование предусматривает рассмотрение сложных информационных объектов и процессов, характеризующихся высокой степенью абстракции. Использование моделей в виде графов позволяет представить эти объекты и процессы в доступной для понимания наглядной форме. В нашем случае в качестве рёбер графа выступают отдельные участки горных выработок, а в качестве узлов – их пересечения. Каждое ребро будет иметь свою длину (длину участка горной выработки) – то есть граф будет взвешенным. Такой подход позволяет очень просто и быстро образовать модель на основе реального объекта и, в то же время, не создаёт особых трудностей при организации хранения и редактирования данных на ЭВМ. Для этого используется сервер баз данных Firebird. Структура графа представляется всего в виде 2 таблиц:

- список вершин графа – здесь хранятся номера всех вершин
- список ребёр графа – здесь хранятся номера двух вершин, которые соединяет данное ребро, а также длина, поперечное сечение, наименование горной выработки и другие характеристики.

Используя несложный пользовательский интерфейс и средства СУБД Firebird можно легко изменять структуру графа, создавая, редактируя и удаляя вершины или рёбра из соответствующих таблиц.

Однако, большие объемы данных, необходимые при моделировании вентиляционной системы, тяжело проанализировать, не прибегая к их визуализации. Известно, что человек лучше всего понимает и проникает в суть исследуемого объекта, когда он может "погрузиться в мир исследуемого объекта ", то есть в пространство модели, и когда его "погружение" усиливается возможностью непосредственно манипулировать данными в пространстве модели. Поэтому в качестве метода визуализации было выбрано создание полноценной 3-х мерной модели объекта. Для реализации этого используется OpenGL - одно из самых популярных прикладных программных интерфейсов (API – Application Programming Interface) для разработки приложений в области двумерной и трехмерной графики.

Для того чтобы обеспечить 3-х мерное представление графа, в таблицу вершин добавлены поля для хранения координат X, Y, Z. Рассмотрим теперь алгоритм построения дерева: - извлекаем все данные о вершинах и рёбрах из БД; -для отображения вершин используем функцию библиотеки GLU - gluSphere(); -для отображения рёбер используем функцию из той же библиотеки gluCylinder (). Но, поскольку для рёбер мы храним только номера вершин, которые они соединяет, у нас нет возможности напрямую рисовать цилиндры. Для этого была реализована функция, позволяющая рисовать ребро графа по координатам вершин, которые он связывает. Также эта функция отображает дополнительные характеристики ветви – наименование горной выработки, направление воздушной струи, аэродинамическое сопротивление и другие параметры. Далее необходимо обеспечить удобный просмотр построенной модели графа. Для этого используется стандартный набор функций OpenGL для модельно-видового преобразования glTranslate(), glRotate(), glScale(). Управление текущей позицией просмотра осуществляется с помощью клавиатуры и мыши.

Для того чтобы пользователь имел возможность изменять структуру модели, необходимо реализовать механизм выделения отдельных объектов. Основная масса способов решения этой задачи основана на вычислении, так называемого, «селектирующего» отрезка (СО), проходящего через точку наблюдения и координаты курсора мыши в мировой системе координат (МСК). После чего, все объекты сцены тестируются на пересечение с этим отрезком и в зависимости от результата выделяются (например, окрашиваются особым цветом). Но в данном проекте было решено использовать другой метод – основанный на буфере цвета. Принцип метода состоит в том, что каждому объекту в сцене присваивается уникальный цвет, которым он собственно и отображается, и по щелчку мыши определяется цвет пикселя, находящегося под курсором. Так как каждому цвету соответствует единственный объект – то его идентификация не вызывает особых трудностей. После выделения, в панели свойств появляется список всех доступных для редактирования характеристик выбранного объекта. Все произведённые изменения сразу же сохраняются в БД.

Разработанные методы визуализации аварийных процессов на экранах дисплеев компьютеров позволили более оперативно и точно вырабатывать оперативные мероприятия для позиций ПЛА, что

особенно важно в экстремальных ситуациях – ликвидации последствий аварий.

Список литературы

1. Коваленко А. Н. Обоснование представления параметров систем управления воздухораспределением посредством лингвистических переменных./ И – т геотехн. мех. АН УССР, - Днепропетровск, 1983. – 21 с. – Деп. в ВИНИТИ 26. 01. 84., №890.

2. Кимельман Э.А., Коваленко А.Н. Диалоговая автоматизированная система принятия решений при управлении воздухораспределением в подземных выработках шахт // Изв. вузов. Горный журнал. – 1987. -№7,С. 100 – 105.

3. Потёмкин В.Я.,Коваленко А.Н., Кимельман Э.А. Принятие решений при управлении воздухораспределением в подземных выработках шахт // Изв. вузов. Горный журнал. – 1987. - №6, С. 114 – 118.

4. Единые правила безопасности при разработке рудных, нерудных и россыпных месторождений подземным способом. Издание второе, переработанное и дополненное. М., «Недра», 1977, 223 С.

5. Устав ВГСЧ по организации и ведению горноспасательных работ. М., «Недра», 1971, 196 С.

Работа поступила в редакцию 18.05.2007.

УДК 531.38

ИССЛЕДОВАНИЕ ТЕЧЕНИЯ КРОВИ В РАЗВЕТВЛЕНИЯХ КРОВЕНОСНЫХ СОСУДОВ СО СТЕНОЗОМ

Скобцов Ю.А., д.т.н., проф., кафедра АСУ, Донецкий национальный технический университет; Родин Ю.А., к.м.н., в.н.с. Институт Неотложной и восстановительной хирургии АМН Украины; Озерко В.С., асс.Донецкий национальный университет.

Актуальность. В структуре смертности населения Украины сердечно-сосудистые заболевания занимают печальное первое место и их численность повышается год от года [1-3]. Удельный вес ишемических нарушений мозгового кровообращения составляет более 20%. Высокий уровень стойкой инвалидизации больных, перенесших инсульт, низкий процент восстановления трудоспособности (60% и 10% соответственно) побуждает рассматривать эту проблему как общегосударственную [2]. По словам Гуреш *Т.* [2] осложняет ситуацию также то, что по сравнению со странами мира в Украине длительное время происходит устоявшийся процесс неуклонного уменьшения количества населения. Его численность сократилась с 52,2 млн. чел. в 1993 году до 48,5 млн. в 2002 году. Ученые прогнозируют, что до 2026 года население Украины уменьшится до 42 млн. человек. Цереброваскулярная патология играет в этом не последнюю роль.

В последнее время все больше внимания уделяется проблеме профилактики ишемического инсульта. Стратегическая задача - на ранних стадиях выявить заболевание и определить тактику последующего лечения, в том числе и хирургического. Одним из эффективных направлений ее решения можно считать подход, применяемый в ряде стран, имеющих положительный опыт в этой области – это ежегодное (2 раза в год) обследование всего населения, достигшего возраста 40 лет с применением ультразвуковых методик. Результаты обследований показывают, что примерно у 14,8% мужского населения выявляются цереброваскулярные заболевания различной этиологии.

Донецкая область, наиболее экологически небезопасный промышленный регион нашей страны, занимает печально лидирующее место по этой патологии. По независимой оценке за 2006 год в нашей области произошло более 15 тыс. новых инсультов, умерло от последствий инсультов более 14 тысяч человек, что превышает количество заболевших и умерших от онкопатологии.

Цель исследования.

Оценить прогностическую эффективность системы ранней ультразвуковой диагностики больных хронической сосудистой мозговой недостаточностью (ХСМН).

Возникающая стенозирование бифуркации сонныхсосудов ведет к нарушению кровоснабжения мозга. Проблемам расчета течений крови посвящены многие работы, в частности, авторов [4,5] данной статьи. В данной работе исследовано влияние степени стеноза кровеносного сосуда на изменение расхода жидкости в пораженном разветвлении сосудов.

Расчетную область определим следующим образом: к искривленному участку исследуемого сосуда примыкают прямолинейные входной и выходной участки, необходимые для стабилизации потока и последующей постановке нейтральных граничных условий на выходе расчетной область. Поток крови будем считать несжимаемой жидкостью, стенки жесткими.

Система расчетных уравнений имеет вид:

$$\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial y}+v\frac{\partial u}{\partial z}+w\frac{\partial u}{\partial z}=-\frac{\partial P}{\partial x}+\frac{1}{\mathrm{Re}}(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}+\frac{\partial^2 u}{\partial z^2}), \quad (1)$$

$$\frac{\partial v}{\partial t}+u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial z}+w\frac{\partial v}{\partial z}=-\frac{\partial P}{\partial y}+\frac{1}{\mathrm{Re}}(\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}+\frac{\partial^2 v}{\partial z^2}) \quad (2)$$

$$\frac{\partial w}{\partial t}+u\frac{\partial w}{\partial y}+v\frac{\partial w}{\partial z}+w\frac{\partial w}{\partial z}=-\frac{\partial P}{\partial z}+\frac{1}{\mathrm{Re}}(\frac{\partial^2 w}{\partial x^2}+\frac{\partial^2 w}{\partial y^2}+\frac{\partial^2 w}{\partial z^2}) \quad (3)$$

$$\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}+\frac{\partial w}{\partial z}=0 \quad (4)$$

Система содержит следующий безразмерный параметр:

$\mathrm{Re}=\frac{U\cdot D}{\nu}$ - число Рейнольдса. В данной работе $Re=1000$.

В качестве начальных данных задается исходное поле скоростей и давления. В качестве начального поля скоростей выбран профиль Пуазейля.

Начальный профиль скорости зададим следующим образом:

$u(0,x,y,z)=u_H*(1-r^2)$, $v(0,x,y,z)=0$, $w(0,x,y,z)=0$.

Для давления за начальное распределение примем гидростатическое. Следовательно, рассматриваемое избыточное давление P будет равно нулю, т.е. P(0,x,y,z)=0. Метод численной реализации и проверки устойчивости изложен в работах [4-8].

На основании ультразвуковых исследований, пример которых представлен на рис.1, получены экспериментальные данные, включающие степень стеноза и конфигурацию сосудов, которые использованы при построении модели.

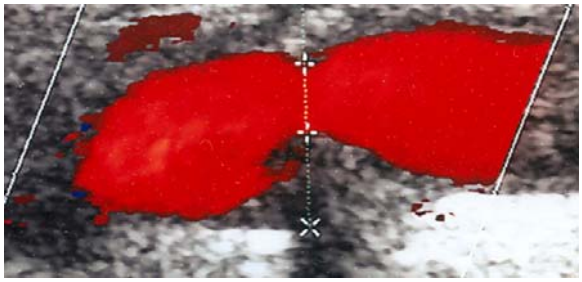


Рис.1. УЗ изображение сонной артерии с атеросклеротической бляшкой

Разработаны математические модели движения жидкости в пораженных разветвляющихся кровеносных сосудах, в основе которых лежит численное решение системы уравнений Навье-Стокса и уравнения неразрывности. Исследовано течение потока крови при наличии атеросклеротических бляшек в разветвлении. Получены распределения полей скоростей и давления для различных степеней стеноза и конфигураций и выполнена их визуализация.

На рис. 2 показано поле скорости и линии тока при начальном уровне развития стеноза. В областях прилегающих к внешним поверхностям развилки начинают формироваться отрывные зоны. С увеличением стеноза процесс вихреобразования происходит более интенсивно.

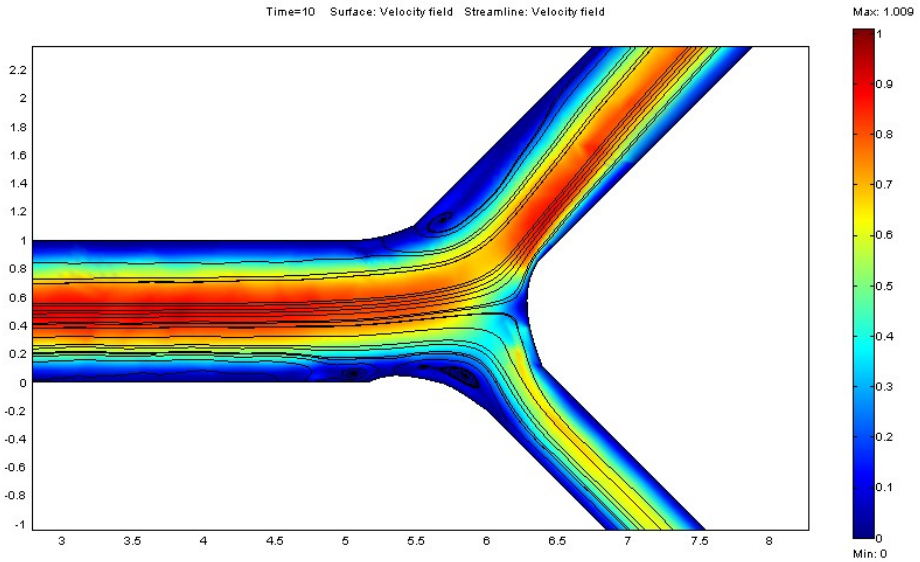


Рис. 2. Линии тока и поле скорости при небольшом уровне стеноза

На рис.3-4 представлены распределения скоростей и давления при увеличенном и большом уровнях стеноза.

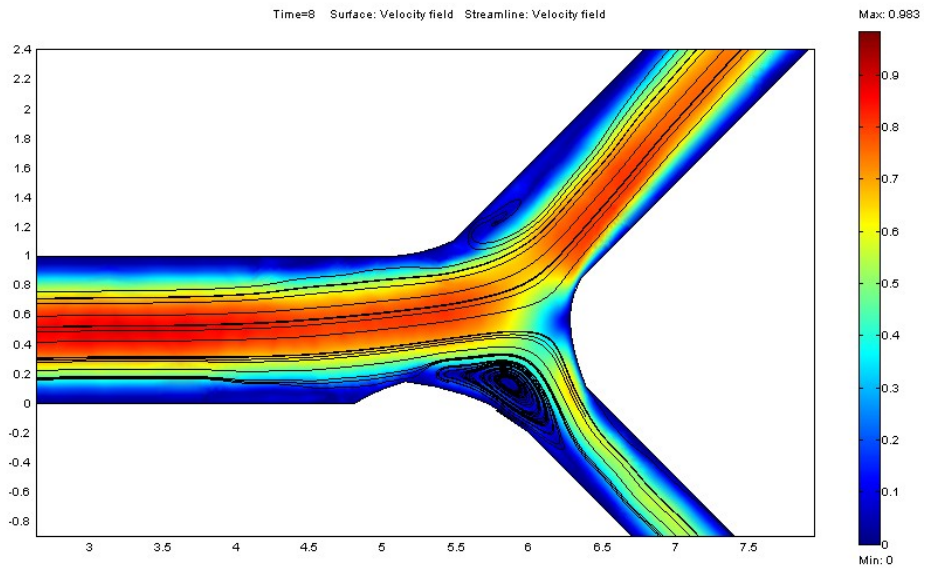


Рис. 3. Линии тока и поле скорости при увеличенном уровне стеноза

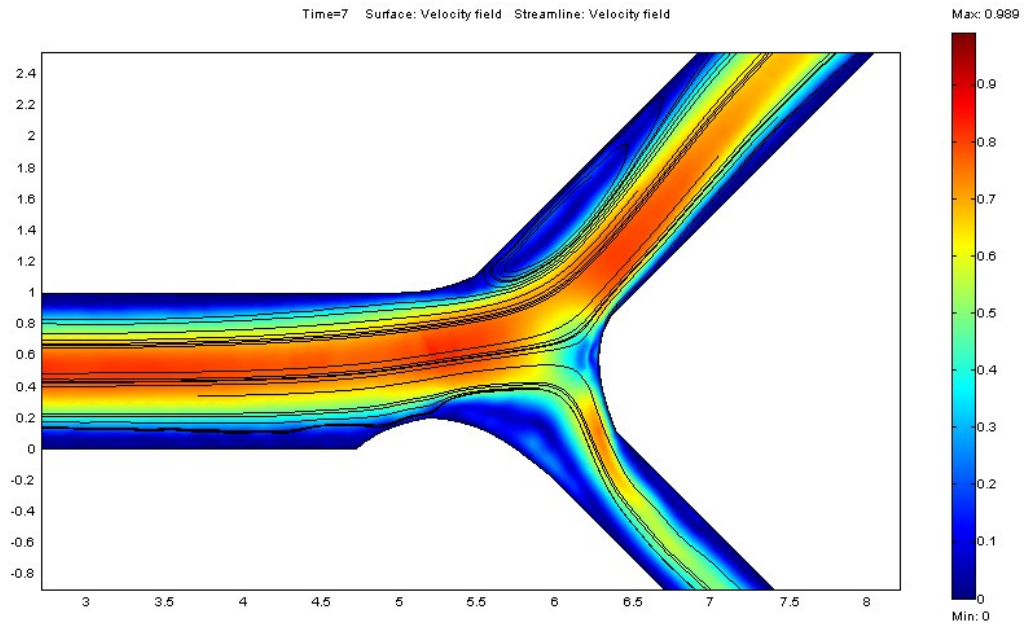


Рис. 4. Линии тока и поле скорости при большом уровне стеноза

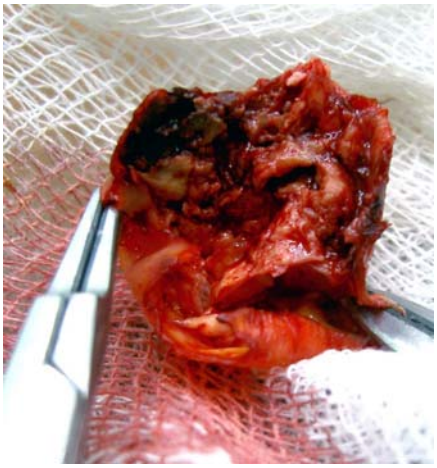


Рис. 5. Данные послеоперационного исследования

На рис.5 показан интраоперационный материал блокировки кровотока в пораженном сосуде. Из анализа расчетных данных видно, что с увеличение уровня стеноза значительно увеличивается отрывная зона в ветви непосредственно прилегающей к атеросклеротической бляшке. Как следствие этого уменьшается проходное сечение и локальное гидродинамическое сопротивление, в результате происходит блокировка кровотока в данной ветви. Кроме того, в верхней ветви тоже происходит увеличение отрывной зоны, хотя и не столь интенсивное, как в нижней. Это связано с тем, что изменяется угол входа потока в ветвь.Следовательно, возникновение стеноза приводит не только к блокировке кровоточа в пораженной ветви, но и к ухудшению кровотока в смежном ответвлении.

Выводы.

1. Сочетание ультразвуковых исследований и математического моделирования позволяет выставлять показания к оперативному лечению сонных артерий до развития острых неврологических событий.

2. Имеется четкая корреляция клинической картины и блокирования кровотока вычисленной математическими методами .

Литература

1. Белов Ю.В.,Базылев В.В.,Степаненко А.Б.Отдаленные результаты хирургического лечения больных с асимптомным стенозом сонных артерий //Хирургия .-2002.-№. 5.-С/4-6.
2. Гуреш Т. Нема кого послати у Декрет!: Демографічна криза в Україні і в світі // Укр. слово. - 2000. - 18-24 трав. - С. 15.

3. Ballotta Enzo, Da Giau Giuseppe, , Piccoli ` Antonio, , Baracchini Claudio. Durability of carotid endarterectomy for treatment of symptomatic and asymptomatic stenoses ? //JVS.-2004.-V.40.-N.-2.-P.270-278
4. Ю.А.Скобцов, Н.В.Финошин, В.С.Оверко, Ю.В.Родин. Исследование течения потока крови в атеросклеротически стенозированном кровеносном сосуде//Труды института прикладной математики и механики НАН УкраиныДонецк:2003,том.8.-с.135-141.
5. Скобцов Ю.А., Родин Ю.В., Озерко В.С. Исследование и визуализация поведения потоков крови в ветвящихся сосудах// Вестник Херсонского национального технического университета.- Херсон:2006.-№1 (24).-с.50-54.
6. Каро, Педли и др. Механика кровообращения. М, Мир.-1981, 624 с.
7. Э. Ричардсон. Динамика реальных жидкостей. М.: Мир, 1965.-328 с.
8. Белоцерковский О.М. Численное моделирование в механике сплошных сред. М, Физматгиз, 1994.- 245 с.

Получено 01.06.07

УДК 681.3

ОБЪЕКТНАЯ МОДЕЛЬ ЕСТЕСТВЕННО-ЯЗЫКОВОГО МЕДИЦИНСКОГО ТЕКСТА

Коломойцева И.А.
Донецкий национальный технический университет

Статья описывает объектную модель естественно-языкового медицинского текста, место этой модели в семантическом анализе текста. Статья содержит примеры семантически значимых объектов и семантических отношений, которые можно встретить в естественно-языковом медицинском тексте. В статье указывается, что семантические объекты и отношения являются основой словарей, которые участвуют в семантическом разборе текста.

Введение. В последнее время все больше и больше срециалистов в различных областях обращаются при поиске информации к Интернету. Огромное количество разрозненной и во многих случаях повторяющейся информации требует автоматизированной обработки.

В настоящее время технологии полного и точного автоматического анализа произвольного текста пока не существует. Наименее разработанными являются модели и методы семантического уровня [1].

В целом стоит отметить, что компьютерная семантика только выходит из стадии поисковых и научно-исследовательских работ.

Области применения семантического анализа очень разнообразны. Одной из самых важных областей является переход от плохо структурированной (ЕЯ-текст) к хорошо структурированной информации, которую можно обработать стандартными и высокоэффективными средствами информационных технологий [1].

Именно плохая структурированность медицинского ЕЯ-текста существенно осложняет его обработку. А потребность анализа больших объемов текстологических данных есть. В первую очередь, это относится к современному, быстро развивающемуся разделу медицины – сравнительной медицине. В связи с этим построение алгоритма извлечения фактологической информации (семантического анализа) из медицинского ЕЯ-текста и построения БД является актуальной теоретической и практической задачей.

Семантический анализ ЕЯ-текста опирается на концептуальный и понятийные словари [1]. Понятийный словарь включает в себя

описание понятий (объектов), которые представлены в тексте. Назначение этого словаря – установить связь между понятиями (объектами) и конкретными словами, встреченными в тексте. Концептуальный словарь оперирует смыслами, описывает свойства и отношения понятий, то есть семантические связи. Стратегия наполнения этих словарей является актуальной научной задачей.

Целью работы является:

1) определение объектов, которые присутствуют в медицинских ЕЯ-текстах, могут являться субъектами семантических отношений и будут использоваться при создании семантических словарей;

2) определение семантических отношений (связей) для медицинского ЕЯ-текста с использованием выявленных объектов;

3) определение роли семантически значимых объектов и семантических отношений в структуре словарей, используемых для семантического анализа медицинского ЕЯ-текста.

Постановка задачи. Задачей данной работы является анализ текстов, полученных в результате поиска в сети Интернет. Эти тексты имеют следующую структуру:

– название болезни;
– общее описание болезни;
– симптомы и причины возникновения;
– методы лечения.

Результаты анализа медицинского ЕЯ-текста должны быть представлены в виде семантически значимых объектов и семантических отношений. Выявленные объекты и отношения станут основой понятийного и концептуального семантических словарей.

Общие положения семантического анализа. Ограничения, накладываемые на текст, по которому проводится семантический анализ [1]:

1) в тексте должен присутствовать смысл и этот смысл должен быть ярко выражен;

2) тексты должны опираться на логически и терминологически отработанную систему понятий;

3) тексты должны быть стилистически и лексически однородными;

4) тексты должны быть фактографическими, содержать описание свойств определенной совокупности объектов, отношений между ними, процессов и действий, в которых они участвуют.

Выход за пределы этих ограничений требует применения специальных методов, ориентированных на специфику решаемой задачи.

Система семантического анализа является тезаурусно (онтологически) ориентированной. Основная проблема в создании реально работающих анализаторов - это создание реально работающего понятийного словаря, то есть словаря, который:

1) обеспечивает требуемую алгоритмами функциональность;

2) обеспечивает удовлетворительное покрытие текстов.

Объекты и семантические отношения. Чтобы использовать естественный язык в качестве основы для построения языка представления знаний, в нем предлагается выделить несколько классов–элементов. Эти классы можно разделить на две категории: семантически значимые объекты предложения и семантические отношения. Объекты еще называют именами [2] и именованными сущностями [3]. Объекты, представленные в медицинских ЕЯ-текстах, описаны в таблице 1. Так как объект ЭЛЕМЕНТЫ из таблицы 1 подразумевает довольно много понятий, то в нем можно выделить ряд подобъектов, которые представлены в таблице 2.

Объекты связываются между собой с помощью семантических отношений. Выдвинута гипотеза, согласно которой множество отношений, в отличие от множеств объектов (имен), конечно [2]. В [2] выделено около 200 не сводимых к друг другу отношений. Остальные виды взаимосвязей между объектами, которые могут встретиться в естественно-языковом тексте, сводимы к этим базовым отношениям. В [4] 200 отношений из [2] сведены к семнадцати. В [2] и [4] тип связи определяется по тому, в каком падеже стоит объект, являющийся субъектом действия в предложении, и какой предлог предшествует этому объекту в предложении. Другой подход к определению семантических отношений предложен в [1]. В этом случае определено всего пять семантических отношений, которые связывают между собой не объекты, а семантические классы. Семантические классы, в свою очередь, являются совокупностью объектов. Более подробный обзор семантических отношений, определяемых для ЕЯ-текстов, представлен в [5, 6, 7] .

В медицинских естественно-языковых текстах можно выделить следующие семантические связи: генеративную, результативную, инструментальную, каузальную, комитативную [4].

Таблица 1. Объекты, представленные в медицинских ЕЯ-текстах

№ п/п	Название объекта	Примеры объектов
1	БОЛЕЗНЬ	Аллергия, атеросклероз, остеохондроз, диабет сахарный
2	КАЧЕСТВА_ЛЮДЕЙ	Возраст, профессия, занятия
3	НАРУШЕНИЯ	Нарушения работы организма человека: авитаминозы
4	ЭЛЕМЕНТЫ	Вещества, которые участвуют в жизнедеятельности человеческого организма
5	СОСТОЯНИЕ_ОРГАНИЗМА	Белковое голодание, витаминная недостаточность, гиподинамия, гиповитаминоз, улучшение состояния организма (лечение), азотное равновесие
6	НАУКА	Биохимия
7	СИСТЕМА_ПИТАНИЯ	Вегетарианство, диета, голодание
8	ДЕЙСТВИЕ_НА_ОРГАНИЗМ	Липотропное воздействие
9	ОБРАЗ_ЖИЗНИ	Сидячий, активный
10	МЕТОД_ЛЕЧЕНИЯ	Лекарство, операция, мануальная терапия
11	СИМПТОМЫ	Боли в органах

Таблица 2. Подобъекты объекта ЭЛЕМЕНТЫ

№ п/п	Название подобъекта	Примеры подобъектов
1	ВИТАМИНЫ	Водорастворимые, жирорастворимые, холин
2	ГОРМОНЫ	Инсулин
3	БЕЛКИ	
4	ЖИРЫ	Гидрированные, липиды
5	УГЛЕВОДЫ	Гликоген, глюкоза, дисахариды, моносахариды, полисахариды
6	ФЕРМЕНТЫ	
7	КИСЛОТЫ	Жирные кислоты (насыщенные, ненасыщенные, полинасыщенные), аминокислоты (метионин, цистин)
8	ВЕЩЕСТВА	Фитонциды, балластные вещества (клетчатка, пектин), холестерин)

Генеративная связь имеет место, когда один компонент обозначает лицо или предмет, принадлежащий некоторой совокупности, категории, обозначаемой вторым компонентом.

Результативная присутствует в тех предложениях, где один компонент выражает следствие действия второго.

Инструментальная означает, что один компонент обозначает орудие действия, обозначаемого другим компонентом.

Каузальная имеет место, когда один компонент обозначает причину появления другого компонента спустя какое-то время.

Комитативная встречается в тех предложениях, когда один компонент обозначает сопровождающее другой компонент действие, сопутствующий предмет, сопровождающее лицо.

Объекты медицинских ЕЯ-текстов, которые связываются семантическими отношениями, представлены в таблице 3.

Таблица 3. Семантические отношения и связываемые ими объекты

Семантическая связь	Связываемые объекты
Результативная	ОБРАЗ_ЖИЗНИ → БОЛЕЗНЬ ОБРАЗ_ЖИЗНИ → СОСТОЯНИЕ_ОРГАНИЗМА СИСТЕМА_ПИТАНИЯ → БОЛЕЗНЬ БОЛЕЗНЬ → СОСТОЯНИЕ_ОРГАНИЗМА БОЛЕЗНЬ → БОЛЕЗНЬ МЕТОД_ЛЕЧЕНИЯ → СОСТОЯНИЕ_ОРГАНИЗМА НАРУШЕНИЯ → СОСТОЯНИЕ_ОРГАНИЗМА НАРУШЕНИЯ → НАРУШЕНИЯ
Инструментальная	НАУКА → МЕТОД_ЛЕЧЕНИЯ НАУКА → СИСТЕМА_ПИТАНИЯ МЕТОД_ЛЕЧЕНИЯ → НАРУШЕНИЯ
Каузальная	ЭЛЕМЕНТЫ → СОСТОЯНИЕ_ОРГАНИЗМА ЭЛЕМЕНТЫ → БОЛЕЗНЬ СИСТЕМА_ПИТАНИЯ → ЭЛЕМЕНТЫ ОБРАЗ_ЖИЗНИ → БОЛЕЗНЬ КАЧЕСТВА_ЛЮДЕЙ → БОЛЕЗНЬ КАЧЕСТВА_ЛЮДЕЙ → СОСТОЯНИЕ_ОРГАНИЗМА ОБРАЗ_ЖИЗНИ → СОСТОЯНИЕ_ОРГАНИЗМА МЕТОД_ЛЕЧЕНИЯ → МЕТОД_ЛЕЧЕНИЯ
Комитативная	БОЛЕЗНЬ → СИМПТОМЫ БОЛЕЗНЬ → НАРУШЕНИЯ НАРУШЕНИЯ → НАРУШЕНИЯ

Выводы. Выделенные в медицинском ЕЯ-тексте объекты и отношения могут служить в качестве словарей для организации семантического разбора. Объекты будут частью словаря перевода, а отношения – концептуального словаря.

Отличие предложенного подхода к представлению медицинского ЕЯ-текста от онтологического состоит в том, что в тексте выделяются не только понятия (объекты), но и выполнена попытка установить связь между объектами в виде семантических связей. При дальнейшей

работе с семантическими связями планируется определить их свойства.

Так как разумно построенная система анализа должна обеспечивать не только извлечение знаний из конкретного текста, но и накопление результатов, как на синтаксическом, так и на семантическом уровне [1], то в качестве перспектив дальнейшей работы можно указать следующее:

1) создание большого корпуса медицинских ЕЯ-текстов с целью их анализа и уточнения количественного и качественного состава семантически нагруженных объектов;

2) определение необходимых и достаточных условий для установления типа семантической связи;

3) разработка системы пополнения правил семантического разбора.

Литература

1. Рубашкин В.Ш. Семантический компонент в системах понимания текста // Труды Десятой национальной конференции по искусственному интеллекту с международным участием (КИИ-2006). - М.: Физматлит. - 2006. - Т. 2. - С. 455-463.

2. Поспелов Д. А. Логико-лингвистические модели в системах управления. – М.: Энергоиздат, 1981. – 232 с.

3. Хорошевский В.Ф. Оценка систем извлечения информации из текстов на естественном языке: кто виноват, что делать // Труды Десятой национальной конференции по искусственному интеллекту с международным участием (КИИ-2006). - М.: Физматлит. - 2006. - Т. 2. - С. 464-478.

4. Осипов Г. С. Приобретение знаний интеллектуальными системами: Основы теории и технологии. – М.: Наука. Физматлит, 1997. – 112 с.

5. Коломойцева И.А. Особенности применения существующих теорий «понимания» текста на естественном языке к медицинским текстам //Научные труды Донецкого государственного технического университета. Серия: Проблемы моделирования и автоматизации проектирования динамических систем, выпуск 29. – Севастополь: «Вебер», 2001. – С. 94–99.

6. Grishman. Information extraction: Techniques and challenges // Maria Teresa Pazienza, editor. Information Extraction. Springer-Verlag, Lecture Notes in Artificial Intelligence, Rome, 1997. – P. 108-110.

7. Using a language independent domain model for multilingual information extraction. By: Azzam, Saliha; Humphreys, Kevin; Gaizauskas, Robert; Wilks, Yorick. Applied Artificial Intelligence, Oct 99, Vol. 13 Issue 7. – P. 705-724.

Работа поступила в редакцию 26.06.2007.

НОВЫЙ КЛАСС СИНТЕТИЧЕСКИХ ИНДИКАТОРОВ

Тихонова О.А.
Донецкий национальный технический университет

Предложен оригинальный алгоритм скользящего усреднения, в котором существенно снижены эффекты недостатков традиционных алгоритмов, считавшихся ранее неустранимыми.

I. Введение. При решении задач технического, экономического, социального и др. видов прогнозирования, исследователи пользуются алгоритмами, использующими скользящее усреднение. Применение скользящего усреднения позволяет выделить в исследуемых графиках тренды или другие характеристики и существенно подавить вредную случайную компоненту. В этом случае, используемые инструменты, обладают существенными недостатками, которые приводят к снижению результативности:

- временная задержка продуктов усреднения относительно элементов ценового графика на величину m/2, где m – величина временного окна усреднения;
- относительно высокая колеблемость продуктов усреднения, которая слабо зависит от величины m;
- влияние линейных частотных искажений как следствие существенной нелинейности амплитудно-частотных характеристик (АЧХ) алгоритмов скользящего усреднения [1];
- линеаризация нелинейных ценовых трендов, путем выделения этих трендов с определенным смещением [2].

II. Постановка задачи. Целью данной статьи является решение комплексной проблемы по одновременному устранению всех четырех недостатков современных индикаторов, использующих в своих алгоритмах скользящее усреднение. Данная цель может быть достигнута путем синтеза нового класса индикаторов, лишенных перечисленных выше недостатков (они названы синтетическими из-за искусственности их алгоритмов формирования по сравнению с традиционными).

Временная задержка продуктов усреднения ценового графика является следствием фазовых задержек всех спектральных компонент выделяемых трендов. Фазочастотная характеристика (ФЧХ) алгоритма усреднения обеспечивает фазовый сдвиг (временную задержку) каждой из множества спектральных составляющих, характеризующих выделенный тренд:

$$\tau_{ф}(f_н) = -\varphi(f_н)/ f_н ,$$

где: $\varphi(f_н)$ – ФЧХ алгоритма усреднения;

$f_н$ – нормированная частота;

$$f_н = \frac{f}{f_д},$$

$f_д$ – частота дискретизации.

Групповая задержка, объясняющая запаздывание выделенного тренда относительно его истинного положения на величину m/2, определяется выражением:

$$\tau_{гр} = -\frac{d\varphi(\overline{f_н})}{d\,\overline{f_н}},$$

где: $\overline{f_н}$ – среднее значение нормированной частоты спектральной плотности мощности выделенного тренда;

$\frac{d}{d\,\overline{f_н}}$ – первая производная ФЧХ алгоритма усреднения.

В этой ситуации целесообразно скомпенсировать $\tau_{гр}$, т.е. свести ее к нулю. В этом случае задержка величиной m/2 будет ликвидирована. К сожалению, полная компенсация $\tau_{гр}$ принципиально невозможна из-за особенностей дискретной фильтрации. Частичная компенсация ($\tau_{гр} \approx 0$) вполне реализуема на практике путем многократного усреднения ценового графика "назад – вперед". Таким образом, усложняя алгоритм усреднения (вводя дополнительную алгоритмическую избыточность) можно компенсировать временную задержку. Для исследования был взят алгоритм традиционного экспоненциального скользящего усреднения (EMA). Как оказалось, использование многократного усреднения с помощью EMA по алгоритму "назад – вперед" компенсирует не только $\tau_{гр}$, но и линеаризует его ФЧХ (это тот самый "специальный" случай о котором отмечено в [1]). В табл.1 приведен алгоритм получения новой скользящей средней для m=4. В табл.1 принято $\alpha = \frac{2}{\left(\frac{m}{2}\right)+1}$.

Это выражение отличается от традиционного поскольку двукратное усреднение увеличивает его совокупный эффект ровно в два раза. Далее скользящее усреднение осуществляется традиционно, т.е. отбрасывается C_1 и добавляется C_5 и процесс многократно повторяется.

Таблица 1. Алгоритм формирования синтетической скользящей средней.

C_i / Q_i	C_1	C_2	C_3	C_4
I	$Q_4=Q_3+\alpha(C_1-Q_3)$	$Q_3=Q_2+\alpha(C_2-Q_2)$	$Q_2=Q_1+\alpha(C_3-Q_1)$	$Q_1=C_4$
II	$Q_5=Q_4$	$Q_6=Q_5+\alpha(Q_3-Q_5)$	$Q_7=Q_6+\alpha(Q_2-Q_6)$	$Q_8=Q_7+\alpha(Q_1-Q_7)$

Таким образом, удается частично скомпенсировать временную задержку m/2 (устранить первый недостаток традиционного скользящего усреднения).

Любой алгоритм скользящего усреднения представляет собой цифровой фильтр нижних частот [1]. Поскольку полоса частот, которую занимает спектральная плотность мощности исследуемых цен закрытия всегда намного шире, чем полоса пропускания такого фильтра, то он является формирующим. Этот фильтр формирует выходной продукт фильтрации (усреднения) с определенными автокорреляционными свойствами, которые определяются его амплитудно-частотной характеристикой (АЧХ). Чем уже полоса пропускания такого фильтра (больше величина m), тем в большей степени усиливаются статистические взаимосвязи между все более и более отдаленными друг от друга значениями C_j. Это негативно влияет на качество усреднения. Для борьбы с этим явлением необходимо использовать алгоритм скользящего усреднения с минимальным значением m=2. Замена направлений усреднения способствует полной временнóй декорреляции продуктов на выходе фильтра при каждом проходе и одновременно увеличивает эффект усреднения (общее значение m суммируется при каждом проходе).

Нужно сказать, что и раньше исследователи делали попытки увеличить эффективность скользящего усреднения путем многократного усреднения его продуктов [4]. Однако это привело, во-первых, к увеличению запаздывания, а во-вторых, к проявлению вредного эффекта Слуцкого-Юла. Временнáя декорреляция, используемая в предложенном алгоритме, оказалась эффективным средством для борьбы с вредным эффектом Слуцкого-Юла. В предложенных индикаторах этот эффект принципиально не возможен.

Следовательно частично устраняется и второй негативный эффект традиционного скользящего усреднения.

И еще, в случае выделения существенно нелинейных трендов наблюдается вредный эффект смещения [2]. Он усиливается, если

длительность тренда соизмерима с величиной m. В случае малых значений m (в нашем случае m=2), эффект смещения минимизируется и линеаризация выделяемого нелинейного тренда так же минимальна.

Предложенный оригинальный метод усреднения способствует существенному снижению всех перечисленных выше недостатков алгоритмов усреднения. Конечно, естественно, что при этом о полной ликвидации перечисленных выше недостатков говорить нельзя.

Программа реализации на ЭВМ.

Предложенный в работе алгоритм был успешно реализован в популярной программе трейдеров TradeStation 2000i фирмы Omega Research на языке Easy Language.

III. Результаты. Ниже на рис. 1 приведены для сравнения фрагмент исследуемого ценового графика EUR/USD и традиционная скользящая средняя (EMA m=9), а так же новая скользящая средняя (m'=2, m=8, где m – величина временного окна усреднения). Здесь же показаны сигналы входов в рынок (Λ отмечены сигналы новой скользящей средней; ▲- отмечены традиционные сигналы).

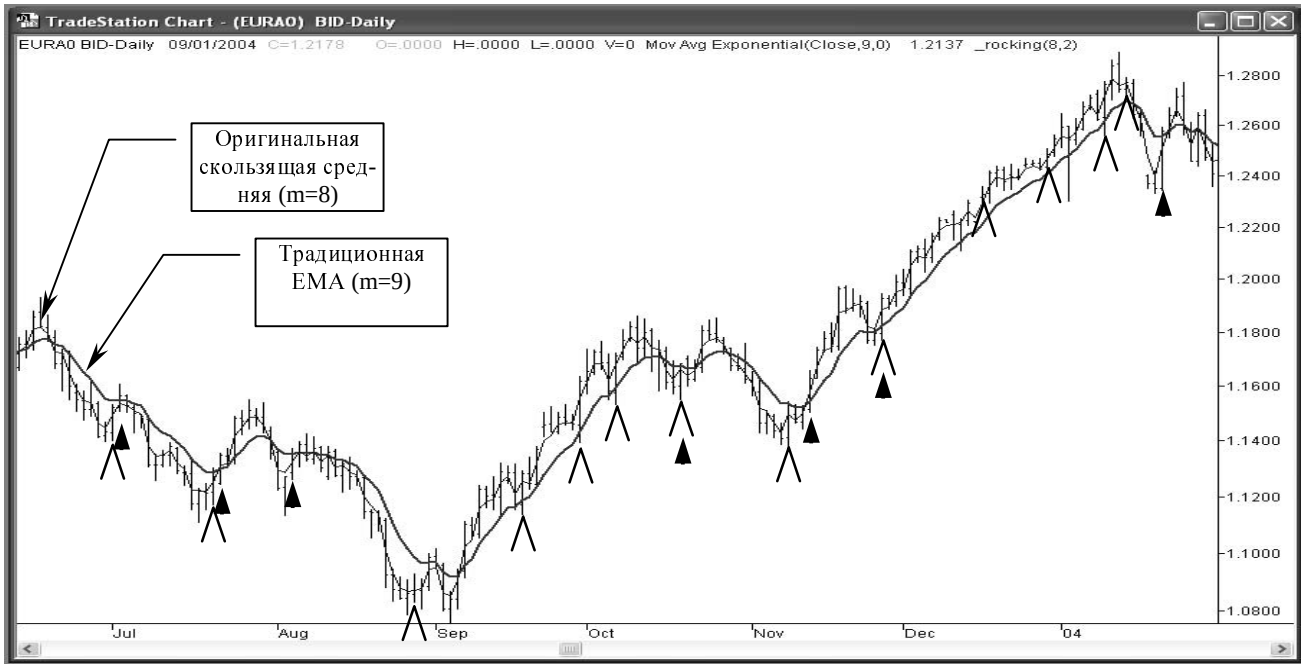


Рис. 1. Традиционная EMA и оригинальная скользящая средняя

На рис. 2 демонстрируется семейство предложенных новых скользящих средних, характеризующихся различным количеством проходов n. С увеличением величины окна усреднения m происходит рост числа проходов n. Это приводит к незначительному отставанию кривых от реальных трендов. Наблюдаемый эффект связан минималь-

ными линейными частотными искажениями и неполной компенсацией временной задержки.

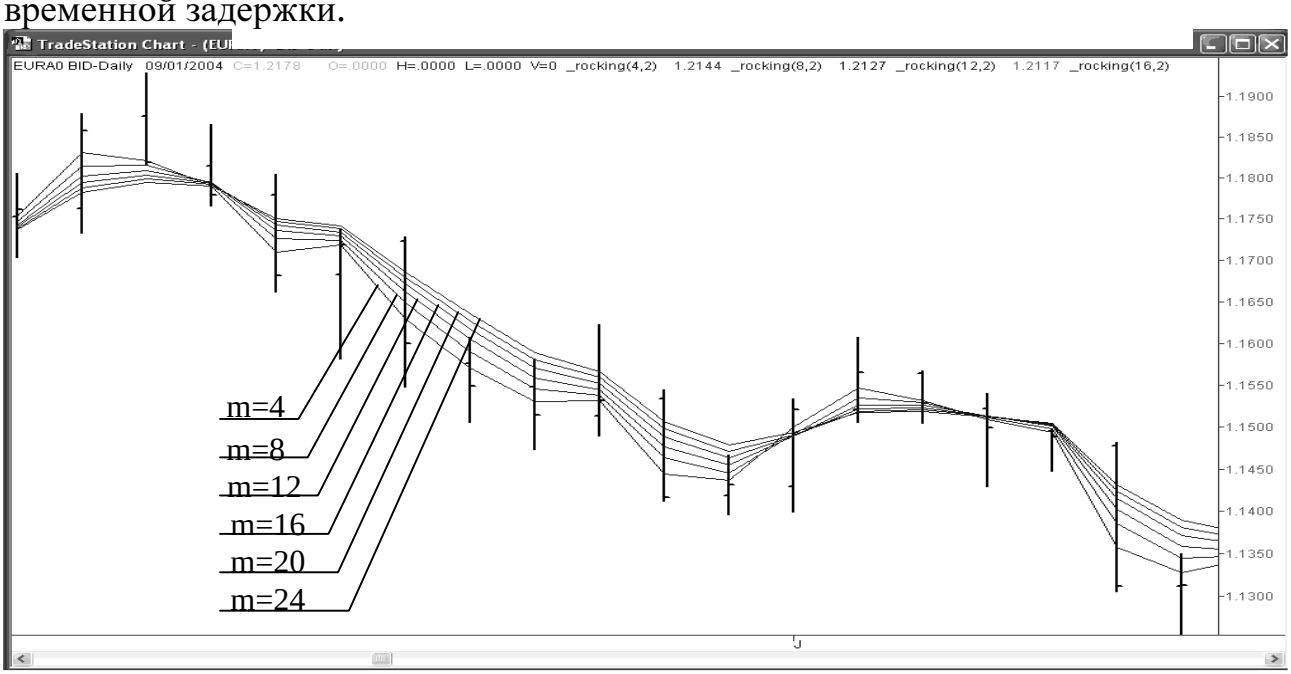


Рис. 2. Эффект многократного усреднения при реализации оригинальной скользящей средней

На рис. 3 показан фрагмент ценового графика рис. 1 и традиционные MACD, полученные с использованием алгоритма:

$$\begin{cases} A_i = EMA_i(13) - EMA_i(26); \\ EMA_i(A_i;9); \end{cases}$$

которые соответствуют "быстрой" и "медленной" MACD.

Для сравнения ниже построены графики MACD' (MACD_- на рисунке), полученные с применением предложенного алгоритма. К сожалению, нам не удалось достичь полной идентичности условий и второй алгоритм из-за особенностей предложенного метода выглядит как:

$$\begin{cases} A_i' = EMA_i'(12) - EMA_i'(24); \\ EMA_i'(A_i;8). \end{cases}$$

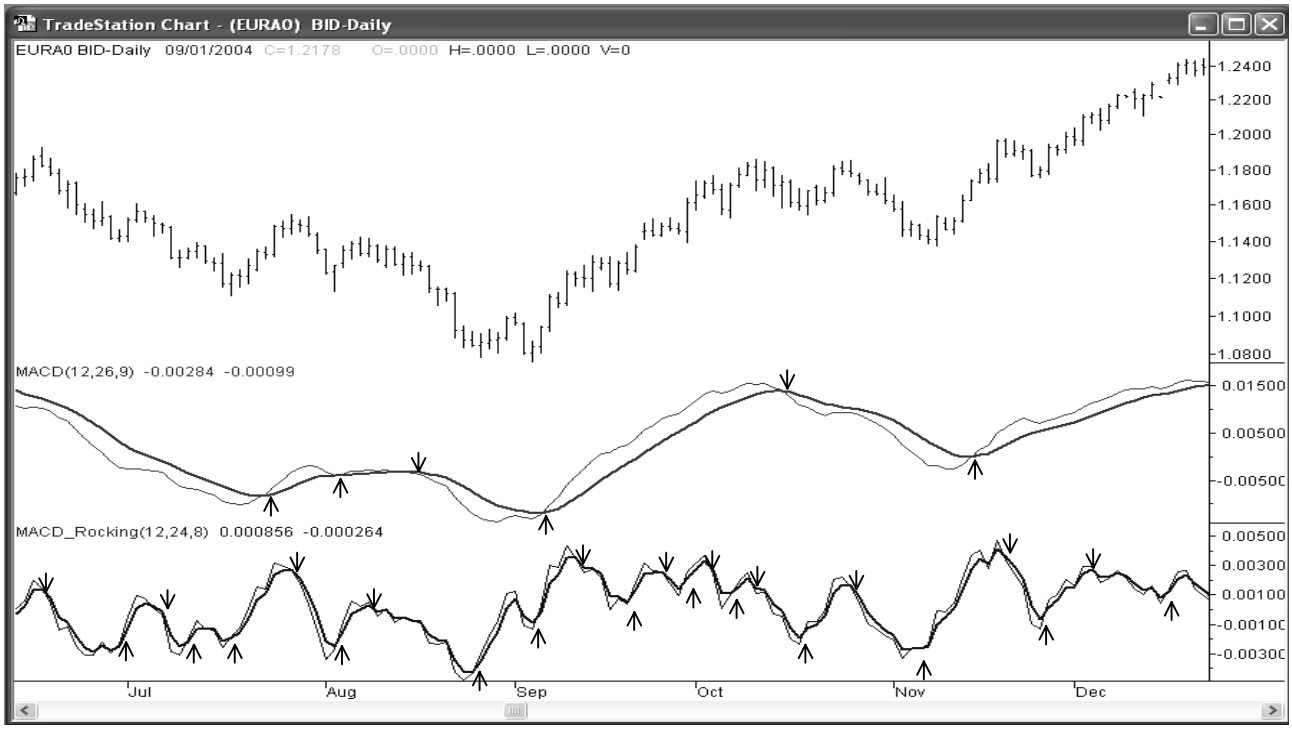


Рис. 3. Сравнения эффективности традиционного MACD и MACD'

Аналогичным образом можно модернизировать и другие популярные традиционные инструменты, введя в них в место обычных новые алгоритмы усреднения.

Исследования показали, что новые инструменты обладают следующими преимуществами по сравнению с традиционными:

- имеют минимальное запаздывание относительно ценового графика;
- обладают минимальной колеблемостью;
- являются по сравнению с традиционными более чувствительными;
- формируют большее количество управляющих сигналов.

IV. Выводы.

По результатам проведенных исследований можно сделать следующие выводы:

1. Временное запаздывание традиционных скользящих средних можно частично скомпенсировать путем усложнения традиционного алгоритма EMA, посредством введения многократного усреднения "назад-вперед".
2. В качестве исходного алгоритма усреднения для реализации нового класса синтетических инструментов целесообразно выбрать традиционный алгоритм формирования EMA при m'=2.

3. Предложенные оригинальные алгоритмы инструментов технического анализа рынков позволили минимизировать известные недостатки традиционных алгоритмов скользящего усреднения (временную задержку и высокую колеблемость их продуктов), которые ранее считались не устранимыми. И, кроме того, наше предположение позволило существенно уменьшить их недостатки, связанные с линейными частотными искажениями и с эффектом линеаризации выделяемых нелинейных трендов.

Вместе с тем, предложенные оригинальные индикаторы обладают рядом недостатков (возможность применения инструментов только кратности, равной четырем: m=4; 8; 12 и т.д.; существенные усложнения алгоритмов; возможность использования для усреднения только традиционного алгоритма EMA).

Литература:

1.Смирнов А., Тихонова О. Секрет совершенства индикаторов Марка Джурика раскрыт? // Валютный спекулянт. 2006, №1, с.32-35

2.Смирнов А., Михайлов С. Выбор типа скользящих средних // Валютный спекулянт. 2003, №7, с.50-55

3.Смирнов А., Гизатулин А. Новый метод сглаживания ценовых графиков // Валютный спекулянт. 2002, №12, с.38-40

4.www.jurikres.com.

5.Кендалл М., Стьюарт А. Многомерный статистический анализ и временные ряды. -М.: Наука, 1976. -736с.

Получено 01.06.07

УДК 681.3

МОДЕЛЬ СЕМАНТИЧЕСКОГО АНАЛИЗА МЕДИЦИНСКОГО ЕСТЕСТВЕННО-ЯЗЫКОВОГО ТЕКСТА

Коломойцева И.А., Юдина Ю.А.
Донецкий национальный технический университет

Введение. Технологии полного и точного автоматического анализа произвольного текста (и даже делового текста) пока не существует. При этом наименее разработанными являются модели и методы семантического уровня. В отличие от синтаксиса и морфологии семантический анализ не имеет прочной дисциплинарной традиции. В компьютерной семантике существует ряд сложных теоретических проблем, подходы к которым только нащупываются. Главные из них: стандартизация языков представления знаний; разрешение синтаксической и лексической омонимии; установление референциальных отношений между единицами текста; анализ контекстов, характеризующихся смысловой неполнотой; разработка семантических словарей, необходимых для поддержки алгоритмов семантического анализа. Нужно также иметь в виду, что для достаточно полного понимания текста от системы анализа помимо способности выявить и формализовать буквальную семантику текста требуется также способность к реализации логического вывода по тексту, - что представляет собой еще менее разработанную проблему. Можно сказать, что компьютерная семантика еще только выходит из стадии поисковых и научно-исследовательских работ [1], [4], [5].

Описание семантического анализатора. В статье предлагается подход, в основе которого лежит компьютерное толкование смысла слова на формальном семантическом языке и функциональное использование этого толкования при вычислении смысла предложения.

С точки зрения этого подхода любое слово русского языка является именем (названием) функции f(x1, …, xn), которая связывается с этим словом и называется его семантикой. Каждое свое конкретное значение слово получает только после подстановки конкретных значений. Смысл слова вычисляется в процессе выполнения функции f.

Предложение – единая законченная суперпозиция функций. Смысл предложения вычисляется в процессе построения и выполнения суперпозиции [2].

Семантический анализатор в процессе построения суперпозиции выполняет два основных действия:

1) выбор правильного значения (компьютерного толкования) слова;

2) связывание выбранных значений в осмысленные выражения (целостные словосочетания), т.е. в выражения, которые имеют независимое семантическое толкование.

Требование выделения целостных словосочетаний и их независимого толкования приводит к тому, что связывание слов не сводится к их простой состыковке. При связывании происходит функциональное взаимодействие слов, в результате которого вычисляется новый смысл, в частности – новый семантико-грамматический тип построенной конструкции. Например, при взаимодействии предлога «В» со словосочетанием прошлом году вырабатывается тип «Когда», со словом «лесу» – тип «Где» и т.п. Особым значением является пустое значение (или бессмыслица), которое приводит к разрыву семантической связи [7].

При взаимодействии прилагательного и существительного вырабатывается грамматический тип существительного, однако, семантика словосочетания может существенно отличаться от семантики самого существительного. Например, в выражении «Белая ворона». Это относится не только к прилагательным и существительным, но и к любому слову, способному присоединять к себе другие слова [6].

Часто используемое в лингвистике понятие валентность следует понимать буквально, в химическом смысле: кислород и водород, реализуя свои валентности, перестают быть кислородом и водородом, а создают новую сущность – воду. С точки зрения информатики, присоединяемые слова являются аргументами, из которых присоединяющее их слово строит новую конструкцию, семантика которой может существенно отличаться от семантики ее составляющих. Любой достаточно развитый язык имеет функциональную природу, и только суперпозиция функций адекватна структуре предложений такого языка.

Функциональная природа языка проявляется на всех его уровнях – от механизма словообразования до механизма построения текста. Достаточно посмотреть на аффикс как на функцию, аргументом которой является корень слова, чтобы увидеть удивительную регулярность механизма словообразования русского языка. Благодаря этой регулярности вначале удалось формализовать семантику словообразования, затем автоматически построить

семантическое описание большого количества производных слов, сведя эти описания к описанию слов более простых по морфемному составу. Это позволило существенно автоматизировать процесс построения компьютерного семантического словаря и, в конечном счете, построить его [2].

Качество любого семантического анализатора можно оценивать по тому, как он вычисляет семантико-грамматическое значение предложно-падежных форм. Точность семантического анализа прямо зависит от качества и полноты семантического словаря.

Процесс анализа текста разбивается на три составные части: морфологический анализ, предварительная пословная обработка текста и семантический анализ.

Морфологический анализатор осуществляет обработку текста, вычисляя необходимые для дальнейшего анализа морфологические (грамматические) характеристики каждой словоформы. На этапе предварительной обработки для каждой словоформы вычисляются наборы ее морфосемантических значений. На этапе семантического анализа осуществляется выбор конкретного морфосемантического значения словоформы и связывание всех слов предложения в единую семантическую структуру. Морфологический анализатор использует морфологический (грамматический) словарь, на двух последующих этапах используется семантический словарь.

Словарная статья компьютерного семантического словаря содержит заголовочное слово и его толкование на семантическом языке. Многие слова, – а это, как правило, часто используемые слова – содержат более одного толкования. Основная задача семантического анализатора при анализе конкретного предложения – правильный выбор альтернативы. Этот выбор определяется классом и предложно-падежной формой аргументов, которые способна присоединить к себе та или иная альтернатива.

Постановка задачи анализа текста. Задача анализа текста имеет два варианта: задача понимания текста и задача извлечения смыслов.

Задача понимания текста означает полное и однозначное сопоставление (обычно небольшого) фрагмента текста некоторой формальной структуре, описывающей его смысл. Это на практике проецируется на перевод или на диалог с пользователем.

Задача извлечения смыслов ставит целью выборку из текста всех элементов понимания, полных и частичных, при этом допускается противоречивость элементов между собой. Эта задача

направлена на обработку больших массивов текстов, в частности на поиск, фильтрацию, статистическую обработку.

Для первой задачи различные варианты прочтения многозначных элементов текста анализируются последовательно, и первое адекватное прочтение всего фрагмента прекращает анализ. При этом если частичное понимание фрагмента не складывается в общую структуру, оно само по себе бесполезно, и требуется удалить это частичное понимание, чтобы гарантировать его неучастие в анализе альтернативных вариантов прочтения. Таким образом, первая задача приводит к применению поиска с возвратами (бэктрекинга) [3].

В случае использования второго варианта рассматриваются все смыслы, полные и частичные, извлекаемые из текста, и в том числе альтернативные прочтения одного и того же. Но и здесь нужно позаботиться о том, чтобы альтернативные варианты прочтения не смешивались на верхних уровнях анализа. Если не ограничиваться совсем простыми случаями, вообще говоря, это задача нетривиальная [3].

В данной работе будет рассмотрен второй случай анализа текста, использующий анализатор, предложенный В.А.Тузовым [2].

Для того чтобы получить из общего текста нужную информацию, её необходимо идентифицировать. Для этого понадобиться структура, в которую помещается ключевое слово в именительном падеже (если это существительное или прилагательное) или в инфинитиве (если это глагол) и номер. Данная структура необходима для поиска ключевого слова в семантическом словаре, а номер – для определения значения, в котором используется это слово.

Точность обработки предложений будет зависеть от полноты семантического словаря, который при необходимости можно дополнять. Кроме того, слова, указанные в структуре, обязательно должны присутствовать в тексте, иначе текст будет распознан не верно.

Рассмотрим работу распознавания текста на примере. Имеется структурированный текст по описанию лекарства, приведенный на рисунке 1. Задача: получить только интересующую нас информацию без лишних предложений. Например, необходимо узнать, через какое количество времени начинает действовать препарат.

Анальгин

Лекарственные формы

субстанция, таблетки 500мг, раствор для инъекций, раствор для инъекций 1г, раствор для инъекций 25%, раствор для инъекций 50%, раствор 25%, раствор 50%, капсулы 250мг, таблетки для детей 50мг, свечи для детей 100мг, свечи для детей 250мг, раствор для инъекций 2.5г, таблетки 40мг, субстанция 200г, субстанция 500г, субстанция 1кг, субстанция 25кг

Производители

Аганал Трейдерс(Индия), АЙ СИ ЭН Лексредства(Россия), АЙ СИ ЭН Марбиофарм(Россия), АЙ СИ ЭН Октябрь(Россия), Акрихин ХФК(Россия), Алкалоид(Македония), Аллерген Ставрополь(Россия), Алтайвитамины(Россия), Аналитико-менеджерская группа(Россия), Антивирал(Россия), Асфарма(Россия), Белвитамины(Россия), Белвитамины-Время Фарм.Произв.Компания(Россия), Белгородвитамины(Россия), Белмедпрепараты(Беларусь), Берлин-Хеми АГ(Германия), Биомед(Россия)

ФармГруппа

Анальгетики-антипиретики - производные пиразолона

Состав

Действующее вещество - Метамизол натрия.

Фармакологическое действие

Противовоспалительное, анальгезирующее, жаропонижающее. Угнетает активность циклооксигеназы, снижает образование эндоперекисей, брадикининов, некоторых простагландинов, свободных радикалов, ингибирует перекисное окисление липидов. Препятствует проведению болевых экстра- и проприоцептивных импульсов по пучкам Голля и Бурдаха, повышает порог возбудимости таламических центров болевой чувствительности, увеличивает теплоотдачу. При приеме внутрь быстро и полно абсорбируется. Разрушается в печени. Экскреция проходит через почки. Действие развивается через 20-40 минут.

Показания к применению

Артралгии, ревматизм, хорея, боли: головная, зубная, менструальная, невралгия, ишиалгия, миалгия, при коликах (почечная, печеночная, кишечная), инфаркте легкого, инфаркте миокарда, расслаивающей аневризме аорты, тромбозе магистральных сосудов, воспалительных процессах (плеврит, пневмония, люмбаго, миокардит), травмах, ожогах, декомпрессионной болезни, опоясывающем лишае, опухолях, орхите, панкреатите, перитоните, перфорации пищевода, пневмотораксе, посттрансфузионных осложнениях, приапизме; лихорадочный синдром при острых инфекционных, гнойных и урологических заболеваниях (простатит), укусах насекомых (комары, пчелы, оводы и др.).

Противопоказания

Гиперчувствительность, угнетение кроветворения (агранулоцитоз, цитостатическая или инфекционная нейтропения), тяжелые нарушения функции печени или почек, простагландиновая бронхиальная астма, наследственная гемолитическая анемия, связанная с дефицитом глюкозо-6-фосфатдегидрогеназы, беременность, кормление грудью (на время лечения прекращают).

Побочное действие

Гранулоцитопения, агранулоцитоз, тромбоцитопения, геморрагии, гипотония, интерстициальный нефрит, аллергические реакции (в т.ч. синдромы Стивенса - Джонсона, Лайелла, бронхоспазм, анафилактический шок).

Передозировка

Симптомы: гипотермия, выраженная гипотензия, сердцебиение, одышка, шум в ушах, тошнота, рвота, слабость, сонливость, бред, нарушения сознания, судорожный синдром; возможно развитие острого агранулоцитоза, геморрагического синдрома, острой почечной и печеночной недостаточности. Лечение: индукция рвоты, чреззондовое промывание желудка, назначение солевых слабительных, активированного угля и проведение форсированного диуреза, ощелачивание крови, симптоматическая терапия, направленная на поддержание жизненно важных функций.

Особые указания

Необходим врачебный контроль. Не рекомендуется регулярный длительный прием вследствие миелотоксичности. Исключается использование для снятия острых болей в животе (до выяснения причины). При назначении больным с острой сердечно-сосудистой патологией необходим тщательный контроль за гемодинамикой. С осторожностью применяют у пациентов с уровнем систолического артериального давления ниже 100 мм рт.ст., с анамнестическими указаниями на заболевания почек (пиелонефрит, гломерулонефрит) и при длительном алкогольном анамнезе. При применении метамизола возможно красное окрашивание мочи за счет выделения метаболита [8].

Рисунок 1 – Описание лекарства «Анальгин»

В структуру помещаем слово ДЕЙСТВИЕ. Смысловые понятия этого слова в семантическом словаре приведены на рисунке 2.

1 - работа, деятельность, совершение чего-либо. Скоординированные действия;
2 - совокупность поступков кого-нибудь. Непродуманные действия;
3 - работа, функционирование какой-нибудь машины, агрегата, предприятия и т.д.;
4 - влияние на кого-, что-нибудь. Позитивное действие;
5 - совокупность и развитие действий в литературных произведениях, кино и т.д.;
6 - боевые действия;
7 - законченная часть драматического, оперного, балетного произведения (синоним акта);
8 - основной вид математического произведения.

Рисунок 2 – Смысловые понятия слова «ДЕЙСТВИЕ» в семантическом словаре

Действие
…
4. ДЕЙСТВИЕ{СущНеодуш$11227~@ОНО$5@Им$11227~@ОНО$5@Вин} $15(!Им!Вин,!Что)
…
6. ДЕЙСТВИЕ{СущНеодуш$11251~@ОНО$5@Им} $11251(!Им,!Какое)

Развивается (имеет два смысла: проявлять новые способности и подвергаться действию раскручивания)
1. РАЗВИВАТЬСЯ{Глаг} Oper00(!Глаг,Magn_a~СПОСОБНОСТЬ$1103(!Им))
2. РАЗВИВАТЬСЯ{Глаг} Caus(ПРИЧИНА$10/05~!От,Lab(!Глаг,РАСКРУЧИВАНИЕ$125~!Им))

Через
1.ЧЕРЕЗ{Предл@черезВин@черезПред@черезКого@вОНИ$5@Им}$71(!Вин\!Пред\!Кого\!ОНИ$5@Им)
2.ЧЕРЕЗ {Предл $13~@Как} Direkt_y(#,СПОСОБНОСТЬ 14~!Вин) ,
3.ЧЕРЕЗ {Предл $15~@Что} Direkt_y(#,ВРЕМЯ$3~!Вин)

Symbol: 20-40

Минут
1.МИНУТА{СущНеодуш$3~@ОНА$5@Вин} $327(!поРод,!вВин\!наВин\!черезВин)

Symbol: .

Рисунок 3 – Альтернативы пословной обработки предложений

Для данного примера выбирается вариант под номером 4, поэтому в структуру, кроме слова, заносится еще и цифра 4.

Семантический анализатор начнет поиск слова ДЕЙСТВИЕ в каждом предложении, начиная с первого. По тексту это слово впервые встречается в предложении: «Действие развивается через 20-40 минут». Это предложение первым начинает анализироваться. Все возможные альтернативы пословной обработки приведены на рисунке 3.

На следующем этапе семантического анализа происходит отбор нужных альтернатив. В данном примере морфологические признаки слова МИНУТА позволят однозначно определить альтернативу предлога ЧЕРЕЗ. Совпадение класса и падежа указывают на вариант под номером 3. Морфологическая часть предлога образует класс под номером 15 с вопросом ЧТО – это совпадает с 4-ой семантикой слова ДЕЙСТВИЕ. Далее аналогичным образом по падежу и классу определяется однозначность слова РАЗВИВАТЬСЯ, что соответствует первому варианту.

После этого можно делать проверку чисел на совпадение. В структуре слову ДЕЙСТВИЕ соответствовала цифра 4. Семантическим анализатором для этого слова была выбрана альтернатива под номером 4. Цифры совпадают, значит предложение подходит и оно будет записано в выходной файл в качестве результата. Таким образом, после заполнения структуры словами нужного смысла получаются необходимые смысловые предложения.

Выводы. Анализ текста с помощью данного семантического анализатора дает неплохие результаты. Однако здесь возникают жесткие требования: точный выбор альтернативы и полнота семантического словаря. Если эти требования не будут выполняться, то результат анализа может быть непредсказуемым. В дальнейшем предусматривается разработка модуля контроля ошибок. Статистика наиболее распространенных ошибок позволит решить вопрос частоты их появления и контроля.

Список литературы

1. Тузов В.А. Компьютерная семантика русского языка. – СПб.: Изд-во СПбГУ, 2004. – 400 с.
2. Тузов В.А. Компьютерная семантика русского языка /Труды международной конференции «Диалог 2001».
3. Пацкин А.И. Применение техники управления событиями для анализа текста в системе абриаль.ДИАЛОГ'2003, Труды межд. семинара. М., 2003.

4. Nirenburg et al., 2004] Nirenburg S., Raskin V. Ontological Semantics. – Cambridge, MA: MIT Press, 2004
5. Рубашкин В. Ш. О методах анализа связного текста // Вопросы информационной теории и практики. - Вып. 49. - М.: ВИНИТИ, 1983.
6. Лукашевич И.П. Проблемы информационного взаимодействия в медицине. // Новости искусственного интеллекта. 2005. № 2.
7. KAON — The KArlsruhe ONtology and Semantic Web tool suite - http://kaon.semanticweb.org/к
8. http://www.medinfo.ru

Получено 01.06.2007.

ББК: 32.973

ПОЧАТКОВІ ЕТАПИ ВИКОРИСТАННЯ КОМП'ЮТЕРА, ЯК КРЕСЛЯРСЬКОГО ІНСТРУМЕНТА

І.Г. Балюба, В.І. Поліщук, Б.Ф. Горягін, Ж.В. Старченко
Донбаська національна академія будівництва і архітектури

Викладено досвід використання основ комп'ютерної системи виконання креслень КОМПАС на початкових етапах підготовки спеціалістів інженерного профілю.

Постановка проблеми. Комп'ютерні системи проектування машин будівель і споруд повністю змінили інженерну практику. Інженер-проектувальник, що не володіє комп'ютерними технологіями не може вважати себе, в повній мірі, спеціалістом, не може знайти роботу за фахом. Креслення, як основа вираження результатів конструювання, є основою комп'ютерних систем проектування. Комп'ютер стає єдиним креслярським інструментом після ручного ескізного зображення об'єкта. Стандартне ручне оформлення креслень складає значну частину часу, який потрібно використовувати в навчальному процесі для креслення при підготовці інженерів. З іншого боку, стандарти оформлення креслень – закладені в комп'ютерні системи, і студенту достатньо знань стандартів, так як засвоєння навичок роботи з комп'ютерними кнопками не можуть іти ні в які порівняння із засвоєнням навичок ручного креслення з додержанням стандартів.

В роботі поставлена задача поділитись досвідом використання комп'ютера, як креслярського інструмента, на самих ранніх стадіях виконання креслень студентами.

Аналіз останніх досліджень. Питання перебудови викладання креслення в напрямку використання комп'ютерних технологій настільки актуальні, що ні один вищий навчальний заклад, відповідного рівня, не може залишатись осторонь. Виходячи з цього, раціонально проаналізувати досвід роботи з цієї тематики в Донбаській національній академії будівництва і архітектури.

Починаючи з 2000 року в академії проводиться система неперервної комп'ютерної підготовки студентів: програмування, креслення, розрахунки. Особливе місце в цій системі виділено для освоєння комп'ютерного виконання креслень. Після серйозної підготовки викладачів на шостому та сьомому семестрах, на

професіональному напрямку «Будівництво», в обсязі 70 годин лабораторних занять, кафедра «Інженерна та комп'ютерна графіка» почала викладання нової дисципліни «Комп'ютерна графіка». Базовим продуктом викладання була прийнята 2D система AutoCAD. Методичною основою навчання студентів був прийнятий, розроблений викладачами кафедри, посібник [1], звіт про цю роботу відображено в [2].

Паралельно, кафедра проводила пошуки вдосконалення системи викладання, працювала індивідуально з окремими студентами в пошуках нових можливостей, брала участь в олімпіадах. Досвід інших ВНЗ, бажання розширити можливості випускників академії, спонукали звернути увагу на програмний продукт КОМПАС, що використовує близькі до діючих в Україні ДЕРЖСТАНДАРТИ. В 2002 році на третьому курсі механічних спеціальностей проведено експеримент викладання КОМПАС-ГРАФИК: на протязі семестру розглянуті основні положення 2D системи та способи побудови креслень деталей і складальних креслень [3], підготовлено методичне забезпечення для його подальшого впровадження. Ефективний модуль системи попереднього тривимірного моделювання програмного продукту КОМПАС з подальшою можливістю побудови асоціативних видів, спонукав використання його в навчальному процесі, він був випробуваний на експериментальній групі студентів в вересні 2003 року [4], [5]. Після успішного експерименту на кафедрі підготовлено і видано навчальний практичний посібник для подальшого впровадження тривимірного моделювання в навчальний процес [6].

Формулювання цілей статті. При ручному виконанні креслень, багато часу затрачується на формування навичок оформлення креслень згідно з діючими ДЕРЖСТАНДАРТАМИ, чого в свій час не було відпрацьовано в школі на уроках креслення. Шрифти, лінії, нанесення розмірів - дуже важливі розділи креслення для освоєння яких в програмах вищої освіти не передбачається часу, але відсутність навичок в яких, особливо в будівельному кресленні, де товщини ліній несуть змістовну загрузку, роблять креслення непридатними для однозначного сприймання. Виникає потреба включати розділи геометричного креслення в програму вищої освіти, суттєво скорочуючи дуже важливі розділи, наприклад, читання креслень та виконання над ними необхідних для цього операцій.

З другого боку, в комп'ютерній системі КОМПАС закладені діючі в Україні ДЕРЖСТАНДАРТИ геометричного креслення, і не тільки його, потрібно тільки їх знання, а вироблення навичок роботи з

відповідними клавішами на комп'ютері потребує мало часу. Раннє знайомство з основами комп'ютерного виконання креслень має самостійну цінність, так як стимулює студентів до самостійного, а потім і аудиторного, більш глибокого, освоєння відповідних комп'ютерних систем. Досвід поєднання ручного і основ комп'ютерного виконання креслень буде ціллю даної статті.

Основна частина. Пропонується проводити уроки креслення в предметній аудиторії і лабораторні роботи в комп'ютерному класі, поділивши відведений час на дві частини. Дві третини часу проводиться в предметній аудиторії, де формуються знання для читання і виконання креслень. Третину часу проводиться в комп'ютерному класі, де наочно, з великою кількістю прикладів, демонструються правила оформлення креслень згідно ДЕРЖСТАНДАРТІВ, виконуються вправи з геометричного креслення в комп'ютерній системі КОМПАС. Комп'ютерна система використовується не в повну силу, етапи виконання вправ споріднені з ручним виконанням. Формується необхідність додержання стандартів відносно форматів, масштабів, ліній, шрифтів і т.і., яка переноситься на виконання креслень в ручному режимі при роботі в предметній аудиторії. Ручне і комп'ютерне виконання креслень, по можливості, узгоджується в часі. Перша виведена комп'ютерна робота на друк являється зразком для оформлення ручних креслень. Зміст роботи в комп'ютерному класі приведено в таблиці 1.

Таблиця 1

№ п.п	Зміст навчальних занять	Кіл. годин
1	"Ручний" та машинний способи виконання креслень. Коротке знайомство з графічною програмою КОМПАС-3D. Графічний інтерфейс програми, робота с Компактною панеллю. Панель властивостей, введення параметрів. Загальні правила виконання креслень. Вивчення ДСТУ, ЕСКД, ГОСТ 2.301-68, ГОСТ 2.302-68, ГОСТ 2.303-68, ГОСТ 2.304-81, ГОСТ 2.306-68.	2
2 3	Графічна робота **ГР1.1: 1 аркуш ф. А3.** Засобами програми КОМПАС-3D виконати побудову контурів деталей – прокладка, вал, корпус. Загальні правила нанесення розмірів на кресленнях: вивчення ГОСТ 2.307-68. Проставити всі необхідні розміри на деталях, заповнити основний напис. Вивід графічної роботи на друк.	4
4	Здача **ГР1.1.** Проекційне креслення. Побудова засобами програми КОМПАС-3D тривимірних моделей деталей.	2

	Основи створення тривимірної моделі деталі. Ескізи та операції. Створення у моделі розсічених видів, перетину моделі площиною, перетину моделі довільним ескізом. Виключення елементів з розрахунку. Використання тривимірної моделі деталі для одержання робочої документації. Створення асоціативного креслення деталі по тривимірній моделі та стандартних видів деталі. Побудови розрізів деталі. **Аудиторна робота:** по ортогональному кресленню технічної деталі побудувати тривимірну модель. Створити у моделі розсічений вид. По тривимірній деталі створити три основних види деталі та побудувати розрізи.	
5 6	Видача **ГР1.3: 1 аркуш ф. А3.** Виконання графічної роботи – побудувати тривимірну модель технічної деталі. Побудувати об'ємний виріз. Побудова асоціативних видів. По тривимірній деталі створити три основних види деталі та побудувати складні розрізи. Побудувати аксонометричну проекцію деталі з частковим вирізом. Проставити розміри на деталях, заповнити основний напис. Вивід графічної роботи на друк. Здача графічної роботи **ГР1.3**	4
7	Основи машинобудівельного креслення. Загальне поняття про різьбу. Види різьбових з'єднань. Зображення з'єднання болтом, дійсне, спрощене та умовне. Використання прикладних бібліотек КОМПАС-3D. Видача **ГР2.1: 1 аркуш ф. А3, або А4** Побудувати з'єднання болтом. Використовуючи бібліотеку кріпильних елементів КОМПАС-3D, накреслити дійсне зображення з'єднання болтом. Вивід графічної роботи на друк. Здача графічної роботи **ГР2.1**	2
8 9	Видача графічної роботи **ГР2.3: 1 аркуш ф**. **А3**, **або А4**. Робочі креслення деталей. По ескізу, виконаному в аудиторії, засобами програми КОМПАС-3D виконати робоче креслення деталі типу "кришка". Вивід графічної роботи на друк. Здача **ГР2.3**	4
	Всього	**18**

Професійне будівельне креслення проводиться в ручному режимі, так як подальше комп'ютерне його виконання передбачено при серйозному освоєнні комп'ютерних систем КОМПАС і AutoCAD.

В академії створюється тенденція поставити використання комп'ютера, як креслярського інструмента, на перше місце, для цього кафедрою виконується науково-дослідна робота. В даний час

проводяться перші практичні випробування використання комп'ютера на ранніх стадіях викладання креслення. Першими висновками є зацікавленість студентів, труднощі практичної організації поєднання ручного і комп'ютерного виконання креслень і стандартні труднощі запровадження нового підходу в викладацькій роботі. Позитивним є ентузіазм виконавців (студентів і викладачів), наявність домашніх комп'ютерів у студентів (відповідний курс стимулює їх придбання). Кафедра констатує, що подібний підхід дає можливість зосередити свою увагу на читанні і виконанні креслень, значно більше вправ виконує студент в ручному режимі, виникла можливість перенести вироблення навичок ручного оформлення креслень згідно ДЕРЖСТАНДАРТІВ на домашню самостійну роботу студентів.

Кафедрою планується подібне використання «Панели рисования» системи Microsoft Word при виконанні деяких епюрів нарисної геометрії, що дасть можливість масового комп'ютерного виконання пояснювальних записок до курсових робіт.

Література

1. Мущанов В.П., Балюба І.Г., Старченко Ж.В. AutoCAD 2000. Компьютерная графика / Макеевка: ДонГАСА, 2001. – 143 с.
2. В.І. Поліщук, Ж.В. Старченко. Про систему неперервної комп'ютерної підготовки студентів будівельних та механічних спеціальностей/ Сучасні проблеми геометричного моделювання. Праці НУ «Львівська політехніка»: Матеріали міжнар. наук.-практ. конф. - Львів, 2003, С.255-256.
3. Мущанов В.П., Поліщук В.І., Старченко Ж.В. Машинная графика и компьютерные технологии. КОМПАС-ГРАФИК 5Х / Учебное практическое пособие. Часть 1. Основы 2D-системы компьютерного проектирования. – Макеевка, ДонГАСА, 2003. -262с.
4. Белицкий Д.Г. Современные компьютерные 3D технологии в машиностроении и их применение в учебном процессе ДонГАСА/ Матеріали другої загальноукраїнської студентської науково-практичної конференції «Молодь, соціальна політика і проблеми національного відродження України» – Донецьк, Донецький інститут ринку і соціальної політики, 2003. – С.25-26.
5. Горягин Б.Ф., Евсеенко А.В. Опыт применения геометрического моделирования на ПЭВМ в системе Компас-3D для создания рабочих чертежей изделий (на примере замка опалубки)/ Матеріали 30 (3 міжнародної) наукової конференції студентів, аспірантів и молодих вчених, том 1. Вісник ДонДАБА, 2004-3(45). – С.52-55.
6. Мущанов В.П., Полищук В.І., Старченко Ж.В. Машинная графика и компьютерные технологии. КОМПАС-3D. Учебное практическое пособие. Часть 2. Основы 3D технологии компьютерного проектирования. – Макеевка: ДонНАСА, 2007. -533 с.

Получено 12.05.2007

УДК 681.518

ИСПОЛЬЗОВАНИЕ МОДЕЛИРОВАНИЯ ПРИ БАЛАНСИРОВКЕ НАГРУЗКИ В КОМПЬЮТЕРНОЙ СЕТИ УПРАВЛЕНИЯ

М.Е. Щербакова
Северодонецкий технологический институт

Предлагается метод балансировки нагрузки в распределенной компьютерной системе управления с использованием нескольких моделей программного комплекса в виде графов взаимосвязей между программными модулями.

В любой распределенной параллельной вычислительной системе очень важно правильно распределить задачи между узлами сети, для того чтобы улучшить загрузку всех компьютеров и, следовательно, повысить производительность распределенной параллельной вычислительной системы. Решение этой задачи особенно важно для распределенных компьютерных систем управления.

Задача формулируется следующим образом. Имеется сеть компьютерной системы управления, состоящая из рабочих станций и контроллеров. Для сети известен состав модулей программного комплекса, их взаимосвязи друг с другом, а также затраты времени на работу каждого модуля. Требуется так распределить программные модули между рабочими станциями и контроллерами, чтобы затраты времени на работу программного комплекса были минимальными.

В случае, если для обмена данными между рабочими станциями и удаленными на большие расстояния контроллерами используются медленные линии связи, например, некоммутируемые или, что еще актуальнее, коммутируемые линии связи, возникает задача уменьшения времени взаимодействия и объема информации, передаваемой по таким линиям связи. Формально это можно записать так:

$$\sum_{i=1}^{n}\sum_{j=1}^{m} F_{ij} \Rightarrow \min \qquad (1)$$

где Fij – время взаимодействия программы j с программой i; m – число программ, выполняемых рабочей станцией, n – число программ, выполняемых контроллером.

Задачу минимизации времени взаимодействия программ из разных узлов сети предлагается решать следующим образом.

1. Строится модель комплекса программ в виде неориентированного связного графа G = (U, V) с множеством узлов V

= $\{v_1, v_2, \ldots v_n\}$, каждый из которых соответствует отдельной программе, и множеством рёбер $U = \{u_1, u_2, \ldots u_m\}$, соответствующих связям между программами. Каждому ребру $[v_i, v_j]$ припишем вес $c(i, j)$, равный затратам времени контроллера на взаимодействия между соответствующими программными модулями при условии, что одна из них выполняется рабочей станцией, другая – контроллером.

2. К графу добавляются ещё два узла: К и РС, представляющие контроллер и рабочую станцию соответственно. С узлом К (контроллер) соединяются узлы графа, представляющие программы, использующие специальное оборудование контроллера или же выполняющие специфичные функции, которые должны всегда выполняться контроллером. Вес таких ребер равен бесконечности: $c(К, j) = \infty$. По аналогии, с узлом РС (рабочая станция) соединяются только те узлы графа, которые должны обязательно выполняться рабочей станцией, если таковые имеются. Вес этих ребер тоже устанавливается равным бесконечности: $c(i, РС) = \infty$.

3. Полученный граф взаимосвязей между программами можно рассматривать как транспортную сеть с источником К и стоком РС. Исходная задача сводится к задаче нахождения минимального разреза построенной таким образом сети (рис. 1).

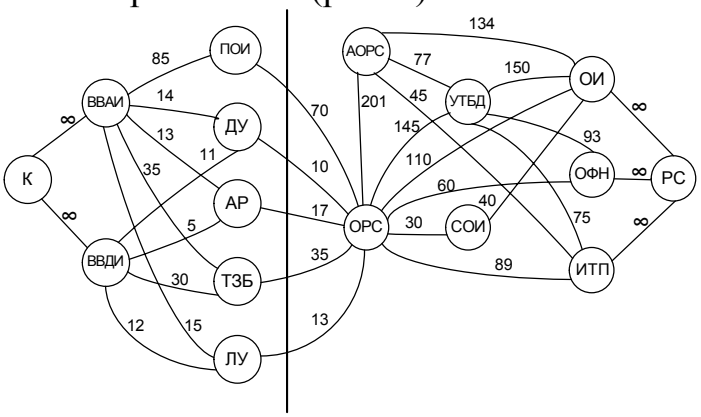


Рис. 1 Пример графа взаимосвязей для минимизации времени взаимодействия программных модулей

Решение задачи было проверено на модели программного комплекса, состоящего из следующих программных модулей: ввода – вывода аналоговой информации (ВВАИ), ввода – вывода дискретной информации (ВВДИ), первичной обработки информации (ПОИ), дистанционного управления (ДУ), автоматического регулирования (АР), технологических защит и блокировок (ТЗБ), логического управления (ЛУ), сервера ОРС (ОРС), адаптера ОРС (АОРС), управления технологической базой данных (УТБД), статистической обработки информации (СОИ), отображения информации (ОИ),

обработки и фиксации нарушений (ОФН), истории технологического процесса (ИТП). В графе на рис. 1 минимальный разрез делит множество программ на 2 части: те, что будут выполняться контроллером, и те, что будут выполняться рабочей станцией. Для нахождения минимального разреза на графе существует множество алгоритмов. В программной реализации метода был использован алгоритм Форда – Фалкерсона.

Вторая модификация разработанного метода минимизирует время выполнения программ контроллерами. Этот вид оптимизации имеет смысл проводить в случаях, когда нужно уменьшить число контроллеров с целью снижения стоимости технических средств и повышения надежности функционирования системы в целом.

Граф в этом случае строится так же, как и в предыдущем, но каждый узел соединяется еще и с узлом РС. Этим ребрам присваивается вес, равный времени выполнения соответствующих программных модулей контроллером (рис. 2).

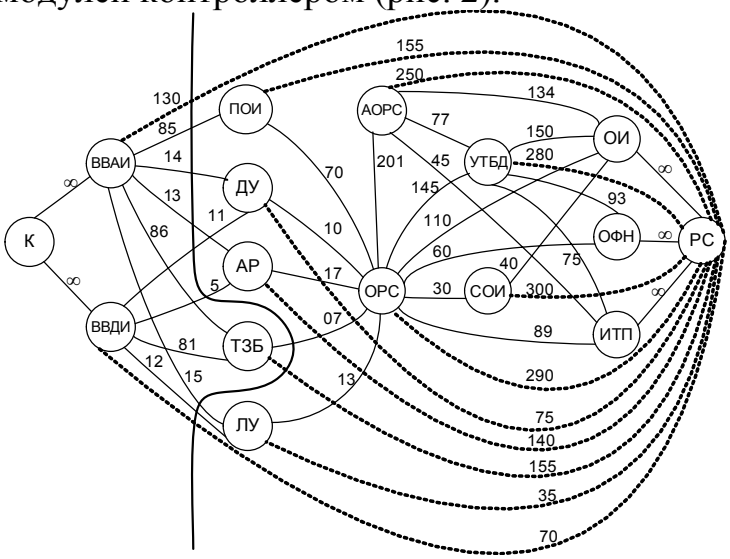


Рис. 2. Граф взаимосвязей для минимизации времени выполнения программ контроллером

Третья модификация метода минимизирует время выполнения программ рабочей станцией и контроллером и является дальнейшим развитием предыдущих двух методов. На этот раз минимизируется сумма времен выполнения программ контроллером, рабочей станцией и времени взаимодействия программ по сети. Использование этого метода минимизирует гарантированное время ответа всей системы в целом, уменьшает время реакции системы на внешние события, повышает оперативность реагирования на запросы технологического персонала. Метод аналогичен предыдущим двум методам во всем,

кроме способа построения графа для распределения программных модулей (рис. 3).

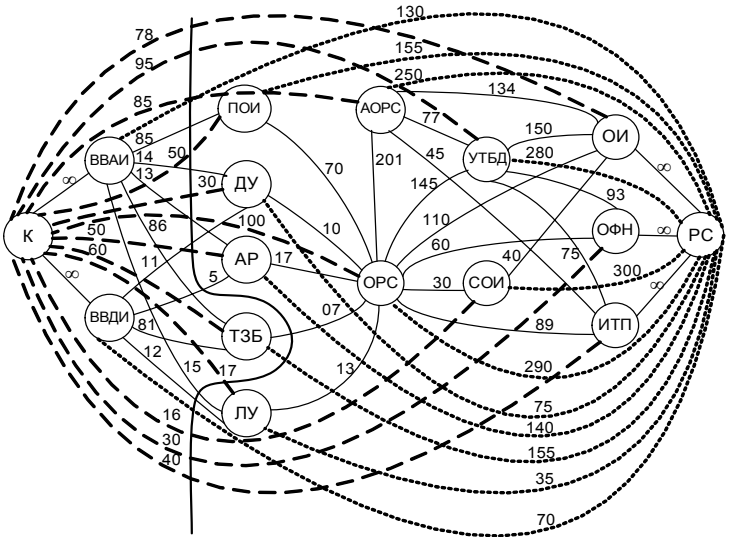


Рис. 3. Граф взаимосвязей для минимизации времени выполнения программ рабочей станцией и контроллером

Все вершины - программные модули соединяются ребрами с узлами РС и К. Этим ребрам присваивается вес, равный времени выполнения программного модуля контроллером и рабочей станцией соответственно. В минимальный разрез входят ребра, веса которых равны временам выполнения программ, размещаемых в рабочей станции, ребра, веса которых равны временам выполнения программ, размещаемых в контроллере, и ребра с весами, равными времени взаимодействия программ по сети. Таким образом, минимизируется суммарное время, которое контроллер и рабочая станция затратят на выполнение всех программ, если эти программы будут выполняться чисто последовательно, т. е. самое большое время одного цикла обработки системой всех входных и выходных данных.

Три вышеописанных метода предназначены для использования в программах установки и балансирования нагрузки пакетов систем управления. В частности, в пакете КВАРЦ [1] метод распределения программных модулей, минимизирующий время выполнения программ рабочей станцией и контроллером, дал возможность улучшить одну из важнейших характеристик систем реального масштаба времени – гарантированное время ответа на 15 – 19%.

Литература

1. Е.В. Щербаков, М.Е. Щербакова, В.К. Охрамович Автоматизированное проектирование ППО КСУ на базе пакета программ «Кварц». Монография /под ред. д.т.н., проф. А.Г. Руденко – Луганск: Издательство Восточноукраинского национального университета имени В.Даля, 2003.-200 с.

Получено 27.06.2007 г.

УДК 629.33: 681.518.5

РАСПРЕДЕЛЕННАЯ СИСТЕМА АВТОМАТИЧЕСКОГО УПРАВЛЕНИЯ ТРАНСПОРТНЫМ СРЕДСТВОМ

Чернов Е.И.

Таганрогский технологический институт Южного федерального университета

В работе рассматривается один из подходов повышения эффективности использования транспортных магистралей и безопасности движения. Автоматического движение транспортного средства реализуется с использованием распределенной микропроцессорной системы моделирования, прогнозирования и управления.

Во всем мире количество автотранспорта на дорогах неуклонно растет (на 45 млн. автомобилей в год) при общем автомобильном парке 850 млн. автомобилей Автомагистрали, сосредоточенные вблизи крупных городов близки к пределу своей пропускной способности, особенно в пиковые часы. Ситуация приближается к полному параличу движения на отдельных, наиболее напряженных участках. Проблема решалась в основном за счет расширения дорожного полотна или прокладки скоростных участков. Сейчас в ряде стран такой потенциал исчерпан как по экологическим, так и по финансовым соображениям, например, в Европе против таких мер выступили экологи, в Японии – нехватка территории, Китай – дна из самых густонаселенных стран, а в США строительство каждой “погонной” мили скоростной магистрали обходится сегодня более чем в 100 миллионов долларов, в России - порядка 1 млн. 350 тыс. долларов за 1км, при ширине дорожного полотна 4м. За повышение безопасности движения и организацию транспортных потоков должны взяться интеллектуальные многопроцессорные системы [1, 2, 3, 4, 5].

Хотя разработки типовых автономных систем вождения автомобилей в среде интеллектуальных транспортных систем проводились во многих странах, все они создавались применительно к легковым автомобилям. При этом цена таких систем очень высока и составляет значительную часть стоимости автомобиля. Для грузовых машин и автобусов такие варианты практически не рассматривались из-за их сложной нелинейной динамики и специальных требований к приводу. Однако, финансовая привлекательность (доля электроники в

цене грузовика незначительна) способствует развитию систем автовождения именно для большегрузных автомобилей [4,5,6,7].

Основная цель разработки – это обеспечить движение транспортных средств (большегрузных грузовиков, автобусов) в колонне на магистрали с использованием принципа «ведущий-ведомый» или «делай как я» или эскалатора. Транспортные средства (ТС) движутся за «головным» на безопасном расстоянии друг от друга с одинаковой скоростью и повторяют все маневры «главного» и изгибы дороги без участия водителя. Так можно формировать пакеты (блоки) ТС, которые движутся в своей (например, правой) полосе движения из пункта А в пункт Б. Такая организация движения пакетов ТС позволит уплотнить транспортный поток, обеспечить равномерную загрузку автомагистрали и безопасность движения.

Для решения поставленной задачи предлагается оснастить ТС следующими стандартными системам и узлами:

- автоматической коробкой переключения передач (АКПП);
- круиз-контролем (система обеспечения движения ТС с постоянной скоростью);
- тормозной системой с ABS, ASR и EDS (система антиблокировки колес, система антипробуксовки ведущих колес, система блокировки дифференциала ведущего моста).

Такие системы уже используются на серийных автомобилях.

К нестандартным узлам следует отнести систему рулевого управления, которая осуществляет повороты руля по командам от электронного блока управления с помощью электродвигателя с редуктором и электронную педаль газа, которая по командам от управляющей системы осуществляет поворот дроссельной заслонки или перемещение топливной рейки в дизелях.

Сенсорная часть системы управления движением (рисунок 1) состоит из:

- дорожных датчиков, которые утоплены в дорожное покрытие или в светоотражательные элементы и излучают слабый опорный радиосигнал, принимаемый специальной антенной бортового компьютера ТС. Датчики устанавливаются на расстоянии 5-10 метров. По амплитуде принимаемого сигнала бортовая система управления определяет отклонение курса ТС в пределах заданной полосы движения;
- видеокамеры, с помощью которой можно идентифицировать впереди идущее ТС (ведущий или вклинившийся автомобиль);
- дальномера, который определяет расстояние до впереди идущего ТС и его маневры;

- приемника-передатчика радиоканала (УКВ–диапазон), предназначенного для информирования сзади идущих автомобилей о маневрах впереди идущего,
- приемника GPS для определения местоположение ТС,
- приемника радиосигнала от дорожных датчиков

Рисунок 1 – Структура распределенной системы управления транспортным средством

Подсистема первичной обработки информации соответственно разделяется на:

- блок обработки радиосигналов, который осуществляет прием сигналов дорожных датчиков, их оцифровку и передачу в распределенную микропроцессорную систему (РМС), где

производится сравнение их с нормированным сигналом и выработка сигнала управления рулем;

- блок обработки видеоизображения, который предназначен для оцифровки изображения и передача сформированного кадра в МВС;
- блок обработки дальнометрической информации служит для выработки посылаемых сигналов для дальномера и приема информации от дальномера с целью определения расстояния до впереди идущего ТС.

Время, характеризующее дистанцию меду двумя ТС передается в РМС, где проводится подсчет расстояния между ТС и скорость их сближения (расхождения);

- блок контроля параметров необходим для передачи параметров движения «ведущего» ТС на антенну радиоканала или GSM для сзади идущих ТС и для приема и обработки этих сигналов на «ведомых» ТС. Навигация на транспортной магистрали базируется на применении технологии GPS.

Распределенная микропроцессорная система осуществляет прием информации из подсистемы входной обработки, проводит расчеты: расстояния между ТС, скорости их сближения (расхождения), идентификации впереди идущего ТС, отклонения движения ТС от заданного полосой движения курса. При этом учитываются все маневры впереди идущих ТС по сигналам полученным по каналу GSM, GPRS. РМС вырабатывает сигналы на соответствующие блоки и узлы управления ТС:

- круиз-контроль для движения с постоянной скоростью,
- блок рулевого управления для сохранения траектории движения по полосе,
- тормозную систему для предотвращения столкновения и поддержания безопасного расстояния между ТС,
- электронная педаль газа (управление топливной рейкой).

РМС представляет локальную сеть, состоящую из нескольких микроконтроллеров (например, Infineon 166/167), объединенных между собой, блоками первичной обработки сигналов и блоками управления ТС интерфейсом CAN. Достаточно высокая производительность РМС позволяет не только управлять работой основных узлов ТС в реальном времени, но и моделировать и прогнозировать ситуацию на дороге, а также проводить диагностику технического состояния «жизненоважных» систем автомобиля, информировать о неполадках головную машину колонны, тем самым обеспечить безопасное движение блоков автомобилей на магистрали.

Оснащение ТС большой грузоподъемности такой интеллектуальной системой незначительно увеличит стоимость автомобиля, в то же время позволит уплотнить транспортные потоки за счет создания быстродвижущихся колонн.

Литература

1. Ding Jian-mei, Wang Ke cong. Harbin gongye daxue xuebao = J. Harbin Inst. Technol. 2002. 34. №5 с.671-674.

2. Tang Zhenmin, Zhao Jindyu. Nanjing Ligong daxue xueboo. Ziran kexue ban = J. Nanjing Univ. Sci. and Technol. Natur. Sci. 2003. 27. №1 с. 6-10.

3. IEEE Spectrum online. 2002. January 10.

4. Мастер 12 вольт. 1998. №1 с. 60-61.

5. Аоки Кейдзи. Denki gakkai vonbunshi. D =Trans. Inst. Elec. Eng. Jap. D. 2000. 120. №12 с. 1508.

6. Yashiro Tonoyuki, Ariyasu Kyoko, Fukui Ryotaro, Shigeno Hiroshi, Matsushita Yutaka. Jono shori gakkai ronbumshi = Trans. Inform. Process. Soc. Jap. 2002. 43. №12 с. 3756-3764.

7. http: // www.techweb.com, http: // www.data.com.

Получено 21.05.07

ОЦЕНКА ЭФФЕКТИВНОСТИ ИНСТРУМЕНТАЛЬНОЙ СИСТЕМЫ ДЛЯ СОЗДАНИЯ ИНТЕЛЛЕКТУАЛЬНЫХ САПР КАК СИСТЕМЫ С БАЗОЙ ЗНАНИЙ

Григорьев А.В.
Донецкий национальный технический университет

Describes a knowledge conception method specific character. Consider the teaching processes organization complication estimation questions and inference in knowledge base.

Введение. Ранее автором в работах [1-6] были изложены основные принципы нового метода представлений знаний в базе знаний инструментальной оболочки для построения интеллектуальных САПР – метаэвристической оболочки (МЭО). Данный метод представления знаний имеет ряд специфических особенностей:

- использование контекстно-свободных грамматик как основы представления знаний;
- сужение грамматик до уровня И-ИЛИ-деревьев;
- наличие продукций работы с образцами, рассматриваемыми как КС грамматики;
- наличие продукций, задающих семантические зависимости в КС грамматиках;
- организация обучения, вывода и процесса уточнения задачи синтеза на основе теоретико-множественных операций (ТМО) над КС порождающими грамматиками.

Завершенность работ по созданию алгоритмов выполнения теоретико-множественных операций над КС грамматиками, модели построения интеллектуальных САПР с выбранным способом представления знаний, а так же ряда смежных работ, делают актуальной задачу оценки эффективности работы интеллектуальных САПР с данным методом представления знаний.

Цель данной работы является:

- анализ существующих оценок качества систем с базами данных;
- выбор или построение собственных оценок;
- формирование вывода о качестве баз знаний и определение перспективных работ.

1. Специфика представления знаний в МЭО

Главная специфика представления знаний в МЭО - построение базы знаний, работающий с формально-языковым представлением моделей, т.е. работа с базой знаний текстов, представленных

грамматиками с определенными над ними дополнительными продукциями. Предложена следующая модификация продукционного метода представления знаний:

1) Факты:

- совокупностью фактов, т.е. основой представления знаний являются контекстно-свободные порождающие грамматики, суженные до уровня И-ИЛИ-дерева, описывающие множество возможных текстов-решений (прототипов или технических заданий - ТЗ), определенных в рамках модели САПР решения типичных задач проектирования;

- почему грамматики сужены до И-ИЛИ-дерева: для грамматик, имеющих циклические ссылки в синтаксических правилах вывода, практически невозможно разработать корректный алгоритм ТМО (в общем случае, из-за теоретической невозможности таких ТМО [7]);

- ТМО над грамматиками нужны для обеспечения: вывода, обучения, уточнения постановки задачи;

2) Правила вывода:

- система правил вывода - семантические продукции над ИЛИ-узлами контекстно-свободных порождающих грамматик, суженных до И-ИЛИ-дерева;

- назначение продукций – сузить число возможных комбинаций альтернатив ИЛИ-узлов до семантически верных, описывающих имеющие смысл прототипы;

- модальность правил вывода над множеством фактов (т.е. текстов описания модели) - существует (соответствует требованиям проектировщика) или – не существует (не соответствует требованиям);

- предполагается деление правил вывода на явные и - неявные;

- неявные правила вывода, оформлены в форме списков прототипов, связанных с терминальными и нетерминальными символами грамматик; формируются автоматически при обучении базы знаний; такие грамматики представляют собой модифицированную предикативную (модифицировано-атрибутную) грамматику, принятую в ПРОЛОГе;

- явные правила вывода, оформлены в виде продукций над ИЛИ-узлами контекстно-свободных порождающих грамматик, суженных до И-ИЛИ-дерева, формируются экспертом по знаниям;

3) Процесс вывода в пределах модуля знаний:

- процесс вывода – сужение множества возможных решений, описанных в модуле знаний - И-ИЛИ-дереве, описывающим

множество возможных решений, в соответствии с рядом требований, задаваемых конечным пользователем (проектировщиком);

4) Работа с базами знаний как с моделями предметных областей:

- имеются операции над базой знаний – продукционный вывод, обучение, формирование требуемой базы знаний;

- обучение – объединение грамматик с грамматиками либо отдельными прототипами, с сохранением или без сохранения имеющихся семантических связей (правил вывода), заданных в одной из двух форм – явной и неявной;

- формирование требуемой базы знаний - система обобщенных макро-продукций, использующих теоретико-множественные операции над базами знаний, поддерживающих диалог с пользователем на предмет выявления требуемой ему предметной области.

Обсудим типичные и специфические черты предлагаемого метода представления знаний:

1) Типичные черты:

- И-ИЛИ-дерево с точки зрения логики – обычный сложный предикат (дерево предикатов, дерево вывода, дерево решений);

- альтернативы в ИЛИ-узлах И-ИЛИ-дерева есть отдельные факты, истинные или ложные;

- И-узлы объединяют факты по И;

- правила вывода также объединяют предикаты (ИЛИ-узлы) по форме «если посылка истина, то истинен вывод»;

- т.о., с точки зрения логики, продукционного вывода, ничего необычного нет;

- в обычной логике пользователь сам определяет правила вывода, связывающие факты в соответствии с известной ему семантикой модели мира.

2) Специфические черты

Специфика нашей семантики модели мира:

- имеется явное разделение фактов на два класса: а) признаки, входящие в ТЗ; б) признаки, характеризующие тексты-прототипы; соответственно, может рассматриваться либо два взаимосвязанных И-ИЛИ-дерева, либо одно, но совмещающее факты двух классов;

- признаки, входящие в ТЗ, ориентированы на диалоговый вариант выбора условия;

- наряду с «обычными» рассматриваются «обобщенные» факты, как некоторая имеющая смысл комбинация частных, «обычных» фактов, т.е. терминальных текстов-прототипов; такая комбинация

задает, в зависимости от типа дерева, либо полное ТЗ, либо полное описание объекта, т.е. текст-прототип;

- любой «обычный» факт, входящий в И-ИЛИ-дерево, трактуется либо как принадлежность, либо – как не принадлежность данной особенности текста к определенному прототипу или – к ТЗ; т.о. факт определяет значение свойства прототипа;

- принадлежность признаков прототипам может определяться пользователем-экспертом по знаниям, например, определяя совместные и – несовместные признаки; принадлежность может определяться автоматически при обучении, путем прямого указания принадлежности данных свойств тем или иным прототипам.

Мы работаем с признаки как обычными предикатами, исключение – предикаты организации диалога синтеза (наличие ответов «все нет», «все да», «часть верна»).

Вывод: можно работать с И-ИЛИ-деревом как с обычным деревом вывода и оценивать сложность как сложность поиска решений по дереву предикатов.

Анализ дерева вывода и И-ИЛИ-дерева:

- обычное И-ИЛИ-дерево вывода есть совмещенное пространство обликов систем (ПОС), т.е. условия выбора в И-ИЛИ-дереве, с елевым пространством систем (ЦПС); ЦПС объединяет терминальные вершины, где описаны либо требуемые решения, объединенные по ИЛИ, либо части решений, объединенные по И; такое И-ИЛИ-дерево, где терминальными вершинами являются решения, обычно строят вручную;

- И-ИЛИ-дерево вывода, такое, как в ПРОЛОГЕ [8,9] или как у Донского, Войтицкого [10-12] - есть предикативное И-ИЛИ-дерево условий выбора, на терминальной вершине такого дерева реально нет описания объекта, оно только подразумевается или может затем отдельно строится; в ПРОЛОГЕ автоматически строится данное И-ИЛИ-дерево решений для поиска в нем цели, но – это автоматическое построение не прототипов, а условий к ним, что бы сравнить их с целью; результат такого построения – истина или ложь по отношению к цели;

- недостатком как обычного, так и предикативного И-ИЛИ-дерева вывода является проблема автоматического построения прототипа по набору признаков, описанных выше в И-ИЛИ-дереве; т.к. форма представления и детальность описания должна быть такая, что бы обеспечить интерпретацию модели в некоторой имитационной системе моделирования, т.е. описание должно быть формальным и

полным; при ручном построении дерева данные требования трудно обеспечить.

Оценка процессов вывода, обучения и уточнения постановки задачи в данном случае эквивалентна оценке сложности выполнения ТМО над КС порождающими грамматиками. Т.о., мы должны оценить сложность алгоритмов ТМО, принятых в МЭО.

2. Краткая характеристика алгоритмов выполнения теоретико-множественных операций, принятых в МЭО

Теоретико-множественные операции над грамматиками:

- на входе – любые две грамматики, соответствующие вышеописанным условиям, заданные своим начальным символом;
- на выходе ряд новых грамматик, содержащих только оригинальные идентификаторы новых синтермов; они предназначены для идентификации составляющих старых синтермов, вошедших как в общую часть, так и в каждую из двух отличных частей.

Результатом так же являются все комбинации объединения общей и двух отличных частей.

Характеристики алгоритма ТМО:

- задача алгоритма – сравнить две грамматики и построить ряд новых грамматик, соответствующих результатам ТМО – объединение, разница, дополнение, пересечение;
- алгоритм выполняет сравнение грамматик как двух множеств слов, т.е. текстов, представленных в сравниваемых грамматиках;
- две грамматики имеют разное назначение – первая это ведущая, и вторая - ведомая;
- ведущая грамматика просматривается по всем возможным вервям, а – ведомая просматривается многократным проходом вперед и назад с целью поиска совпавших слов или групп слов, заданных синтермами в текущем месте ведущей грамматики;
- при сравнении алгоритм делит исходные синтермы в обеих грамматиках на две части – общая (совпавшая), и - отличная, не совпавшая;
- создает новые оригинальные имена новых синтермов, как совпавших так и не совпавших (амперсенд четный и – амперсенд нечетный);
- обход обеих грамматик выполняется слева - направо и сверху (от корня) – вниз (к ветвям);
- представляет собой многократный проход по первому (ведущему) И-ИЛИ-дереву с целью нахождения соответствующих компонент во втором дереве;

- как для ведомого, так и для ведущего дерева есть прямой и обратный ход алгоритма;

- прямой ход – проверка некоторой ветви дерева;

- обратный – возврат к месту ветвления с параллельным формированием четного и нечетного амперсендов в каждом из пройденных ранее синтермов;

- прямой ход – разворачивание синтермов на компоненты по ходу просмотра;

- обратный ход – сворачивание развернутых ранее синтермов, при автоматическом создании новых синтермов – совпавшей и – не совпавшей части, на которые разбиваются «старые» ИЛИ-синтермы; каждая новая часть содержит подмножество альтернатив из старого ИЛИ-синтерма;

- используются стеки для хранения цепочки разложенных синтермов; знак «&» или «\/» в верхушке стека имеет смысл граничного символа разложения;

- синтерм считается не совпавшим, если хотя бы один его компонент не совпал;

- не совпавшие синтермы раскрываются далее, с целью поиска либо совпадающих синтермов, либо с целью деления нижележащего множество цепочек терминальных символов на совпавшую и не совпавшую часть;

Алгоритм является модификацией стандартного алгоритма грамматического разбора LL(1), в котором:

- на входе одна цепочка текста длиной N символов;

- одна читающая головка над лентой входной лентой;

- грамматика – это правила вывода, управляющие движением головки;

- на выходе – ответ «принадлежит ли данная цепочка к множеству синтаксически верных выражений, т.е. правильная она – или ошибочная».

Сложность измеряет:

- число поворотов головки;

- общее число просмотренных символов во входной ленте;

В нашем случае имеется две грамматики, и соответственно:

- две головки;

- две ленты - текущие;

- общее число лент с двух сторон – размер языков.

3. Оценка базы качества базы знаний, построенной на основе предлагаемого метода представления знаний

Все известные меры, которые возможно применить для оценки качества систем с базами знаний могут быть поделены на две группы:

1) Меры оценки механизмов логического вывода и структуры БЗ включают фактуальные оценки и технологические меры; например, фактуальные оценки включают: сложность БЗ, информативность БЗ, надежность вывода решения, достоверность выведенного решения, устойчивость БЗ, быстродействие системы представления и обработки знаний и т.д.

2) Меры оценки критериальных свойств для разработчиков - когнитологов и экспертов: релевантность знаний (их подтвержденность и достаточность), полнота умений (необходимость и достаточность процедур), проверенность знаний (их тестированность и целостность), уровень интеллектуальности (обучаемость, гибкость стратегий рассуждения и интерфейса), наличие мета-знаний и т.д.

Следует отметить громоздкость рассмотренной системы оценки качества БД. Вслед за [10-12] мы будем рассматривать только главные выходные показатели эффективности вывода в базе знаний, задающие эффективность D-алгоритма синтеза логической области дедуктивной выводимости (ЛОДВ), соответствующего по структуре нашему алгоритму, а именно:

- L(S) – длину логического описания ОДВ (число конъюнктов в ДНФ ЛОДВ);
- M(S) – число замещений литералов в процессе работы алгоритма.

При этом, L(S) задает пространственную сложностью, M(S) – временную сложностью алгоритма. Временная и пространственная сложность вывода есть функции, зависящие от аргументов, являющимися структурными показателями сложности базы знаний:

- H(S) – высота И-ИЛИ-дерева, т.е. наибольший ярус дерева (ярус целевой вершины полагается нулевым).
- N(S) – число продукций в ЛСП S;
- q(S) – число нетерминальных литералов в ЛСП S, увеличенное на 1;
- x(s) – число терминальных литералов ЛСП S;
- $k_{\vee}$ - наибольшая степень ИЛИ-вершин И-ИЛИ-дерева, соответствующего ЛСП S;
- $k_{\&}$ - наибольшая степень И-вершин И-ИЛИ-дерева, соответствующего ЛСП S.

Длина ЛОДВ, построенного D-алгоритмом (ленточная сложность), не превысит

$$L_D(S) \le k_{\vee}^{k_{\&}} * k_{\vee}^{(q(S)-H(S)+1)*(H(S)-2)}. \qquad (1)$$

Общее число построенных в процессе работы D-алгоритма конъюнктов не превысит (временная сложность):

$$M_D(S) \le k_{\vee} + k_{\vee}^{k_{\&}} * k_{\vee}^{\frac{k_{\vee}^{(q(S)-H(S)+1)*(H(S)-1)}-1}{k_{\vee}-1}}. \qquad (2)$$

Т.о., в [10-12] рассматривается типичное предикативное дерево, т.е. дерево условий. Крона – терминальные символы, могущие быть истинными или ложными. Все прочее – множество выражений. Имеются конечные, упорядоченные множества символов: терминальные, целевые (верхние), нетерминальные. Терминальные символы – так же упорядочены. Аксиомы – подмножества упорядоченных терминальных символов, имеющих значение 0 или 1. Правила вывода: если верно некоторое множество выражений над символами (по И), то верно выражение для данного отдельно взятого символа. Теорема – это доказуемая аксиома. Задается как проверка значения 0 или 1 некоторого символа (целевого, нецелевого или терминального). Теоремы и аксиомы – это факты. Доказать теорему – означает показать, что все факты (по И), входящие в терему – истинны.

В грамматической трактовке пространственная сложность – это ленточная сложность, временная сложность – так же временная сложность. В нашем случае - терминальный символ – это часть текста. Значение – 1, если он есть в цепочке, и - если 0, то его просто нет. Т.о., любая цепочка символов потенциально включает все терминальные символы, одни со значением 1 (есть), другие – со значением 0 (нет). Терминальные символы – упорядочены по кроне. Правило вывода – синтерм по И для правого символа (целевого, нетерминального, терминального). Альтернативные правила вывода для одного и того же правого символа – синтерм по ИЛИ. Имя правила вывода – имя синтерма по И. Имя правого символа – имя синтерма по ИЛИ, включает альтернативные правила вывода. Т.о., имеем И-ИЛИ-дерево. Доказать теорему – означает показать, что существует (не существует) данная цепочка терминальных символов. Доказательство может быть прямым – от аксиом к целям, и обратным – от цели к аксиомам. Список «открыт» в D-алгоритме синтеза ЛОДВ – это раскрытие синтерма, сначала по ИЛИ (если он ИЛИ), а затем по И (если он И). Закрытие списка – аналог сворачивания

просмотренного синтерма. Весь D-алгоритм – аналог лексического анализа, когда отыскивается некоторая фраза в грамматике. Если успех – теорема доказана, не успех – теорема не доказана. Совокупность правил – И-ИЛИ-дерево. Нетерминальный символ – подмножество И-ИЛИ-дерева. Если рассматривается нетерминальный символ, то это подмножество слов (имеют место ИЛИ узлы). Вопрос ставится так – есть ли хоть одна фраза из набора, заданного данным И-ИЛИ-деревом, соответствующим данному нетерминальному символу. Можно рассчитывать сложность как число операций над стеками – свернуть и развернуть.

Тогда рассмотренный выше подход может трактоваться как оценка простейшего случая ТМО над контекстно-свободными (КС) порождающими грамматиками, а именно - пересечение отдельно взятого текста с порождающей КС грамматикой. Оценки сложности для случая сравнения двух грамматик можно построить, исходя из следующих соображений. В наиболее сложном случае, когда пересечение двух грамматик пусто, процесс проверки можно трактовать как проверку всех возможных текстов из одной грамматики со второй грамматикой. Введем обозначения:

N1 – число текстов в первой грамматике;

N2 – число текстов во второй грамматике;

K(N2) – сложность проверки одного текста из первой грамматики со второй грамматикой;

K(N1,N2) - суммарная сложность проверки первой грамматики со второй грамматикой;

Тогда возможно дать следующую приблизительную оценку сложности сравнения двух грамматик:

$K(N1,N2) \geq N1*K(N2)$.

Тут K(N2) предполагается эквивалентной оценке сложности вывода из работ Донского и Войтицкого, т.е. соответствует соотношениям (1) и (2).

Заключение. В работе получены оценки качества баз знаний, построенных на основе предложенного метода представления знаний, заданные временной и ленточной сложностью, выраженных оценками (1) и (2). Следует отметить, что оценка (1) как и оценка (2) экспоненциально зависит от максимальной степени вершин типа «И» и высоты И-ИЛИ-графа, а оценка (2) еще экспоненциально зависит и от числа нетерминальных символов.

Т.о., сложность вывода в предлагаемом методе представления знаний является экспоненциальным, что является малопригодным для

практического применения в условиях больших баз знаний, что является недостатком данного метода.

Однако соотношения (1) и (2) могут рассматриваться как источник рекомендаций по ограничению структурных параметров логической системы продукций в целях построения базы знаний, допускающей эффективное решение задачи синтеза вывода данного факта. Следовательно, актуальной является задача построения средств и методов снижения сложности вывода в условиях предлагаемого метода представления знаний.

Литература

1. Григорьев А.В. Теоретико-множественные операции над грамматиками как механизм работы со знаниями в интеллектуальных САПР. Вісник Східноукраїнського національного університету імені Володимира Даля, N 2(48). Луганск, ВУТУ, 2002. С. 186-194.
2. Григорьев А.В. Алгоритм выполнения теоретико-множественных операций над грамматиками в среде специализированной оболочки для создания интеллектуальных САПР. Наукові праці національного технічного університета. Серия «Проблемы моделирования и автоматизации проектирования динамических систем» (МАП -2002). Выпуск 52: Донецк: ДонНТУ, 2002. - С.83-93.
3. Григорьев А.В., Каспаров А.А. И/ИЛИ-дерево как средство абстрактного представления знаний. Наукові праці національного технічного університета. Серия «Інформатика, кібернетика ти обчислювальна техніка». Выпуск 39: Донецк: ДонНТУ, 2002. - С.36-42.
4. Григорьев А.В. Семиотическая модель базы знаний САПР. Научные труды Донецкого государственного технического университета Серия: Проблемы моделирования и автоматизации проектирования динамических систем, выпуск 10: - Донецк, ДонГТУ, 1999. - С. 30-37.
5. Григорьев А.В. Принципы организации вывода решений в базе знаний инструментальной оболочки для создания интеллектуальных САПР. // Практика і перспективи розвитку інституційного партнерства». Вісник ДонГТУ – ТРТУ. Донецьк: РВА ДонНТУ, 2003 – С.96-106.
6. Григорьев А.В. Специфика выполнения теоретико-множественных операций над контекстно-свободными грамматиками в условиях различных форм дополнительных семантических правил в семиотической модели интеллектуальных САПР. Научные труды

ДонНТУ. Серия «Проблемы моделирования и автоматизации проектирования динамических систем». Выпуск 5 (116). Донецк: ДонНТУ, 2006. – С.91-104.

7. Математическая теория формальных языков / А.Е. Пентус, М.П. Пентус. – М.: Интернет-университет информационных технологий; БИНОМ. Лаборатория знаний, 2006. – 247 с.

8. Братко И. Программирование на языке Пролог для искусственного интеллекта. М.: Мир, 1990. – 560 с.

9. Стобо Д.Ж. Язык программирования Пролог. М.: Радио и связь, 1993. – 368 с.

10. Е.И. Войтицкий. Исследование алгоритмов логического вывода в продукционных системах и оценка их сложности. КДС – 96. Ялта, 1996. С. 163-172.

11. Донской В.И. Логические продукционные системы: анализ и синтез // Техническая кибернетика и системный анализ, 1994, N 4, С. 11-22.

12. Donskoi V.I. Psevdo-boolean optimization with a disjunctive constraint // Comp. Maths Phys., Vol.34, No. 2, pp. 389-398, 1994.

Получено 12.05.2007.

МОДУЛЬ ЗНАНИЙ КАК СРЕДСТВО ОГРАНИЧЕНИЯ СТРУКТУРНЫХ ПАРАМЕТРОВ БАЗЫ ЗНАНИЙ ИНСТРУМЕНТАЛЬНОЙ ОБОЛОЧКИ ДЛЯ ПОСТРОЕНИЯ ИНТЕЛЛЕКТУАЛЬНЫХ САПР

Григорьев А.В.
Донецкий национальный технический университет

Анализируются возможные методы снижения сложности организации процессов обучения и вывода в базе знаний в условиях специфики представления знаний в инструментальной оболочке для построения интеллектуальных САПР. Предлагается модификация данных методов, ограничивающих значения структурных показателей базы знаний с целью обеспечить приемлемую сложность процессов функционирования базы знаний.

Введение. В работах [1-6] были изложены основные принципы нового метода представлений знаний в базе знаний инструментальной оболочки для построения интеллектуальных САПР – мета-эвристической оболочки (МЭО). Данный метод представления знаний имеет ряд специфических особенностей: использование контекстно-свободных грамматик как основы представления знаний; сужение грамматик до уровня И-ИЛИ-деревьев; организация обучения, вывода и процесса уточнения задачи синтеза на основе теоретико-множественных операций (ТМО) над КС порождающими грамматиками и т.д.

В работе [6] изложена специфика предлагаемого метода представления знаний и получены оценки качества баз знаний, показавшие, что сложность вывода в предлагаемом методе представления знаний имеет экспоненциальную зависимость от структурных параметров базы знаний. Последний факт является недостатком данного метода, делая его малопригодным для практического применения в условиях больших баз знаний. Однако полученные оценки сложности вывода могут рассматриваться как источник рекомендаций по ограничению структурных параметров логической системы продукций в целях построения базы знаний, допускающей эффективное решение задачи синтеза вывода. Т.о., актуальной является задача построения средств и методов снижения

сложности вывода в условиях предлагаемого метода представления знаний.

Цель данной работы является:

- классификация средств и методов снижения сложности вывода;
- выбор и модификация средств под специфику данной задачи.

1. Анализ средств и методов, влияющих на структурные показатели базы знаний

Концептуальная модель предметной области МЭО (далее просто КМ), имеет две части, два этапа. Это:

- модель объекта проектирования, представляющая собой на практике комплекс взаимосвязанных моделей объекта различных уровней абстракции;
- модель собственно САПР, представляет собой набор процедур, определенных над моделью объекта проектирования, т.е. над системой моделей.

Задача логического вывода решается во второй части и представляет собой одну из процедур. Однако, наличие двух компонент КМ ставит следующие задачи:

- построение методов формирования модели объекта проектирования, косвенным путем положительно влияющих на сложность логического вывода;
- построение методов организации логического вывода, снижающих сложность процесса вывода.

При этом можно отметить, что методы формирования модели объекта проектирования влияет косвенным образом на структурные показатели модуля знаний, а построение методов логического вывода - влияет на них прямым образом. Рассмотрим детальнее возможные методы и средства, относящиеся к двум возможным путям решения поставленной проблемы.

1.1. Анализ средств и методов, влияющих косвенным образом на структурные показатели модуля знаний

Рассмотрим методы построения модели объекта проектирования, косвенным образом влияющие на сложность логического вывода. Среди них можно назвать:

1) Принцип блочного иерархического построения структурной модели объекта проектирования;

2) Принцип ограничения когнитивной сложности для отдельного структурного блока в иерархической структурной модели объекта;

3) Средства активного обеспечения когнитивной сложности блоков объектов не выше заданного уровня.

Выполним анализ упомянутых средств отдельно.

Принцип блочного построения структурной модели является классическим принципом построения модели. В его основе лежит интуитивный путь приведения модели к форме представления, удобной для понимания, что составляет достоинство принципа. В общем случае, деление модели на модули есть прерогатива пользователя, но такой произвол можно отнести и к недостаткам данного подхода.

Идея ограничения когнитивной сложности блоков в структурной модели так же не нова. Соответствующие методы позволяют уйти от интуитивного понимания когнитивной сложности к конкретным числовым оценкам. Можно назвать для примера технологию SADT, где действует принцип «все, о чем есть смысл говорить, должно быть представлено не более, чем из семи частей». Достоинство подхода:

- ориентация на построение структурных моделей;
- форма представления эффективная в том смысле, что позволяет ускорить процесс построения моделей объектов, а затем – и их понимание.

Недостатки подхода:

- отсутствие средств для приведения «плохих» моделей к оптимальному виду;
- отсутствие учета того факта, что построенные «оптимальные» модели могут быть синтезированы с помощью базы знаний и, таким образом, требуется обеспечить эффективную работу последней.

Средства обеспечения сложности объектов не выше заданного уровня позволяют решать проблему приведения «плохих» моделей к оптимальному виду. К ним можно отнести:

- задачу о рюкзаке в дискретной математике;
- задачу компоновки в САПР радиоэлектронной аппаратуры и т.п.

Достоинства:

- ориентация на структурную модель (касается задачи компоновки);

Недостатки:

- не учитывают специфику представления структурной модели в МЭО.

1.2. Анализ средств и методов, влияющих прямым образом на структурные показатели модуля знаний

Рассмотрим методы построения базы знаний о методах проектирования объекта, прямым образом влияющие на сложность логического вывода. Среди них можно назвать:

1) Средства модульного построения базы знаний, включая обеспечение процесса работы с совокупностью модулей знаний в процессе вывода;

2) Принцип разбиения заведомого слишком сложного И-ИЛИ-дерева решений на более мелкие И-ИЛИ-деревья с целью увеличения эффективности процессов вывода в базе знаний;

3) Средства ограничения числа решений, включенных в базу знаний.

Выполним анализ упомянутых средств отдельно.

Принцип модульного построения базы знаний не нов и применяется повсеместно. Для примера можно назвать инструментальную систему для построения экспертных систем CLIPS, имеющей полный набор средств работы с совокупностью модулей знаний, включая: формирование частичной видимости базы данных из тех или иных модулей; создание стека фокусов, т.е. указателей на модули знаний; средства работы со стеками модулей и т.д. Разбиение базы знаний на модули ориентировано на распараллеливание процесса вывода, на создание логически и семантически связанных потоков вывода. Достоинством данного принципа является то, что процесс вывода строится с учетом семантики предметной области, а сам процесс вывода становится прозрачнее и понятнее пользователю, а так же имеет более высокую эффективность. Недостаток: отсутствуют общие методики построения модулей знаний в базах знаний САПР.

Идея сведения большого сложного И-ИЛИ-дерева решений в процессе вывода к ряду маленьких деревьев с тем, что бы повысить эффективность базы знаний за счет снижения сложности процесса вывода, не нова – можно назвать, для примера, работы Донского, Войтицкого [8-10]. Недостатки данных работ:

- идея применялась не на задаче создания модулей знаний, а - в контексте динамического создания локальных И-ИЛИ-деревьев в так называемых слоях «большого» И-ИЛИ-дерева, с целью уйти от полного перебора всех альтернатив И-ИЛИ-дерева при выводе;

- работа имела форму чисто теоретической разработки, без практических приложений.

Достоинство: предлагался источник рекомендаций по ограничению структурных параметров логической системы продукций в целях построения базы знаний, допускающей эффективное решение задачи синтеза вывода данного факта.

Средства ограничения числа решений, включенных в базу знаний на взгляд автора, косвенно уже давно определены. Классические исследования, проведенные еще в свое время под руководством Поспелова Д.А. [11], были посвящены тому, какие функции от признаков использует человек в своей повседневной практике, когда он формирует новые понятия. В экспериментах было показано, что наиболее простыми квалификационными признаками являются конъюнкции признаков или дизъюнкции одиночных признаков. С функциями, содержащими 4-5 признаков, человек работает свободно, при возрастании числа квалифицирующих признаков сверх этого сложность задачи классификации начинает возрастать. Более сложными являются ситуации, когда необходимо применять одновременно операции конъюнкции и дизъюнкции. Наиболее сложным случаем классификационной функции является вариант не только наличия какого либо признака, но и факт отсутствия некоторого признака. Пример «нечеловеческой» классификационной функции: $y = x_1\overline{x_2} \vee \overline{x_1}x_2$, где x_1, x_1 - некоторые признаки. К достоинствам подхода можно отнести наличие оценок сложности некоторой совокупности решений для человека. Недостатки: отсутствие связи с содержимым базы знаний.

2. Решение задачи снижения сложности вывода в условиях предложенного метода представления знаний

Т.о., проблема ограничения структурных показателей модуля знаний распадается на две задачи:

- ограничение когнитивной сложности синтезируемого решения;
- ограничение числа склеиваемых решений в модуле знаний.

При этом можно отметить, что ограничение сложности синтезируемого решения влияет косвенным образом на структурные показатели модуля знаний, а ограничение числа склеиваемых решений - влияет прямым образом.

2.1. Ограничение когнитивной сложности модели объекта проектирования

Исходя из недостатков и достоинств рассмотренных выше методов косвенного снижения сложности логического вывода, автором был предложен ряд своих методов, позволяющих решить задачу ограничения когнитивной сложности модели объекта проектирования. К ним относятся:

- метод оценки уровня когнитивной сложности, приемлемой для данного класса пользователей на основе тестирования ряда объективных и субъективных характеристик пользователя [12-14];
- метод оценки уровня когнитивной сложности моделей, представленных на внутреннем формальном языке спецификаций МЭО [1];
- алгоритм активного обеспечения допустимой когнитивной сложности модели, представленной на внутреннем формальном языке спецификаций МЭО, являющийся решением модифицированной задачи о рюкзаке, но решенной в трактовке задачи компоновки [12-14].

Примечание 1. Принцип предельной когнитивной сложности в неявной форме применялся автором уже в системе проектирования микропроцессорных систем на основе микропроцессорного секционного комплекта Am2900 - системы EMULAT.

Т.о., предлагаемые средства ограничения и алгоритм снижения когнитивной сложности - это механизм, созданный для внутреннего формального языка спецификаций описания объектов проектирования в МЭО. Поясним суть данных методов, определив их достоинства и недостатки. Данный механизм необходим в случае обучения базы знаний методикам проектирования сложных объектов на основе ряда известных прототипов-решений. Если имеется одноуровневая структурная схема объекта большой сложности, полученная в практике проектирования, то данный алгоритм может, ориентируясь на требования пользователя, упростить ее до разумных пределов когнитивной сложности, построив новую иерархию вложенных подблоков. Порядок объединения алгоритмом подблоков в блоки, вообще говоря, в сильной степени зависит от специфики данной схемы. Допускается, что другая близкая, в понимании пользователя, по назначению исходная схема, но имеющая существенные отличия, может быть разбита алгоритмом по-другому. Т.о., нет 100%-й возможности получить ряд схем, в которых логически близкие блоки действительно близки по структуре. Следовательно, схемы, используемые для склеивания с целью создания модуля знаний, могут

породить не совсем корректную методику проектирования. Однако, данный недостаток неизбежен при малой степени компетентности эксперта, допускаемой общей методикой построения инструментальной оболочки.

Однако следует отметить, что в результате либо автоматического преобразования текстов к более когнитивно простой форме, либо - создания модели путем редактирования под управлением системы контроля предельной сложности, вместо одного сложного текста описания модели будут получена иерархия более простых текстов.

Покажем, что подобное действие благотворно влияет на показатели структурной сложности базы знаний. Общая суть процесса влияния такова: большие тексты, склеенные много раз, порождают И-ИЛИ-дерево, структурные параметры которого (высота, число дизъюнктов и т.д.) имеют мощность, меньшую или равную сумме значений структурных параметров совокупности И-ИЛИ-деревьев, порожденных склейкой тех же текстов, но разбитых на маленькие части-тексты. Т.е. много мелких И-ИЛИ-деревьев позволяют организовать вывод быстрее, чем одно большое И-ИЛИ-дерево.

Данная оценка верна, исходя из экспоненциальной формы зависимости сложности от структурных параметров. Т.о., оценка сложности совокупности локальных баз знаний меньше, чем сложность «большой» базы знаний.

Предлагаемое решение позволяет использовать средства пассивного контроля предельной когнитивной сложности схемы, определенной для данного класса пользователей, а так же алгоритм активного обеспечения допустимой когнитивной сложности для ограничения структурных параметров базы знаний с целью обеспечения допустимого уровня сложности вывода, а именно временной и ленточной сложности.

2.2. Ограничение числа склеиваемых решений

Предлагаемое решение:

1) Модуль знаний напрямую соответствует типу структурного блока, имеющемуся в структурной модели объекта проектирования и отвечает требованиям ограниченной когнитивной сложности.

2) При построении модуля знаний необходимо ограничивать число возможных решений, содержащихся в модуле знаний с тем, чтобы ограничить классификационную сложность множества решений, содержащихся в модуле знаний.

Прокомментируем второе решение. Независимо от ограничения на размер склеиваемых текстов, в том числе и - ограничению по когнитивной сложности, в процессе обучения базы знаний можно клеить любое количество текстов-моделей. От числа склеиваемых текстов и их специфики напрямую зависят основные структурные показатели модуля знаний, т.е. высота И-ИЛИ-дерева, число дизъюнктов и т.д.

Следовательно, ограничений по когнитивной сложности недостаточно, что бы получить базу знаний приемлемого качества. Т.о., идея ограничения когнитивной сложности объекта должна найти свое отражение в идее ограничения структурных показателей отдельного модуля знаний. Часть методики проектирования, ограниченная модулем знаний, должна быть прозрачна, доступна для понимания пользователя. Это означает, что:

1) любое отдельное синтезируемое решение должно быть обозримо;

2) число параметров, по которым производится синтез, их значений, т.е. фактически множество возможных отличий различных решений тоже должно быть когнитивно ограничено.

Первое требование обеспечивается разработанными ранее средствами ограничения когнитивной сложности. Второе требование не поддерживается в предлагаемом комплексе средств. Предлагается следующее новое решение, учитывающее специфику используемого метода представления знаний и использующее результаты классификации признаков, полученные группой Поспелова Д.А., рассмотренные выше.

Будем рассматривать простое и сложное решение задачи.

При этом важно отметить, что методика проектирования, не имеющая циклов, позволяет проектировать только ряд стандартных схем, без построения бесконечных регулярных составляющих.

Простейшее решение задачи таково. При отсутствии регулярности в схеме, пользуясь тем же принципом SADT, можно ограничить множество решений по мощности 6-7 решениями. Т.е. - клеить надо не более 6-7 близких по назначению решений. Таким путем, с учетом близости склеиваемых решений, а так же их когнитивной ограниченной сложностью, можно ожидать, что классификационная сложность готового модуля знаний будет ограничена.

В более сложном случае необходимо управлять процессом склеивания, оценивая на каждом шаге текущую классификационную сложность. Наша цель при этом - обеспечить для пользователя задание им в диалоге синтеза 6-7 параметров, сведенных в

конъюнкты. Оценить классификационную сложность процесса синтеза по модулю знаний означает выяснить, какого рода диалоговые вопросы и в каком количестве будут задаваться пользователю при синтезе решений. Порядок прохода по ИЛИ-узлам в ходе синтеза эквивалентен для пользователя объединению их логически по "И". При этом наличие или отсутствие некоторого структурного компонента в схеме трактуется как простейшее двузначное условие, объединенное по "И" с прочими условиями. Следовательно, для оценки классификационной сложности И-ИЛИ-дерева необходимо оценить наибольшую длину цепочки дизъюнктов, имеющуюся в И-ИЛИ-дереве, двигаясь от корня к кроне. Если длина цепочки более 7 - это сигнал для окончания "клейки" прототипов.

Если же тот или иной вариант структуры имеет регулярные составляющие, т.е. массивы, то они рассматриваются как отдельные модули знаний. Число вариантов размера массива, выбираемых в И-ИЛИ-дереве, должно быть конечным и заданным в некотором интервале. Такой подход обеспечит доступную классификационную сложность, поскольку речь будет идти об одном параметре, обладающим ограниченным числом значений.

Заключение. В работе получены следующие результаты:

1) Выполнена стыковка двух компонентов предложенной КМ МЭО - модели объекта и модели САПР, заданной как база знаний. Т.о., специфика модели объекта, ориентированного на САПР, нашла свое отражение в специфике базы знаний, ориентированной на задание методик проектирования в САПР. Данная стыковка и обеспечила общее повышение эффективности КМ.

2) Предложен новый комплекс из известных средств и методов, обеспечивающий решение новой задачи – задание оптимальной формы представления структурной модели, обеспечивающей повышение эффективности процессов вывода в базе знаний, предназначенной для синтеза данных структурных моделей. Практически все известные подходы и методы, применяемые в комплексе средств, претерпели модификацию вследствие новой специфики применения.

3) Предложен комплекс средств и методов, обеспечивающий решение задачи ограничения структурных показателей отдельного модуля знаний. Основой комплекса является метод управления процессом склеивания прототипов на основе динамической оценки классификационной сложности базы знаний. Предложенный метод

реализуется в условиях отсутствия массивов неограниченных размеров в базе знаний.

Перспективной задачей является практическая оценка степени снижения сложности вывода на основе применяемых методов.

Литература

1. Григорьев А.В. Теоретико-множественные операции над грамматиками как механизм работы со знаниями в интеллектуальных САПР. Вісник Східноукраінського національного університету імені Володимира Даля, N 2(48). Луганск, ВУТУ, 2002. С. 186-194.

2. Григорьев А.В. Алгоритм выполнения теоретико-множественных операций над грамматиками в среде специализированной оболочки для создания интеллектуальных САПР. Наукові праці національного технічного університета. Серия «Проблемы моделирования и автоматизации проектирования динамических систем» (МАП -2002). Выпуск 52: Донецк: ДонНТУ, 2002. - С.83-93.

3. Григорьев А.В., Каспаров А.А. И/ИЛИ-дерево как средство абстрактного представления знаний. Наукові праці національного технічного університета. Серия «Інформатика, кібернетика ти обчислювальна техніка». Выпуск 39: Донецк: ДонНТУ, 2002. - С.36-42.

4. Григорьев А.В. Семиотическая модель базы знаний САПР. Научные труды Донецкого государственного технического университета Серия: Проблемы моделирования и автоматизации проектирования динамических систем, выпуск 10: - Донецк, ДонГТУ, 1999. - С. 30-37.

5. Григорьев А.В. Принципы организации вывода решений в базе знаний инструментальной оболочки для создания интеллектуальных САПР. // Практика і перспективи розвитку інституційного партнерства». Вісник ДонГТУ – ТРТУ. Донецьк: РВА ДонНТУ, 2003 – С.96-106.

6. Григорьев А.В. Специфика выполнения теоретико-множественных операций над контекстно-свободными грамматиками в условиях различных форм дополнительных семантических правил в семиотической модели интеллектуальных САПР. Научные труды ДонНТУ. Серия «Проблемы моделирования и автоматизации проектирования динамических систем». Выпуск 5 (116). Донецк: ДонНТУ, 2006. – С.91-104.

7. Оценка эффективности инструментальной системы для создания интеллектуальных САПР как системы с базой знаний. В этом же сборнике.

8. Е.И. Войтицкий. Исследование алгоритмов логического вывода в продукционных системах и оценка их сложности. КДС – 96. Ялта, 1996. С. 163-172.

9. Донской В.И. Логические продукционные системы: анализ и синтез // Техническая кибернетика и системный анализ, 1994, N 4, С. 11-22.

10. Donskoi V.I. Psevdo-boolean optimization with a disjunctive constraint // Comp. Maths Phys., Vol.34, No. 2, pp. 389-398, 1994.

11. Поспелов Д.А. Ситуационное управление: теория и практика. М.: Наука, 1986. - 288 с.

12. Григорьев А.В. Оценка когнитивной сложности моделей. Научные труды Донецкого государственного технического университета. Серия: Информатика, кибернетика и вычислительная техника, выпуск 6: - Донецк: ДонГТУ, 1999. - С. 252-259.

13. Методика тестирования для определения когнитивной сложности моделей различных предметных областей. Научные труды Донецкого государственного технического университета. Серия: Информатика, кибернетика и вычислительная техника, выпуск 6: - Донецк: ДонГТУ, 1999. - С. 246 251.

14. Григорьев А.В. Ограничение когнитивной сложности моделей. В кн. Прогрессивные технологии и системы машиностроения:: Международный сб. научных трудов. - Донецк: ДонГТУ, 2000. Выпуск. 10 - С. 49-58.

Получено 12.05.07.

УДК 004.31

СПЕЦИАЛИЗИРОВАННЫЕ ПАРАЛЛЕЛЬНЫЕ ПРОЦЕССОРЫ ДЛЯ ОТСЛЕЖИВАНИЯ ПЕРЕМЕЩЕНИЙ ОБЪЕКТОВ В ВИДЕОПОТОКЕ

А. А. Середа, Ю.В. Ладыженский
Донецкий национальный технический университет

В статье рассмотрено применение специализированных параллельных процессоров для повышения скорости отслеживания объектов в видеопотоке, связь задачи отслеживания объектов с задачей определения движения при кодировании видео и подходы к созданию процессоров для отслеживания объектов.

Задача отслеживания перемещений объектов возникает при построении систем безопасности, контроля технологических процессов, организации человеко-машинного интерфейса и в других областях человеческой деятельности. Под отслеживанием подразумевается определение координат объекта или отдельных его частей в определенные моменты времени. Одним из подходов к решению данной задачи является автоматизированный анализ видеозаписи движения объектов.

Во многих алгоритмах отслеживания объектов в видеопотоке решается задача отслеживания перемещения фрагмента изображения, который слабо изменяется на протяжении некоторого времени и обладает особенностями, делающими его отличным от других фрагментов такой же формы [1, 2].

Пусть S_t – текущий кадр, S_{t-d} – некоторый предшествовавший ему кадр, $P_{t-d} = \{(x_i^{t-d}, y_i^{t-d})\}$ – множество координат пикселей отслеживаемого фрагмента в предыдущем кадре. Требуется найти фрагмент текущего кадра, состоящий из пикселей с координатами $P_t = \{(x_i^t, y_i^t)\}$, такой, что мощности множеств P_t и P_{t-d} равны, $x_i^t = x_i^{t-d} + dx$, $y_i^t = y_i^{t-d} + dy$ и различие между этими фрагментами минимально. Степень различия может определяться различными способами, часто используется сумма модулей разностей яркостей $SAD(P_t, P_{t-d}) = \sum_{(x,y)\in P_{t-d}} \left| S_t(x+dx, y+dy) - S_{t-d}(x,y) \right|$.

Искомый фрагмент P_t может быть найден путем перебора всех фрагментов в окрестности исходного фрагмента P_{t-d} и сравнения их с P_{t-d}. Такой перебор обладает большой вычислительной сложностью.

Другие методы поиска достигают большего быстродействия за счет снижения точности, используя 2 идеи [3, 4, 5]:

1) перебирать не все, а только некоторые фрагменты P_t;

2) сравнивать не все пиксели фрагментов, а лишь часть.

В быстрых алгоритмах поиск подходящего фрагмента является итеративным. Фрагменты, рассматриваемые на текущем шаге, определяются результатами сравнений на предыдущем шаге. Начальная позиция для поиска может быть предсказана на основе результатов анализ соседних блоков и предыдущего кадра.

Решение задачи поиска фрагментов может занимать большую часть вычислительного времени при отслеживании объектов. При этом поиск фрагмента сводится к малому числу простых операций над элементами двумерных массивов (нахождение абсолютной разности и сложение) и может быть эффективно реализован на специализированных параллельных процессорах.

Подобная задача (motion estimation) решается и при кодировании видео, в частности, по стандартам MPEG 1, 2 и 4. Для ее решения разработаны алгоритмы [3, 4, 5] и специализированные процессоры [5, 6]. Решение задачи при отслеживании объектов имеет следующие особенности:

1) фрагменты могут иметь произвольную форму и размер, в то время как при кодировании видео используются прямоугольные блоки со сторонами, являющимися степенью двойки;

2) для произвольно расположенных фрагментов P_{t-d} ищутся произвольно расположенные фрагменты P_t, в то время как при кодировании видео для расположенных в строго определенных местах блоков P_t ищутся произвольно расположенные блоки P_{t-d};

3) при отслеживании фрагментов в течение длительного времени не должна накапливаться ошибка в координатах фрагмента, что обычно не требуется при кодировании видео;

4) может потребоваться найти все подходящие фрагменты, в то время как при кодировании видео достаточно любого подходящего.

Поэтому процессоры, определяющие векторы движения при кодировании видео, плохо подходят для отслеживания перемещений объектов. В настоящее время в исследованиях, посвященных отслеживанию объектов, не уделяется достаточно внимания специализированным процессорам, поэтому актуальной является их разработка.

Типы специализированных процессоров, которые могут быть применены при отслеживании объектов, показаны на рис. 1.

Рис. 1 – Типы специализированных процессоров

Возможно построить систолический массив, выполняющий полный перебор фрагментов, имеющий полную загрузку вычислительных ячеек [7]. Основное достоинство подхода – простота и регулярность структуры.

Процессор, предназначенный для выполнения одного из алгоритмов сокращенного поиска, может выполнять параллельно сравнение различных элементов двух блоков. Для этого может быть использован систолический массив [5]. Однако, сравнения блоков, относящихся к разным шагам работы алгоритма, должны выполняться последовательно, т. к. от результатов сравнений зависит направление дальнейшего поиска. Достоинствами подхода по сравнению с предыдущим являются низкое энергопотребление и меньшие аппаратные затраты [6]. Недостатком является меньшая точность.

При использовании готовых программируемых сигнальных процессоров производительность на единицу стоимости будет наименьшей [5]. При этом не нужно разрабатывать и производить новое аппаратное обеспечение.

Для повышения быстродействия можно использовать несколько однотипных процессоров, работающих параллельно. При этом возможны различные архитектуры вычислительной системы (связь через общую шину, процессорная матрица, гиперкуб).

Для учета особенности 1) можно добавить в имеющиеся архитектуры процессоров, рассчитанные на поиск прямоугольных блоков, поддержку битовой маски или матрицы весовых коэффициентов для пикселей искомого блока. Тогда процессоры смогут работать с блоками произвольной формы, что позволит использовать более точные алгоритмы отслеживания, но может привести к увеличению аппаратных затрат.

Матрица весовых коэффициентов может быть использована при реализации алгоритма, подобного описанному в [2].

Во многих алгоритмах отслеживания используется метод вычитания фона (background subtraction) [8]. Использование при сравнении фрагментов кадров битовых масок, определяющих

принадлежность пикселей кадра к фону или объекту, может повысить точность, однако привести к повышению аппаратных затрат.

Для учета особенности 3) можно:

1) определять перемещения фрагментов с субпиксельной точностью;

2) сравнивать блоки, находящиеся в кадрах, разделенных большим временным промежутком (длина которого лимитирована временем, в течение которого изменения отслеживаемого фрагмента малы).

Создание и применение специализированных параллельных процессоров, предназначенных для отслеживания перемещения участков изображения в видеопотоке, реализующих указанные подходы, позволит существенно повысить скорость решения задачи.

Литература

[1] А.А. Середа, Ю.В. Ладыженский. *Компьютерный анализ видеозаписей футбольного матча: гибридный алгоритм отслеживания движения объектов.* // Сб. трудов II Междунар. студ. науч.-тех. конф. "Информатика и компьютерные технологии 2006".
URL: http://cs.dgtu.donetsk.ua/studconf/2006/CS_2006_Proceedings.pdf

[2] R. Cucchiara, C. Grana, G. Tardini. *Track-based and object-based occlusion for people tracking refinement in indoor surveillance.* Proceedings of the ACM 2nd international workshop on Video surveillance & sensor networks 2004, New York, NY, USA, October 15, 2004.
URL: http://doi.acm.org/10.1145/1026799.1026814

[3] M. Rehan, P. Agathoklis, A. Antoniou. *Flexible Triangle Search Algorithm for Block-Based Motion Estimation. EURASIP Journal on Advances in Signal Processing*, vol. 2007, Article ID 28782, 14 pages, 2007.
URL: http://www.hindawi.com/GetArticle.aspx?doi=10.1155/2007/28782

[4] A. Hamosfakidis, Y. Paker. *A Novel Hexagonal Search Algorithm for Fast Block Matching Motion Estimation. EURASIP Journal on Applied Signal Processing,* vol. 2002, no. 6, pp. 595-600, 2002. URL:
http://www.hindawi.com/GetArticle.aspx?doi=10.1155/S111086570220311X

[5] Sheu-Chih Cheng, Hsueh-Ming Hang. *A Comparison of Block-Matching Algorithms Mapped to Systolic-Array Implementation. IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, no. 5, pp. 741-757, 1997.
URL: http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=633491

[6] *Adaptive Motion Estimation Processor for Autonomous Video Devices.* T. Dias, S. Momcilovic, N. Roma, L. Sousa. *EURASIP Journal on Embedded Systems,* vol. 2007, Article ID 57234, 10 pages, 2007.
URL: http://www.hindawi.com/GetArticle.aspx?doi=10.1155/2007/57234

[7] Систолические структуры: Пер. с англ. / Под ред. У. Мура, Э. Маккейба, Р. Уркхарта. – М.: Радио и связь, 1993. – 416 с.: ил.

[8] Y. Chen, C. Han, X. Kang. *Background Information Fusion and its Application in Video Target Tracking.*
URL: http://www.fusion2004.foi.se/papers/IF04-0747.pdf

Получено 01.06.07

УДК 681.3

ПОВЫШЕНИЕ ПРОИЗВОДИТЕЛЬНОСТИ ИНТЕРНЕТ-ОРИЕНТИРОВАННОЙ СЕТЕВОЙ ИНФРАСТРУКТУРЫ: МОДЕЛИ, АЛГОРИТМЫ И МЕТОДЫ.

С. В. Рычка, А. Я. Аноприенко, Хасан Аль Абабнех
Донецкий национальный технический университет

Розглядаються комбіновані методи підвищення продуктивності Інтернет-орієнтованої мережної інфраструктури шляхом використання моделі маршрутизатора зі застосуванням алгоритму керування чергами та механізму явного повідомлення про перенавантаження.

Совсем недавно абсолютное большинство корпоративных и университетских сетей состояло из небольшой группы хостов и сервера, осуществляющего доступ к глобальной сети и предоставляющего все Интернет-сервисы. Типичным современным вариантом является разветвленная инфраструктура с множеством сетевого активного оборудования, объединяющего сотни и тысячи хостов, разнесенных территориально [1]. Практически экспоненциально продолжает увеличиваться число пользователей, работающих как в глобальной сети, так и в корпоративных и университетских сетях [2]. Появление корпоративных сетей глобального масштаба ознаменовало собой новый этап в развитии сетевой инфраструктуры, который настоятельно требует существенных качественных изменений в подходе к управлению перегрузками и повышению производительности Интернет-ориентированной сетевой инфраструктуры [3].

Под Интернет-ориентированной инфраструктурой (ИОИ) будем понимать корпоративную либо научно-образовательную компьютерную сеть, активно использующую ресурсы сети Интернет. Участников обмена данными в сети Интернет можно разделить на: транзитные узлы передачи данных, к которым относятся магистральные, национальные и региональные Интернет-провайдеры (Internet Service Provider - ISP), и конечные пользователи или автономные системы, между которыми и происходит обмен данными. Как правило, ISP имеют избыточную пропускную способность относительно конечных пользователей, и их основной задачей является обеспечение качества обслуживания (Quality of Service - QoS) в предоставлении канала связи гарантированной пропускной

способности для ИОИ. Таким образом, ISP, в отличие от ИОИ, в меньшей мере сталкиваются с проблемой перегрузок и уменьшением производительности сети. Для ИОИ “узким местом” продолжает оставаться канал связи с ISP.

Проблема перегрузки в канале связи с ISP с использованием стека протоколов TCP/IP возникает задолго до того, когда количество передаваемых данных начинает приближаться к значению допустимой пропускной способности [4]. Помимо пропускной способности канала связи, на производительность ИОИ в значительной степени влияют потери пакетов, следствием чего являются повторные передачи пакетов [5]. Увеличение размера очередей в маршрутизаторах позволяет уменьшить число потерянных пакетов вследствие перегрузки, но приводит к увеличению задержек, что также негативно сказывается на производительности. На рис. 1 представлена модель QoS-маршрутизатора, обычно находящегося на границе между ИОИ и ISP, на который также возлагаются функции управления перегрузкой в канале связи с ISP.

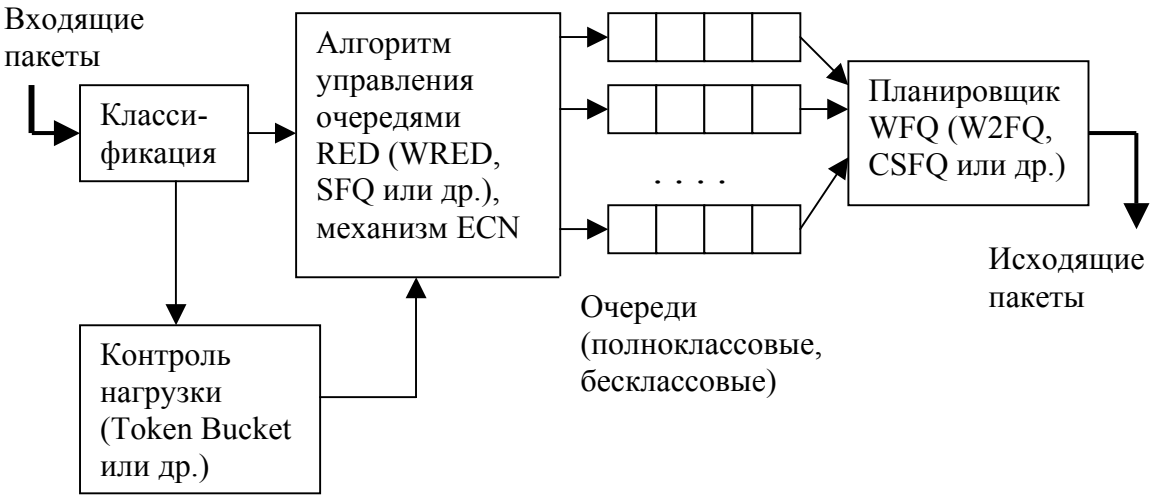


Рис. 1. Модель QoS-маршрутизатора.

После поступления в QoS-маршрутизатор, пакет проходит процедуру классификации, для определения принадлежности к соответствующему классу обслуживания в дальнейшем. Затем, после возможного прохождения стадии измерения и контроля нагрузки с помощью алгоритма Token Bucket или др., пакет обрабатывается алгоритмом управления очередями RED (Random Early Detection – Вероятностное заблаговременное определение перегрузки) [6] или др. Также пакет может быть обработан с помощью механизма TCP ECN

(Explicit Congestion Notification – явное уведомление о перегрузке) [7,8], если приемник и передатчик его поддерживают. Далее пакет попадает в очередь, соответствующую классификации. Выборка из очередей с разными классами обслуживания происходит с помощью планировщика обслуживания пакетов WFQ (Weight Fair Queuing – взвешенная справедливая буферизация) или др. Потоки данных, принадлежащие к различным классам обслуживания, выбираются по очереди, в соответствии с указанным QoS для каждого из классов, и направляются по указанному маршруту.

В данной схеме, с точки зрения методов повышения производительности, нас интересуют алгоритмы управления очередями и механизм ECN. Первоочередной задачей повышения производительности является недопущение перегрузки в канале связи между ИОИ и ISP. Так как основная доля трафика ИОИ приходится на входящий трафик, а возможность непосредственного влияния на скорость поступления входящего трафика отсутствует, это осложняет ситуацию. Но при использовании QOS-маршрутизаторов эта задача, в основном, успешно решается с помощью алгоритмов управления очередями, которые определяют, какой пакет и с какой вероятностью необходимо отбросить для предотвращения возникновения или развития перегрузки. Одним из самых распространенных алгоритмов управления очередями является алгоритм RED. При помощи алгоритма RED осуществляется вероятностный сброс пакетов, принадлежащих различным соединениям, при возникновении перегрузки, что позволяет соблюдать принцип справедливого распределения ресурсов и предотвращает от волнообразного увеличения нагрузки. Сброс пакета заставит источник нагрузки уменьшить размер окна перегрузки cwnd (Congestion window) и уменьшить нагрузку на сеть.

Однако, справившись с перегрузкой в канале связи за счет сброса пакетов, мы получаем новую проблему, связанную с необходимостью повторных передач данных, что существенно снижает производительность сети. Плюс ко всему, после сброса пакета TCP-сессия начинает работать в режиме медленного старта, который, несмотря на свой экспоненциальный характер, также снижает производительность за счет частичного простоя канала до момента, пока значение cwnd не сравняется с пороговым значением алгоритма медленного старта ssthresh (slow start threshold) и не начнется линейное увеличение окна перегрузки, которое будет расти, пока не сравняется с окном получателя rwnd (Receiver window) или пока снова не произойдет потеря пакета. Таким образом, метод

отбрасывания пакетов с помощью алгоритма управления очередями является неявным уведомлением о перегрузке и не позволяет поддерживать производительность ИОИ на высоком уровне. Для повышения производительности ИОИ необходимо свести к минимуму как потерю пакетов на всем пути их следования, так и сброс пакетов с помощью алгоритма управления очередями. Этого можно добиться, используя явное уведомление о перегрузке.

Новый механизм TCP ECN, предоставляющий явное уведомление о перегрузке, в первую очередь призван управлять нагрузкой соединений TCP, чувствительных к задержкам и потерям пакетов. При этом не используется сброс пакета для подавления источника передачи данных в качестве индикации перегрузки. В случае использования ECN, источник должен быть проинформирован о перегрузке заранее, то есть до ее наступления, а не по факту самой перегрузки, чтобы успеть снизить скорость передачи данных, не дожидаясь потери или сброса пакета. Информация о приближении перегрузки передается QoS-маршрутизатором приемнику. Приемник, в свою очередь информирует источник о перегрузке, и источник должен уменьшить значение cwnd и ssthresh так же, как будто бы произошла потеря пакета или его сброс без использования механизма ECN. На данный момент одновременное использование механизма ECN и алгоритмов управления очередями позволяет эффективно управлять перегрузками. Так как не все хосты поддерживают использование механизма ECN, QoS-маршрутизатор будет обрабатывать возникновение перегрузки по-разному: используя явное уведомление о перегрузке посредством механизма ECN или сброс пакета с помощью алгоритма управления очередью. Повышение производительности ИОИ при использовании механизма ECN, по сравнению со сбросом пакетов, достигается за счет отсутствия необходимости в повторных передачах, но не решает проблему неэффективного использования пропускной способности за счет уменьшения размера cwnd до одного максимального размера сегмента (Maximum Segment Size - MSS) и повторения процедуры медленного старта.

В итоге можно сделать следующий вывод: повысить производительность можно за счет совместного использования явного уведомления о перегрузке, реализованного в механизме ECN и адаптивно регулируя значение окна перегрузки, что позволит при возникновении краткосрочной перегрузки не снижать до минимума скорость передачи TCP-соединения и повысить пропускную способность на 15-30% в зависимости от типа трафика.

Литература

1. Аноприенко А. Я., Джон С. Н. Рычка С. В. Особенности моделирования и оценки эффективности работы сетевой инфраструктуры. – Наукові праці Донецького державного технічного університету. Серія: Обчислювальна техніка та автоматизація. Випуск 38. – Донецьк: РВА ДонДТУ, 2002. – с. 205-210.
2. Аноприенко А. Я., Рычка С. В., Хасан Аль Абабнех. Способы и средства моделирования вычислительных сетей с целью обеспечения эффективности функционирования web-сервисов. – Моделирование и компьютерная графика: Материалы 1-й международной научно-технической конференции, г. Донецк, 04-07 октября 2005 г. – Донецк, ДонНТУ, Министерство образования и науки Украины, 2005. – с. 156-159.
3. Аноприенко А. Я., Джон С. Н. Рычка С. В. Моделирование университетской сетевой инфраструктуры. – Вісник Кременчуцького державного політехнічного університету: Наукові праці КДПУ. – Кременчук: КДПУ, 2001. – Вип. 2(11). – с. 306-308.
4. Аноприенко А. Я., Рычка С. В. Комбинированные подходы к управлению перегрузками в сетях TCP/IP. – Наукові праці Донецького національного технічного університету. Серія: "Проблеми моделювання та автоматизації проектування динамічних систем" (МАП-2006). Випуск 5(116). – Донецьк: ДонНТУ, 2006. – с. 53-65.
5. Gerla M., Kleinrok L. "Flow Control: A Comparative Survey". IEEE Transactions on Communications, vol. 28 (4), 1980, pp. 553-574.
6. Floyd S., Jacobson V., "Random Early Detection gateways for Congestion Avoidance", IEEE/ACM Transactions on Networking, vol. 1, no. 4, August 1993, p. 337-341.
7. Floyd S., "TCP and Explicit Congestion Notification", ACM Computer Communication Review, V. 24 N., October 1994, p. 10-23, http://www-nrg.ee.lbl.gov/papers/tcp_ecn.4.pdf
8. Ramakrishnan K., Floyd S., Black D. The Addition of Explicit Congestion Notification (ECN) to IP. RFC3168, September 2001, http://www.ietf.org/rfc/rfc3168.txt?number=3168.

Получено 01.06.07

УДК 543.422:621.384.3

ВИКОРИСТАННЯ ІСНУЮЧИХ ЗАСОБІВ МОДЕЛЮВАННЯ ЕЛЕКТРОННИХ СХЕМ ПРИ ЗАПРОВАДЖЕННІ ДО УЧБОВОГО ПРОЦЕСУ ЕЛЕКТРОННИХ ПІДРУЧНИКІВ З НАПРЯМКУ „ЕЛЕКТРОНІКА”

Зорі А.А., Тарасюк В.П.
Донецький національний технічний університет,
кафедра електронної техніки
E–mail: Vita_post@mail.ru

Вступ. Згідно Державної програми «Інформаційні та комунікаційні технології в освіті і науці» на 2006-2010 роки у Донецькому національному технічному університеті прийнято заходи по створенню та впровадженню у навчальний процес програмних засобів, а саме: електронного підручника « Аналогова схемотехніка та імпульсні пристрої» для підготовки бакалаврів та магістрів напрямів навчання «Електроніка», «Телекомунікації», „Комп’ютерні науки”, „Приладобудування” та інших споріднених напрямів. Підручник розраховано для самостійного вивчення дисциплін студентами денної, заочної та дистанційної форм навчання.

Електронний підручник «Аналогова схемотехніка ті імпульсні пристрої», складається з 2-х частин, і відповідає трьом профілюючим дисциплінам, а саме: електронні системи; мікроелектроніка; функціональна електроніка.

Постановка задачі проектування. Електронний підручник буде містити комплекти віртуальних лабораторних робіт з комплектами запитань, програмами для перевірки отриманих результатів вимірювань та обробки даних, тощо [1].

Математичний опис досліджуваного процесу може бути представлено у вигляді аналітичного співвідношення, алгоритму, таблиці, графіка є так званою математичною моделлю. Така модель дозволяє, у разі адекватності об'єкту, одержати ті ж результати, що і при проведенні натурних експериментів. Адекватність моделі дуже часто витримується в певному діапазоні властивостей об'єкту, оскільки побудова універсальної моделі дуже часто складна і невиправдана. На практиці складаються моделі, що відображають домінуючі характеристики об'єкту [2-4].

Рішення задач і результати дослідження. Проблема моделювання електронних схем є проблемою автоматизованого проектування, тобто освітлення однієї із задач проектування – постановку за-

дачі і методи розрахунку і аналізу електронних схем.

В даний час складність і відповідальність схем, що розробляються, вже давно перейшла той поріг, який визначався можливостями так званого «ручного» розрахунку, тобто розрахунку, здійснюваного розробником на основі свого особистого досвіду і умінням володіти законами Ома і калькулятором.

Уявимо собі об'єм робіт, що вимагається для дійсно серйозного розрахунку електронної схеми середнього ступеня складності. При виконанні типового розрахунку з електротехніки в схемі, що містить 5 – елементів, було необхідно розрахувати характер перехідних процесів. Для цього складалося характеристичне рівняння ланцюга, проводився розрахунок початкових умов, потім записувалися аналітичні вирази для струмів і напруг в окремих елементах схеми, і, нарешті, проводилася табуляція одержаних розрахункових співвідношень. На цю роботу йде приблизно 20 годин.

Реальна задача розробника набагато складніша. Навіть якщо проектується подібний простий ланцюг, то необхідно додатково оцінити амплітудні і діючі значення струмів і напруг в схемі, для визначення меж застосовності тих або інших елементів. Потім уточнити їх еквівалентні схеми, що приблизно в 3–4 рази підвищить порядок схеми, провести повторний уточнений розрахунок, знов перевірити застосовність елементної бази і при необхідності внести корективи. Потім необхідно виконати конструкторське опрацьовування вузлів пристрою, і, можливо, знов внести корективи у зв'язку з технологічними особливостями виготовлення виробу.

І якщо до того ж врахувати, що всі реальні схеми як мінімум на порядок складніші, то створюється враження, що їх взагалі спроектувати не можна. Звичайно, в цьому питанні багато в чому допомагає інтуїція розробника, проте, покладаючись тільки на інтуїцію одержати в результаті проектування електронну схему адекватну початковим даним дуже важко. Це тим більше важко, якщо врахувати що для сучасних схем однією з основних вимог є висока надійність і постійне поліпшення енергетичних, економічних і інших, часто суперечливих показників. Виходом з такого положення може бути тільки автоматизація проектування, що дозволяє істотно скоротити час розробки і уникнути істотних помилок проектування.

В цілому проектування є комплексом робіт по дослідженню, дослідженню, аналізу, розрахунку, оптимізації, конструкторському і технологічному забезпеченню нового виробу або ефективнішого прототипу. Теоретично в результаті проектування повинен бути виданий комплект конструкторської документації, що дозволяє повністю виго-

товити і налагодити нову розробку в умовах серійного виробництва. У принципі, теоретично можна уявити собі можливість проведення такої роботи на ЕОМ, без участі людини, і навіть випуску готової продукції, відповідної заданому технічному завданню (ТЗ). Проте складність сучасної електронної техніки така, що до практичного рішення такої задачі поки що дуже далеке.

Існують САПР, що дозволяють автоматизувати проектування для окремих уніфікованих блоків і вузлів, проте комплексне рішення проблеми найближчим часом не передбачається. Напевно, в цьому є і здоровий глузд, бо, повністю автоматизувавши виробництво, ми замкнемося на готовому наборі стандартних рішень, що приведе до випуску морально застарілої техніки. Проте як вельми ефективний інструмент в руках інженера, САПР здатні забезпечити рішення десятків сучасних задач. У нашу задачу входить вивчення автоматизованих методів аналізу і моделювання електронних схем, які є необхідною ввідною частиною до розкриття проблеми автоматизованого проектування [2].

Не дивлячись на те, що задачі проектування на різних етапах різні, можна скласти загальну схему проектування, показану на рис. 1.

Не дивлячись на те, що фізичні закони описуються цілим рядом відомих співвідношень, на практиці не вдається на їх основі одержати безпосередньо математичну модель. Перешкодою є складність сучасних інтегральних мікросхем (БІС – 105, СБІС – 106 і більш). Тому, разом з математичними моделями, застосовують системи моделювання схемотехніки, які дозволяє одержати необхідні характеристики.

Так, наприклад, системи моделювання схемотехніки Electronics Workbench, MicroCAP-8, Orcad, DesignLab та ін [1-5]. Весь представлений клас систем призначений для моделювання і аналізу електричних схем. Система може проводити аналіз схем на постійному і змінному струмах. При аналізі на постійному струмі визначається робоча точка схеми в сталому режимі роботи. Результати цього аналізу не відображаються на приладах, вони використовуються для подальшого аналізу схеми. Аналіз на змінному струмі використовує результати аналізу на постійному струмі для отримання лінеарізованих моделей нелінійних компонентів. Аналіз схем в режимі АС може проводитися як в тимчасовій, так і в частотній областях.

У Electronics Workbench можна досліджувати перехідні процеси при дії на схеми вхідних сигналів різної форми. Програма також дозволяє виробляти аналіз цифро-аналогових і цифрових схем великого

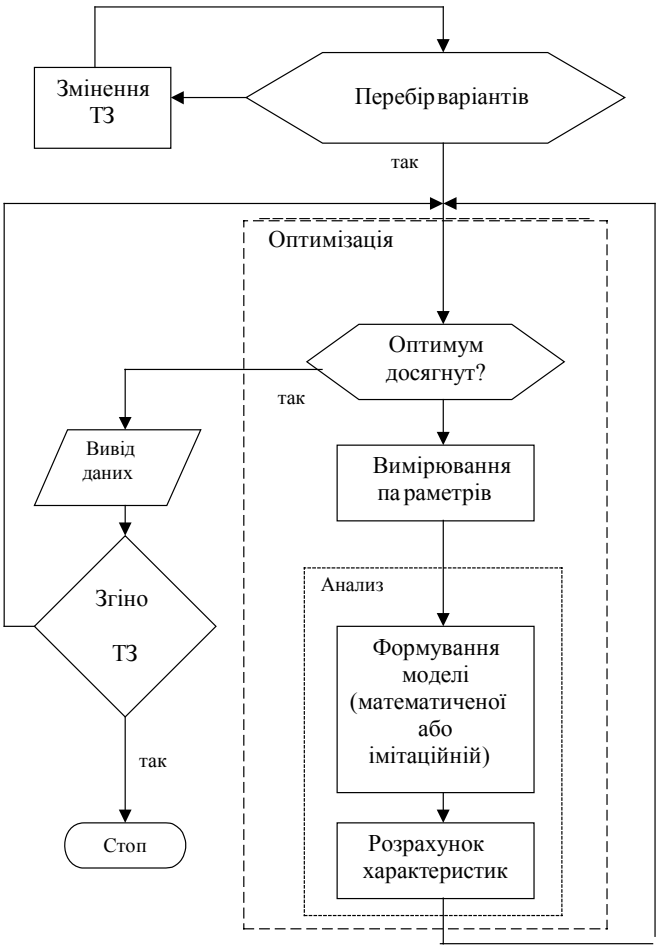


Рис. 1 – Блок схема формування моделі

ступеня складності. Бібліотеки, що є в програмі, включають великий набір широко поширених електронних компонентів. Є можливість підключення і створення нових бібліотек компонентів.

Результати моделювання можна вивести на принтер або імпортувати в текстового або графічного редактора для їх подальшої обробки. Програма Electronics Workbench сумісна з програмою P-SPICE, тобто надає можливість експорту і імпорту схем і результатів вимірювань в різні її версії.

У бібліотеки компонентів програми входять пасивні елементи, транзистори, керовані джерела, керовані ключі, гібридні елементи, індикатори, логічні елементи, пристрої трігерів, цифрові і аналогові елементи, спеціальні комбінаційні і послідовні схеми. Активні елементи можуть бути представлені моделями як ідеальних, так і реальних елементів. Можливе також створення своїх моделей елементів і додавання їх в бібліотеки елементів.

У програмі використовується великий набір приладів для проведення вимірювань: амперметр, вольтметр, осцилограф, мультиметр,

Боде-плотер (графічний пристрій частотних характеристик схем), функціональний генератор, генератор слів, логічний аналізатор і логічний перетворювач. Широкий набір приладів дозволяє виробляти вимірювання різних величин, задавати вхідні дії, будувати графіки. Всі прилади зображаються у вигляді, максимально наближеному до реального, тому працювати з ними просто і зручно [5].

Можливо також і змішане моделювання аналого-цифрових електронних пристроїв, реалізоване повною мірою досвідченим користувачем пакету, здатним в нестандартній ситуації створювати власні макромоделі, що полегшують імітаційне моделювання без втрати істотної інформації про поведінку системи.

Висновки.

1. Використання комп'ютерного моделювання у складі електронних підручників у навчальному процесі дозволить студентам усіх форм навчання, а саме: денної, заочної та дистанційної, самостійно оволодіти знаннями з аналогової електроніки та імпульсних пристроїв.

2. Застосування розроблених програмних продуктів, використаних у електронних підручниках, а також застосування дешевих у виробництві електронних носіїв інформації замість дорогих паперових, дозволить отримати значний економічний ефект від впровадження результатів розробки у навчальному процесі вищих навчальних закладів України.

Література.

1. Разевиг В.Д. Система сквозного проектирования электронных устройств Design Lab 8.0. – Москва, «Солон», 1999.

2. Разевиг В.Д. Система схемотехнического моделирования Micro–Cap 6. – М.: Горячая линия–Телеком, 2001. – 344 с., ил.

3. Влах М. Сингхал. К. Машинные методы анализа и проектирования электронных схем. – М.: радио и связь, 1988.–560с.

4. Калабеков Б.А. и др. Методы автоматизированного расчета электронных схем в технике связи. Учебное пособие для ВУЗов.–М.: Радио и связь, 1990.– 272 с.

5. Кардишев Г.А. Виртуальная электроника. Компьютерное моделирование аналоговых устройств. М.: Телеком, 2002. – 260с

Отримано 14.05.07

УДК 543.422:621.384.3

МОДЕЛЮВАННЯ ВИМІРЮВАНЬ КОНЦЕНТРАЦІЇ МЕТАНУ ДЛЯ УМОВ ВУГІЛЬНИХ ШАХТ ДОНБАСУ

Вовна О.В., Хламов М.Г.
Донецький національний технічний університет,
кафедра електронної техніки
E–mail: Vovna_Alex@ukr.net

Вступ. Одним з основних напрямів «Програми науково-технічного розвитку Донецької області на період до 2020 року» передбачено нарощування видобутку вугілля на основі технічного переозброєння вугільних шахт і впровадження прогресивної технології. Рішення цієї задачі пов'язане з подальшим підвищенням концентрації гірських робіт і інтенсифікацією процесу вуглевидобування, а також з прискореним переходом забоїв на глибші горизонти. Із зростанням глибини розробок збільшуватимуться газоносність вироблень, температура пластів і вміщуючих порід, а також частота і інтенсивність газодинамічних проявів.

Для вирішення питань створення безпечних умов праці гірників у поєднанні із зростаючою потребою у видобутку вугілля на пластах, небезпечних по раптових викидах вугільного пилу і газу, необхідне детальне вивчення газодинамічних явищ, що неможливе без широких експериментальних досліджень, що спираються на ефективні робочі засоби виміру. Серед таких засобів важливе місце займають швидкодіючі інформаційно-вимірювальні системи отримання даних про процеси зміни концентрації метану в атмосфері підземних вироблень вугільних шахт.

Отримані результати досліджень [1] дозволяють визначити початкові данні для розроблення швидкодіючого вимірювача концентрації метану системи газового захисту вугільної шахти:

– величину установки по метану слід приймати в 2 рази менше нижньої межі вибухонебезпеки метану – 2,5 %;

– значення допустимого часу спрацьовування і відключення електроенергії складає 0,8 с [2] по об'ємній частці метану і 2,0 с при швидкодії наростання концентрації метану 0,5 % в секунду.

Постановка задачі проектування. Для побудови швидкодіючого вимірювача об'ємної концентрації метану в умовах вугільних шахт необхідно вирішити наступні основні завдання:

– розробити математичну модель спектру поглинання метану в інфрачервоної області спектру;

– розробити імітаційну математичну модель оптичних і аналітичних каналів інфрачервоного оптико-абсорбційного газоаналізатора для визначення концентрації метану у вугільних шахтах;

– розробити на основі отриманих результатів імітаційного моделювання багатоканальні мікропроцесорні інфрачервоні оптико-абсорбційні газоаналізатори для контролю об'ємної концентрації метану, що відповідають вимоги європейських стандартів для умов вугільних шахт Донбасу.

Рішення задач і результати дослідження. Метод, покладений в основу побудови швидкодіючих інформаційно-вимірювальних систем вимірювання об'ємної концентрації метану, заснований на здатності атмосферних газів вибірково поглинати інфрачервоне випромінювання в спектральній області, і відноситься до бездисперсійних методів абсорбційної спектроскопії.

Відповідно до експериментальних даних про молекулярне поглинання спектр поглинання метану має складну структуру. Для ІЧ-області характерні так звані коливально-обертальні смуги поглинання, що є послідовностями ліній поглинання, контур кожною з яких визначається міжмолекулярними зіткненнями і описується виразом [3]:

$$K_i(\nu) = \frac{a_i}{\pi} \cdot \frac{\delta_i}{(\nu - \nu_{0i})^2 + \delta_i^2}, \qquad (1)$$

де $a_i = \int_0^\infty K_i(\nu) d\nu$ – інтегральна інтенсивність і-ої лінії; $\delta_i = 0{,}18\ см^{-1}$ – напівширина обертальної лінії; ν_{0i} – центр лінії.

Спектр структури носить характер довго- і короткохвильових не повністю симетричних крил при центральній обертальній складовій коливально-обертальні смуги поглинання, що є послідовностями ліній поглинання, контру кожною з яких визначається міжмолекулярним зіткненням і описується коефіцієнтом поглинання $K(\lambda)$ смуги, який виражається сумою коефіцієнтів поглинання окремих ліній:

$$K(\nu) = \sum_{i=1}^{\infty} K_i(\nu) = \sum_{i=1}^{\infty} \frac{a_i}{\pi} \cdot \frac{\delta_i}{(\nu - \nu_{0i})^2 + \delta_i^2}., \qquad (2)$$

Перетин поглинання для найбільш інтенсивних ліній молекулярного метану в області 3,31 мкм складає близько $1{,}8 \cdot 10^{-18}$ $см^2$/молекула. Для обчислення інтегральної інтенсивності лінії, цю величину потрібно помножити на число Авогадро ($6{,}022142 \cdot 10^{23}$ $моль^{-1}$, число частинок (атомів, молекул, іонів) в 1 молі речовини). Таким чином, інтегральна інтенсивність, визначається у відповідності:

$$a_i = K(\nu_i = \nu_{0i}) \cdot \pi \cdot \delta_i \cdot N_A, \left[\frac{см}{моль}\right]. \quad (3)$$

Для спрощення розрахунків [3] була розглянута модель смуги поглинання, в якій рівновіддалені лінії рівної δ мають розподіл інтенсивності в кожній з гілок смуги, відповідне розподілу молекул по обертальних рівнях енергії. На рис. 1 приведені графіки функції, обчислені для смуги поглинання метану з використанням прийнятої моделі.

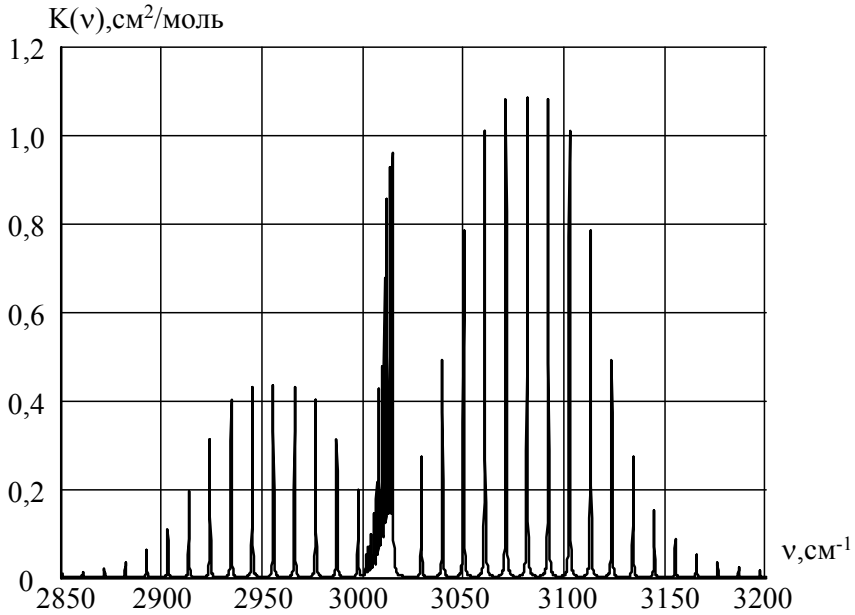


Рис. 1 – Спектр поглинання метану

У загальному вигляді, поглинання випромінювання в газі описується законом Бугера-Ламберта-Бера, який пов'язує поглинання зразка з його товщиною і концентрацією досліджуваного компоненту газової суміші і записується в наступному вигляді:

$$-\ln\left(\frac{I}{I_0}\right) = \ln\left(\frac{I_0}{I}\right) = K(\nu) \cdot C \cdot l = D(C, l, \nu), \quad (4)$$

де I_0 – інтенсивність або енергія падаючого випромінювання, I – інтенсивність минулого випромінювання, C – концентрація активною компоненти речовини, що поглинає випромінювання; l – одиниця товщини (довжина траси, кювети); $K(\nu)$ – показник поглинання; $\ln\left(\frac{I_0}{I}\right)$ – зазвичай називають оптичною щільністю речовини D.

При вимірюванні об'ємної концентрації метану $C_{об\%}$, $[об\ \%]$ потрібно скористатися формулою перерахунку з об'ємної концентрації газового компоненту в молярну концентрацію C_M, $\left[\frac{моль}{см^3}\right]$:

$$C_M\left(C_{об\%},T,P\right)=\frac{C_{об\%}\cdot P}{100\cdot R\cdot T}\cdot 10^{-6}=\frac{16{,}04\cdot 10^{-8}\cdot C_{об\%}\cdot P}{T}, \qquad (5)$$

де P – загальний тиск газової суміші в Па; T – температура в ^{0}К; $R=8{,}314472$ – універсальна газова постійна в Дж/(моль·К).

Таким чином, оптичну щільність об'ємної концентрації метану можна розрахувати, скориставшись співвідношенням:

$$D\left(C_{об\%},l,T,P,\nu\right)=K\left(\nu\right)\cdot C_M\left(C_{об\%},T,P\right)\cdot l. \qquad (6)$$

Коефіцієнт передачі оптичного каналу – K_{OK} (див. рис. 2), можна представити наступною залежністю:

$$K_{OK}\left(C_{об\%},l,T,P,\nu\right)=e^{-D\left(C_{об\%},l,T,P,\nu\right)}. \qquad (7)$$

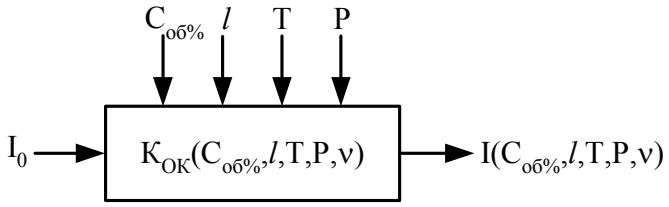
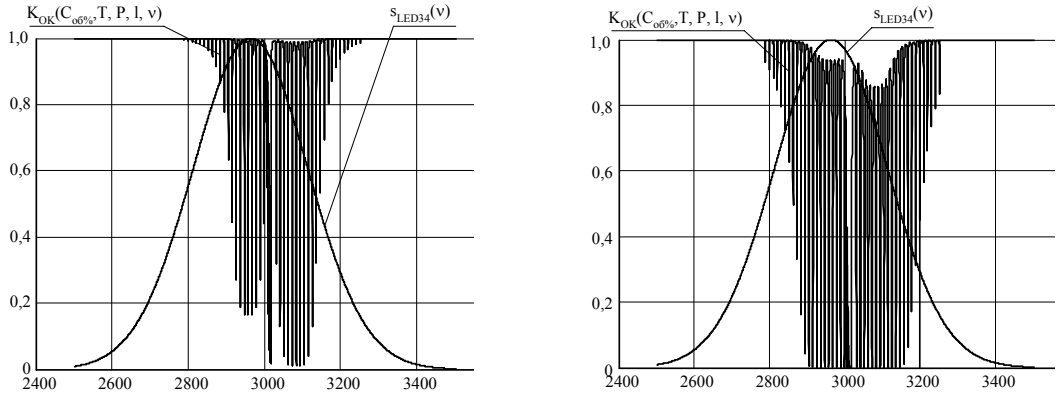


Рис. 2 – Коефіцієнт передачі оптичного каналу

Як джерело інфрачервоного випромінювання застосовується світловипромінювального діод LED34 [4] з довжиною хвилі випромінювання 3,35 мкм. Нормована спектральна характеристика світловипромінювального діода LED34 при температурі + 22 ^{0}С приведена на рис. 3. Також на рис. 3 приведені графіки функції коефіцієнта передачі оптичного каналу, обчислені для вибраної смуги поглинання метану з використанням прийнятої моделі.



$C_{об\%}=0{,}1\ об\%$ $\qquad$ $C_{об\%}=1{,}25\ об\%$

Рис. 3 – Коефіцієнта передачі оптичного каналу $K_{OK}\left(C_{об\%},l=100\ см,T=293^0\ K,P=101325\ Па,\nu\right)$ і нормована спектральна характеристика LED34 при температурі + 22 ^{0}С

У імітаційній математичній моделі, що розробляється, врахована переважна більшість чинників, що впливають на проходження інфрачервоного випромінювання по кюветах газоаналізатора. Оптична неоднорідність каналу викликана застосуванням в каналі матеріалів з різними оптичними властивостями. Оптична неоднорідність враховується коефіцієнтами заломлення. Втрати випромінювання на межах розділу і в середовищах з різними коефіцієнтами заломлення враховуються коефіцієнтами віддзеркалення і пропускання межами розділу, розсіяння і поглинання випромінювання в цих середовищах.

У цій роботі такі середовища виступають оптичні лінзи, що виготовляються із спеціальних матеріалів: спеціальних сортів оптичного скла, природних і синтетичних кристалів. Дані лінзи вважаються тонкими (їх товщина і діаметр багато менше фокусних відстаней) в зв'язку, з чим втратами на розсіяння і поглинання можна нехтувати. З цієї ж причини вважатимемо, що потоки випромінювання падають на межі розділу середовищ по нормалі до поверхні розділу.

У діапазоні малих кутів падіння проміння на межу розділу, коефіцієнт віддзеркалення практично залишається постійним величиною. Тому мале відхилення кута падіння на межу розділу обумовлене кривизною лінзи практично не впливає на коефіцієнт віддзеркалення. Джерело випромінювання має кінцеві розміри і потік, що виходить з лінзи виявляється таким, що розходиться, тому при визначенні коефіцієнта введення випромінювання в об'єктив враховується кут розбіжності проміння.

На основі приведеної математичної моделі побудована імітаційна модель вимірювача об'ємної концентрації метану, що враховує вплив комплексу обурюючих чинників, і запропоновані технічні рішення, що забезпечують підвищення точності і оперативності отримання вимірювальної інформації. Принциповим є конструкторське рішення, що реалізовує методи побудови вимірювача, які забезпечують вимірювання середнього значення концентрації метану, по довжині вимірювальної траси.

При цьому проблема впливу пилу в робочому вимірювальному каналі на результати вимірювання вирішується методами компенсації шляхом введення апаратурної надмірності. Для компенсації впливу рудничного пилу на результат вимірювання концентрації метану у вимірювачі введені два просторових і в них по два частотні канали [5].

Висновки.

1. Розроблена математична модель спектру поглинання метану в інфрачервоної області спектру.
2. Розроблена імітаційна математична модель оптичних і аналітич-

них каналів інфрачервоного оптико-абсорбційного газоаналізатора для визначення концентрації метану у вугільних шахтах;

3. Методи математичного і імітаційного моделювання, а також сучасні оптичні і мікроелектронні засоби дозволяють розробити багатоканальні мікропроцесорні інфрачервоні оптико-абсорбційні газоаналізатори для контролю об'ємної концентрації метану, що відповідають вимогам європейських стандартів для умов вугільних шахт Донбасу.

Література.

1. Волошин Н.Е. Внезапные выбросы и способы борьбы с ними в угольных шахтах. – Киев.: Техника, 1985. – 127 с.

2. ГОСТ 24932–80. Приборы шахтные газоаналитические. Общие технические требования. Методы испытания.

3. Бреслер П.И. Оптические абсорбционные газоанализаторы и их применение. Л.: Энергия, 1980. – 164 с.

4. Ioffe Physico-Technical Institute [Електронний ресурс] / Mid-IR Diode Optopair Group (MIRDOG), Dr. Boris Matveev, Web: Maxim Remennyi. – Електронні дані, Режим доступу: http://mirdog.spb.ru. – Загл. з екрану. – Мов. рос. англ.

5. Вовна А.В., Хламов М.Г. Алгоритм компенсации влияния пыли на точность измерения концентрации метана в угольной шахте. Наукові праці Донецького національного технічного університету. Серія: «Обчислювальна техніка та автоматизація» Випуск 107. – Донецьк: ДонНТУ, 2005. – 164 – 170 с.

Отримано 14.05.07

УДК 621.3

ПОГРЕШНОСТИ ПРОСТРАНСТВЕННЫХ МОДЕЛЕЙ ФУНКЦИЙ ПРЕОБРАЗОВАНИЯ ЧУВСТВИТЕЛЬНЫХ ЭЛЕМЕНТОВ ДАТЧИКОВ ДАВЛЕНИЯ

С.И.Клевцов
Технологический институт
Южного федерального университета в г. Таганроге

В статье приводятся оценки погрешностей измерения давления с использованием различных моделей пространственных аппроксимаций функций преобразования чувствительных элементов датчиков давления, учитывающих влияние внешних факторов

Достижение низкой погрешности измерения физической величины в интеллектуальном датчике-измерителе базируется на использовании модели его градуировочной характеристики, адаптированной к особенностям поведения функции преобразования чувствительного элемента (ЧЭ).

Для обработки сигналов, поступающих с чувствительного элемента, в интеллектуальном датчике используются различные модели его градуировочной характеристики, которые, в основном, базируются на пространственном представлении функции преобразования чувствительного элемента [1, 2, 3]. Характерным для этих моделей является использование единого представления градуировочной характеристики для всей области изменения полезного сигнала и внешних факторов (области определения градуировочной характеристики).

Наиболее простой моделью градуировочной характеристики датчика, широко используемой для градуировки аналогового датчика является кусочно-линейная пространственная аппроксимация [2].

Кусочно-линейная аппроксимация в интеллектуальном датчике возможна при достаточно линейной функции преобразования чувствительного элемента и соответствующих требованиях к точности измерений. Количество идентификационных точек, используемых для аппроксимации зависит от требуемой точности. Для обеспечения погрешности не выше *0,5* в работе [2] был выбран массив, содержащий *51* идентификационную точку поверхности $Z = f(P,T)$. Число калибровочных точек по оси *P* выбрано равным 17, а по оси *T* составляет три точки: $T_{мин}$, $T_{ср}$, $T_{макс}$. Использование предложенной авторами [2] модели кусочно-линейной аппроксимации позволило уменьшить погрешность, связанную с нелинейностью функции

преобразования с 5% до 0,05%, а температурную погрешность с 2,5% до 0,1% (на 100С).

Несмотря на существенное снижение отдельных составляющих погрешности вычисления давления, следует отметить, что на диапазоне в 60^0С изменений температуры (при проведении измерений в диапазоне температур -40^0С ÷ $+80^0$С и $T_{ср}=20^0$С) только температурная составляющая погрешности может составить величину порядка 0,6%, что не позволяет рекомендовать данный метод для прецизионных вычислений. Тем не менее, его использование целесообразно и оправдано, когда к датчику не предъявляются повышенные требования по точности.

Для проведения более точных измерений в настоящее время для аппроксимации градуировочной характеристики широко используются полиномы. Давление представляется полиномиальной функцией сигнала канала давления аналоговой части датчика [1,3]:

$$P_m = \sum_{i=0}^{n} a_{im} U_m^i , \qquad (1)$$

где P_m – давление при температуре Tm;

U_m – выходной сигнал с чувствительного элемента датчика.

$$a_{i_m} = \sum_{k=0}^{s} \gamma_{i_k} T_m^k , \qquad (2)$$

где a_{im}, γ_{ik} - коэффициенты аппроксимации.

Как правило, используются полиномы второго порядка *(n=s=2)*.

Для упрощения вычислений обычно формируется единое представление градуировочной характеристики для всего диапазона измеряемой физической величины и внешнего воздействующего фактора и, следовательно, определяется один набор коэффициентов аппроксимации.

В работе [1] показано, что компенсация характеристик пьезорезисторного кремниевого сенсора давления с использованием аппроксимации обратной функции передачи сенсора полиномом второго порядка по давлению и температуре позволила снизить максимальную погрешность измерения по отношению к характеристике сенсора в идеальном случае до ±0,2%, а в температурном диапазоне от 0 до 70°С – до значений менее ±0,1%. Температурная зависимость напряжения смещения нуля при этом находится в границах ±0,2% полного диапазона. Температурная компенсация уменьшила температурную зависимость чувствительности в 3,16 раза по отношению к некомпенсированному сенсору, чувствительность увеличилась 1,65 раза, максимальная нелинейность характеристики уменьшена в 26 раз.

Однако, следует отметить, что указанный результат был достигнут при использовании довольно значительного массива исходных данных в примерно 10000 точек, что неприемлемо даже при производстве небольшой партии датчиков.

Задача достижения прецизионной точности измерения значения физической величины в интеллектуальном датчике может быть решена при использовании предложенной в [4] мультисегментной модели градуировочной характеристики с линейными или нелинейными пространственными элементами. В рамках предлагаемой модели реальная градуировочная кривая заменяется системой локальных поверхностей так, чтобы они в совокупности повторяли ее пространственной конфигурацию. Тип аппроксимирующей локальной поверхности определяется многими факторами, одними из наиболее значимых являются требуемые точностные характеристики, объем и полнота исходных данных.

С позиций простоты и скорости вычислений, что важно при работе датчика в реальном времени, предпочтение следует отдать модели локальной поверхности в виде плоскости.

Исследование возможностей пространственной плоскостной модели градуировочной характеристики проводилось с использованием данных испытаний датчика давления, имеющего в своем составе первичный преобразователь давления (канал давления) и первичный преобразователь температуры (канал температуры). Эталонные давление и температура, подаваемые на датчик при его испытаниях, находились в пределах 0 – 80 кгс/см2 и -4^{0}С − +35^{0}С . Относительная погрешность данных испытаний для канала давления составляет погрешность не выше 0,05%, для канала температуры - не выше 0,07%. График $P=P(U_P)$ имеет небольшую нелинейность, которую можно оценить в 2,1% во всем диапазоне изменения P.

Абсолютная погрешность определения давления при использовании пространственной характеристики в виде системы плоскостей в среднем не превышает 0,014 кгс/см2 при диапазоне значений входного сигнала давления от 0 кгс/см2 до 80кгс/см2.

Учитывая разброс исходных данных в ~ 0,05% можно говорить о погрешности определения давления за счет неточности построения градуировочной характеристики в размере < 0,03%.

При существенной нелинейности данных испытаний в системе параметров $\{P, U_p, U_T\}$ пространственную форму градуировочной характеристики целесообразно формировать в виде системы нелинейных пространственных элементов.

Эта модель градуировочной характеристики была исследована на примере датчика абсолютного давления. Исходными данными служат

результаты испытаний при изменении давления в диапазоне от 0 до 60 кгс/см2 с шагом в 6 кгс/см2. Температура соответствовала одному из следующих значений: –40°С, –20°С, 0°С, 25°С, 40°С, 60°С, 80°С.

В качестве модели нелинейного пространственного элемента был использован полином второй степени. При разбиении области определения градуировочной характеристики рассматривались: разбиение на участки размером [18кгс/см2×400С] (1-я схема разбиения) и на участки размером [24кгс/см2×600С] (2-я схема разбиения).

Результаты приведены на рис. 1 и 2.

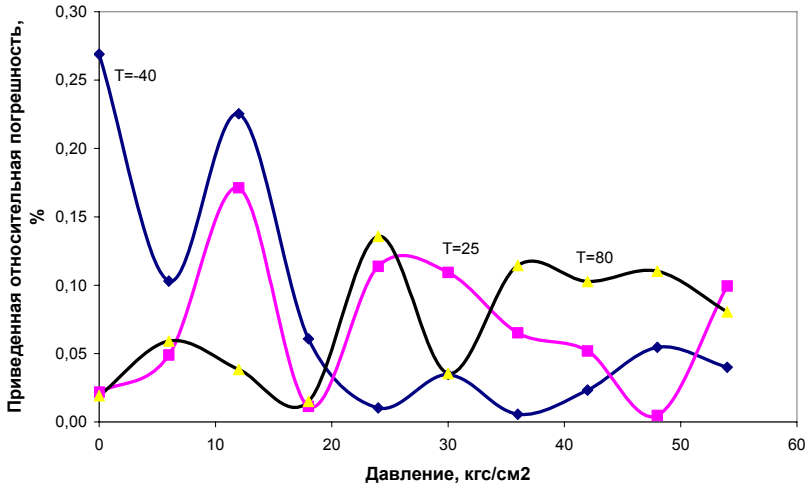


Рис. 1 – Относительная погрешность определения абсолютного давления для первой схемы разбиения

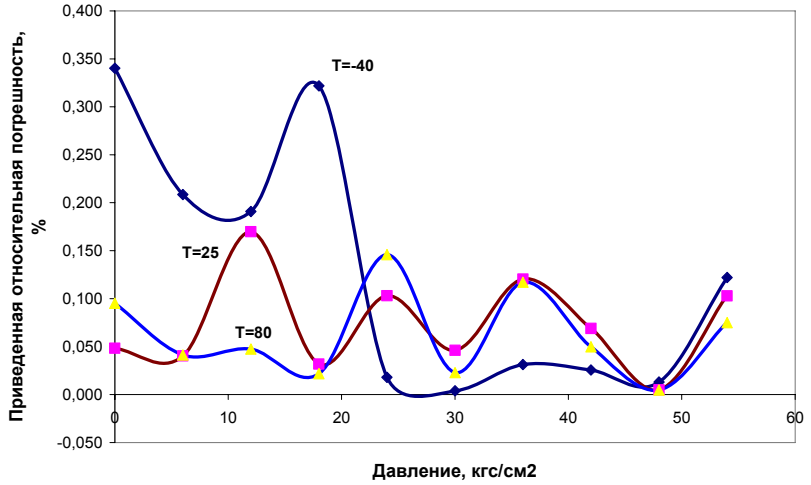


Рис. 2 – Относительная погрешность определения абсолютного давления для второй схемы разбиения области определения градуировочной характеристики

Полученные данные показывают высокую точность аппроксимации градуировочной характеристики в обоих случаях разбиения области определения характеристики. Тем не менее, чем

меньше размеры участков, тем более точный результат дает реализация метода.

Погрешность измерения давления, как показывают результаты компьютерного моделирования, для низких температур максимальная при низких давлениях и понижается более чем в 3 раза при увеличении давления. С увеличением температуры разница между максимальным и минимальными значениями погрешности уменьшается (рис.1 и 2). При высоких температурах наблюдается рост погрешности при увеличении подаваемого давления, однако разница между максимумом и минимумом погрешности не столь велика, как при низких температурах.

Анализ результатов показывает, что модель аппроксимации градуировочной характеристики системой локальных пространственных элементов обеспечивает высокую точность вычисления значений физической переменной во всем диапазоне воздействующих факторов при произвольной нелинейности данных градуировочных испытаний датчика. При незначительной нелинейности эффективна аппроксимация локальными плоскостными элементами. В случае существенной нелинейности реальной градуировочной характеристики эффективная аппроксимация возможна при использовании нелинейных локальных элементов.

Литература

1. Шапонич Д., Жигич А. Коррекция пьезорезистивного датчика давления с использованием микроконтроллера.//Приборы и техника эксперимента. - 2001. - №1. – С. 54-60.

2. Бобровников Н.Р., Яркин С.В., Гридин Ю.Н., Стрыгин В.Д., Чертов Е.Д. Математическое обеспечение микропроцессорных преобразователей аналоговых пневматических сигналов.//Приборы и системы. Управление, контроль, диагностика. – 2002.- №2 – С.36-39.

3. Клевцов С.И., Линьков В.С., Веретельников Ю.А., Кузьминов В.Г. Погрешности вычисления давления в интеллектуальном датчике при матрично-полиномиальной аппроксимации его градуировочной характеристики// Известия ТРТУ. 2004. №2. – С.30-48.

4. Клевцов С.И., Клевцова А.Б. Мультисегментная пространственная модель градуировочной характеристики интеллектуального датчика. //Материалы международной научной конференции "Цифровые методы и технологии". Ч.4. - Таганрог: Изд-во "Антон", ТРТУ, 2005. – С.21-26.

5. Клевцов С.И. Модели и методы построения прецизионных градуировочных характеристик интеллектуальных датчиков давления.// Известия ТРТУ. 2007. №3. – С.110-118.

Работа поступила в редакцию 04.05.2007.

УДК 621.313.333.018

МОДЕЛИРОВАНИЕ ТОКОВ И ТЕМПЕРАТУР АСИНХРОННЫХ ДВИГАТЕЛЕЙ ПРИ РАЗЛИЧНЫХ ХАРАКТЕРИСТИКАХ НЕСИММЕТРИИ СЕТИ: МЕТОДИЧЕСКИЕ ПОДХОДЫ

О.Г. Пинчук
Донецкий национальный технический университет

Моделирование теплового состояния асинхронных двигателей в среде программирования MATLAB и ее расширения Simulink позволяет прогнозировать температуры обмоток электродвигателей и исключить их тепловую перегрузку в тяжелых режимах работы, одним из которых является эксплуатация при несимметричном напряжении сети.

Асинхронные двигатели (АД) широко используются в промышленности (их доля среди всех двигателей составляет 85-90%) [1]. Ежегодно на долю электродвигателей приходится 25-30% общего числа повреждений электрооборудования. В подавляющем большинстве случаев 85-95% отказы двигателей происходят из-за повреждения обмоток [1]. Одной из причин перерегрева обмоток АД является несимметрия питающего напряжения [1].

В качестве характеристик несимметрии напряжения сети принято использовать [2, 3]: коэффициент несимметрии обратной последовательности ($К_{2U}$), напряжения прямой последовательности ($U_{пр}$) и сдвиг по фазе между симметричными составляющими напряжения прямой и обратной последовательности: $\Delta\psi = \psi_{пр} - \psi_{обр}$.

Влияние сдвига по фазе $\Delta\psi$ на тепловое состояние АД было рассмотрено в работе [4]. Так, в [4] показано, что при $К_{2U}$ = const и изменении $\Delta\psi$ температура наиболее нагруженной фазы статора изменяется в широких пределах. При некоторых значениях $\Delta\psi$ температуры обмоток статора достигают наибольшего ($\Delta\psi = \Delta\psi_{max}$) и наименьшего ($\Delta\psi = \Delta\psi_{min}$) значения. Расчет электромеханических и тепловых характеристик при проектировании АД для наиболее тяжелого случая эксплуатации (при $\Delta\psi = \Delta\psi_{max}$) позволит исключить тепловые перегрузки, связанные с изменением $\Delta\psi$ при эксплуатации АД. Поэтому разработка методики расчета токов и температур при $\Delta\psi = \Delta\psi_{max}$ является актуальной. Значительный интерес представляют эксплуатационные характеристики АД в случае одновременного изменения параметров несимметрии питающего напряжения ($U_{пр}$, $К_{2U}$) и $\Delta\psi = \Delta\psi_{max}$. Требуют исследования особенности АД различного типоразмера при несимметричном напряжении сети

Целью данной работы была оценка теплового состояния АД различного типоразмера (мощностью до 400кВт) в продолжительном режиме при различных характеристиках несимметрии питающего напряжения ($U_{пр}$, $К_{2U}$, $\Delta\psi = \Delta\psi_{max}$).

Для аналитических исследований при анализе установившихся режимов работы АД в условиях несимметрии источника питания был использован метод симметричных составляющих с применением Т-образной схемы замещения [5]. Математическое моделирование АД выполнялось в среде программирования MATLAB-Simulink [7]. Согласно Т-образной схеме замещения для прямой и обратной последовательности симметричные составляющие токов статора ($I_{1пр}$, $I_{1обр}$) и ротора ($\underline{I}'_{2пр}$, $\underline{I}'_{2обр}$) равны [5]:

$$\underline{I}_{1пр} = \frac{\underline{U}_{пр}}{\underline{Z}_{пр}}; \qquad \underline{I}_{1пр} = \frac{\underline{U}_{пр}}{\underline{Z}_{пр}}, \tag{1}$$

$$\underline{I}'_{2пр} = \underline{I}_{1пр}\frac{\underline{z}_0}{\underline{z}_0 + \underline{z}'_{2пр}} \quad \underline{I}'_{2обр} = \underline{I}_{1обр}\frac{\underline{z}_0}{\underline{z}_0 + \underline{z}'_{2обр}}. \tag{2}$$

где $\underline{z}_{прi} = \underline{z}_1 + \frac{\underline{z}_0 \cdot \underline{z}_{2s\,прi}}{\underline{z}_0 + \underline{z}_{2s\,прi}}$, $\underline{z}_{обрi} = \underline{z}_1 + \frac{\underline{z}_0 \cdot \underline{z}_{2s\,обрi}}{\underline{z}_0 + \underline{z}_{2s\,обрi}}$, - сопротивления Т-образной схемы замещения для прямой и обратной последовательности.

Токи в обмотках ротора при несимметрии имеют несинусоидальный характер [5, с.446] и не зависят от $\Delta\psi$. Их действующие значения определяются по формулам [6, с.207]:

$$I_{2A} = I_{2B} = I_{2C} = \sqrt{I'^2_{2\,пр} + I'^2_{2\,обр}}. \tag{3}$$

Результирующие токи в обмотках статора в значительной степени зависят от величины $\Delta\psi$ [4], что связано с соответствующим изменением угла между векторами симметричных составляющих токов. Так, из уравнений (1) угол сдвига симметричных составляющих тока статора ($I_{1пр}$, $I_{1обр}$) относительно симметричных составляющих напряжений ($U_{пр}$, $U_{обр}$) равен:

$$\psi_{I_{1пр}} = \psi_{U_{пр}} - \psi_{z_{пр}}; \qquad \psi_{I_{1обр}} = \psi_{U_{обр}} - \psi_{z_{обр}}. \tag{4}$$

Тогда сдвиг по фазе между составляющими тока статора прямой и обратной последовательности:

$$\Delta\psi_I = \psi_{I_{1пр}} - \psi_{I_{1обр}} = (\psi_{U_{пр}} - \psi_{U_{обр}}) - (\psi_{z_{пр}} - \psi_{z_{обр}}), \text{ или:}$$

$$\Delta\psi_I = \Delta\psi - \Delta\psi_z. \tag{5}$$

где $\Delta\psi_I$ — разность фаз токов прямой и обратной последовательности; $\Delta\psi$ — разность фаз напряжений прямой и обратной последовательности; $\Delta\psi_z$ — разность фаз сопротивлений для прямой и обратной последовательности

$$\Delta\psi_z = \psi_{z_{пр}} - \psi_{z_{обр}} = \mathrm{arctg}\,(\underline{Z}_{пр}) - \mathrm{arctg}\,(\underline{Z}_{обр}). \qquad (6)$$

Величина $\Delta\psi_z$ может быть найдена по паспортным данным асинхронного двигателя. Например, на основании приведенных в справочной литературе данных относительно параметров Т-образной схемы замещения [5, с.411] диапазон изменения углов для асинхронных машин серии 4А составляет: $\Delta\psi_z \approx -(25 \div 70^o)$. Из (5) следует, что при $\Delta\psi = 0^o$, принимаемом по умолчанию в соответствии с ГОСТ 13109 – 97, $\Delta\psi_I = \Delta\psi_z$. То есть при $\Delta\psi = 0^o$ симметричные составляющие токов статора всегда будут сдвинуты на некоторый угол $\Delta\psi_z \approx -(25^o \div 70^o)$. При изменении $\Delta\psi$ угол сдвига между симметричными составляющими токов статора изменяется. В случае, когда $\Delta\psi = \Delta\psi_z - 180^o$ сдвиг по фазе между составляющими токов прямой и обратной последовательности равен $\Delta\psi_I = 180^o$. Векторная диаграмма (ВД) при $\Delta\psi_I = 180^o$ для фазы А представлена на рис. 1. Результирующие значения токов, полученные из ВД (рис. 1):

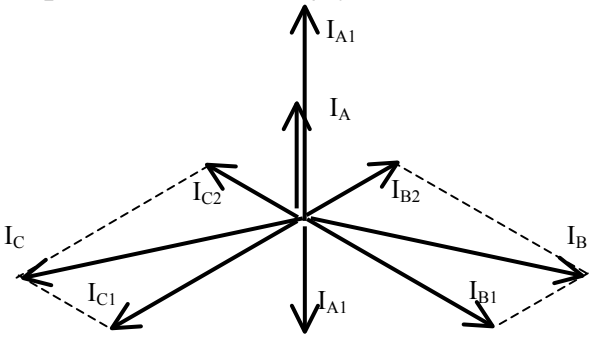


Рис. 1 Векторная диаграмма токов в обмотках статра при $\Delta\psi = \Delta\psi_{min}$

$$I_{1A} = I_{1пр} - I_{1обр}, \qquad (7)$$

$$I_{1B} = I_{1C} = \sqrt{\left(I_{1пр} + I_{1обр}\right)^2 - I_{1пр} I_{1обр}} \qquad (8)$$

При $\Delta\psi = \Delta\psi_z$ – сдвиг по фазе между симметричными составляющими токов в соответствии с уравнением (5) равен $\Delta\psi_I = 0^o$. Тогда векторная диаграмма примет вид (рис. 2). Полученные из ВД (рис.2) значения токов в фазах статора равны:

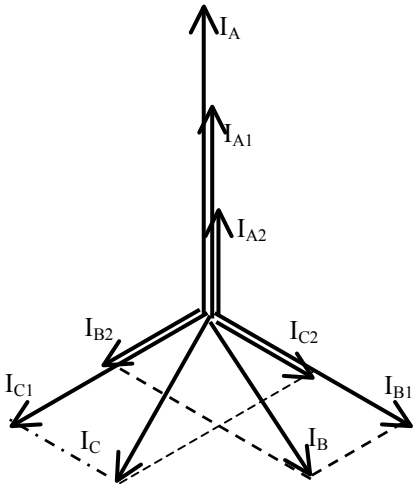


Рис. 2 Векторная диаграмма токов в обмотках статра при $\Delta\psi = \Delta\psi_{max}$

$$I_{1B} = I_{1C} = \sqrt{\left(I_{1пр} + I_{1обр}\right)^2 - 3 I_{1пр} I_{1обр}} \qquad (9)$$

$$I_{1A} = I_{1пр} + I_{1обр} \qquad (10)$$

Из выше изложенного следует, что при K_{2U} = const и изменении $\Delta\psi$ действующее значение тока наиболее нагруженой фазы статора может принимать значения от $I_{1ф} = \sqrt{\left(I_{1пр} + I_{1обр}\right)^2 - I_{1пр} I_{1обр}}$ до $I_{1ф} = I_{1пр} + I_{1обр}$. Таким образом, при заданном уровне несимметрии напряжения сети наибольшие токовые перегрузки в обмотках статора имеют место при совпадении направлений симметричных составляющих токов одной из фаз, что достигается при значениях $\Delta\psi = \Delta\psi_z$, которые для удобства обозначим $\Delta\psi_{max}$. Наименьшие токовые перегрузки в обмотках статора при заданном K_{2U} наблюдаются при противоположных направлениях $\underline{I}_{пр}$ и $\underline{I}_{обр}$, что достигается при значении $\Delta\psi = \Delta\psi_z - 180^o$, которые для удобства обозначим $\Delta\psi_{min}$.

Дальнейшее исследовании влияния несимметрии напряжения сети на эксплуатационные характеристики АД выполним для наиболее тяжелого случая, который имеет место при $\Delta\psi = \Delta\psi_{мах}$ [4, с.125].

Необходимо отметить, что значения $I_{пр}$ и $I_{обр}$ существенно зависят как от характеристик несимметрии источника питания (K_{2U}, $U_{пр}$, $\Delta\psi$), так и от параметров схемы замещения рассматриваемого АД. Определяющим фактором, влияющим на токи прямой последовательности, является параметр Т-образной схемы замещения – X_o^* [6, с.446], а на токи обратной последовательности – индуктивное сопротивление короткого замыкания $X_к^*$ [6, с.446]. Их относительные значения для общепромышленных АД мощностью до 400кВт могут изменяться в пределах $X_к^* = 0.15 \div 0.25$, $X_0^* = 2 \div 4$ [6].

В качестве примера в табл. 1 представлены результаты расчетов токов обратной последовательности статора ($I_{1обр}$) и ротора ($I_{2обр}$) при помощи метода симметричных составляющих с применением Т-образной схемы замещения при различных значениях $X_к^*$ и $U_{обр}$.

Таблица 1

Токи обратной последовательности статора ($I_{1обр}$) и ротора ($I_{2обр}$) при различных значениях $X_к^*$ и $U_{обр}$

$\frac{U_{обр}}{U_н}$	$x_к^* = 0.1$		$x_к^* = 0.15$		$x_к^* = 0.20$		$x_к^* = 0.25$		$x_к^* = 0.3$	
	$I_{1обр}^*$	$I_{2обр}^*$	$I_{1обр}^*$	$I_{2обр}^*$	$I_{1обр}^*$	$I_{2обр}^*$	$I_{1обр}^*$	$I_{2обр}^*$	$I_{1обр}^*$	$I_{2обр}^*$
0	0	0	0	0	0	0	0	0	0	0
0.02	0.15	0.39	0.12	0.36	0.10	0.32	0.08	0.28	0.02	0.14
0.04	0.29	0.56	0.24	0.52	0.19	0.45	0.15	0.40	0.04	0.20
0.1	0.75	0.89	0.60	0.82	0.48	0.70	0.37	0.52	0.10	0.33

Результаты расчета токов прямой последовательности для статора ($I_{1пр}$) и ротора ($I_{2пр}$) при различных относительных значениях X_o^* и различных значениях $U_{пр}$ представлены в табл. 2.

Таблица 2

Токи обратной последовательности статора ($I_{1обр}$) и ротора ($I_{2обр}$) при различных значениях X_0^* и $U_{пр}$

$\frac{U_{пр}}{U_н}$	$x_o^*=2$		$x_o^*=2.5$		$x_o^*=3$		$x_o^*=3.5$		$x_o^*=4$		$x_o^*=4.5$	
	$I_{1пр}^*$	$I_{2обр}^*$	$I_{1пр}^*$	$I_{2обр}^*$	$I_{1обр}^*$	$I_{2обр}^*$	$I_{1обр}^*$	$I_{2обр}^*$	$I_{1обр}^*$	$I_{2обр}^*$	$I_{1обр}^*$	$I_{2обр}^*$
1.0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
0.95	1.01	1.01	1.03	1.04	1.04	1.07	1.07	1.09	1.08	1.10	1.09	1.11
0.9	1.03	1.03	1.07	1.08	1.09	1.15	1.13	1.20	1.23	1.47	1.26	1.5

По данным табл. 1-2 были определены токи в обмотках статора при $\Delta\psi = \Delta\psi_{max}$ из выражений (7)-(8), а токи в обмотках ротора – из (2). Далее расчитывались потери при различных характеристиках несимметрии ($U_{пр}$ и $U_{обр}$, $\Delta\psi = \Delta\psi_{max}$), которые являются исходными данными для теплового расчета.

Расчёт теплового состояния АД осуществлялся с помощью системы дифференциальных уравнений, полученных на основании эквивалентной тепловой схемы (ЭТС) [3]. Для этого использовалась ЭТС, разработанная проф. Федоровым М.М., которая позволяет учесть неравномерный нагрев электродвигателя при различных характеристиках несимметрии источника питания [3]. Ее особенностью является тот факт, что в виде отдельных узлов выделены элементы конструкции каждой фазы статора. Это связано с тем, что мощность источников тепла в обмотках статора при несимметрии может иметь существенные отличия. Система алгебраических уравнений в матричной форме, полученная с помощью ЭТС, для расчета температур в установившемся состоянии, имеет вид: $\Lambda\theta = P$. Здесь $\theta = (\theta_1, \ldots, \theta_8)^T$ - вектор столбец превышения температур соответствующих узлов над температурой окружающей среды θ_0; $\Lambda = (\lambda_{ij})$ – матрица теплопроводностей ЭТС АД; $P = (P_1, \ldots, P_8)^T$ – вектор потерь мощности в узлах АД.

В режиме S1 несимметрия питающего напряжения сопровождается ростом температур всех узлов, однако, наибольшие тепловые нагрузки испытывает лобовая часть наиболее нагруженной фазы статора. Значения температур лобовой части наиболее нагруженной фазы статора θ_5) при различных характеристиках напряжения ($U_{пр}$, K_{2U}, $\Delta\psi = \Delta\psi_{max}$) и параметрах схемы замещения ($X_к^*$ и X_o^*) в продолжительном режиме S1 представлены в табл. 3. В качестве базовых были приняты температуры соответствующих узлов при номинальном входном напряжении.

Сравнение полученных результатов с экспериментальными значениями температур АД МТН 111-6 ($X_к^* = 0.2$, $X_о^* = 3.55$) и АИУМ225М4 ($X_к^* = 0.22$, $X_о^* = 4.03$) показывает, что расхождение между приведенными в табл. 3 и экспериментальными значениями не превышает 5-7%.

Таблица 3

Температура наиболее нагретого узла АД при различных характеристиках напряжения ($U_{пр}$, K_{2U}, $\Delta\psi = \Delta\psi_{max}$) и параметрах схемы замещения ($X_к^*$ и X_0^*) в продолжительном режиме S1

$U_{пр}$	$X_о^*$	$K_{2U} = 0\%$	$K_{2U} = 2\%$			$K_{2U} = 4\%$			$K_{2U} = 10\%$		
		$X_к^*$ =0.15÷0.27	$X_к^*$ =0.1	$X_к^*$ =0.15	$X_к^*$ =0.25	$X_к^*$ =0.1	$X_к^*$ =0.15	$X_к^*$ =0.25	$X_к^*$ =0.1	$X_к^*$ =0.15	$X_к^*$ =0.25
$1.0U_н$	2÷4	1.03	1.14	1.09	1.07	1.25	1.21	1.19	1.55	1.45	1.35
$0.95U_н$	4	1.06	1.20	1.13	1.10	1.31	1.28	1.24	1.84	1.75	1.56
	3	1.00	1.14	1.09	1.07	1.29	1.25	1.21	1.68	1.68	1.53
	2	0.97	1.11	1.06	1.04	1.25	1.22	1.19	1.78	1.65	1.52
$0.9U_н$	4	1.08	1.22	1.17	1.15	1.53	1.49	1.45	2.23	2.05	1.85
	3	0.98	1.12	1.07	1.05	1.31	1.28	1.27	1.96	1.81	1.68
	2	0.94	1.08	1.03	0.99	1.27	1.24	1.22	1.84	1.69	1.53

Из данных, представленных в табл. 3 следует, что при допустимых ГОСТ 13109 –97 значениях показателей качества электрической энергии ($K_{2U} \le 2\%$, $|\delta U| \le 5\%$) температурные перегрузки в следствие изменения $\Delta\psi$ могут достигать 10-20%. По результатам исследований выявлено (см. табл. 5), как должна изменяться нагрузка в режиме S1 при несимметрии напряжения, чтобы температура наиболее нагруженного узла АД не превышала допустимое значение.

Таблица 5

Нагрузка на валу АД в режиме S1 при различных характеристиках напряжения ($U_{пр}$, K_{2U}, $\Delta\psi = \Delta\psi_{max}$) и $\theta = \theta_{доп}$

$U_{пр}$	$X_о^*$	$K_{2U} = 0\%$	$K_{2U} = 2\%$		$K_{2U} = 4\%$		$K_{2U} = 10\%$	
		$X_к^*$=0.15÷0.25	$X_к^*$=0.15	$X_к^*$=0.25	$X_к^*$=0.15	$X_к^*$=0.25	$X_к^*$=0.15	$X_к^*$=0.25
1.0	2 ÷ 4	1.00	0.95	0.98	0.93	0.96	0.75	0.79
0.95	2	1.00	0.95	0.98	0.93	0.96	0.75	0.79
	4	0.98	0.94	0.96	0.92	0.95	0.74	0.78
0.9	2	1.00	0.95	0.98	0.93	0.96	0.75	0.79
	4	0.87	0.85	0.87	0.82	0.86	0.70	0.76

Выводы. В статье установлено, что при K_{2U} = const наибольшие тепловые перегрузки обмоток статора имеют место при совпадении по направлению симметричных составляющих токов одной из фаз, что достигается при $\Delta\psi = \Delta\psi_{max}$. Наименьшие токовые перегрузки в об-

мотках статора при заданном значении $К_{2U}$ наблюдаются при противоположных направлениях $\underline{I}_{пр}$ и $\underline{I}_{обр}$, что достигается при $\Delta\psi = \Delta\psi_{min}$.

Предложен алгоритм расчета токов и температур элементов конструкции обмоток АД при $\Delta\psi = \Delta\psi_{max}$. Применение алгоритма при проектировании АД позволит исключить тепловые перегрузки, связанные с изменением $\Delta\psi$ при эксплуатации. С помощью предложенного алгоритма была выполнена оценка теплового состояния АД различного типоразмера (мощностью до 400кВт) в продолжительном режиме при различных характеристиках несимметрии питающего напряжения ($U_{пр}$, $К_{2U}$, $\Delta\psi = \Delta\psi_{max}$), в результате чего установлено:

- при допустимых ГОСТ 13109 –97 значениях показателей качества электрической энергии δU и $К_{2U}$ ($К_{2U} \leq 2\%$, $|\delta U| \leq 5\%$) тепловые перегрузки, связанные с изменением $\Delta\psi$ могут достигать 10-20%, в результате сокращение срока безаварийной эксплуатации может снизиться до 20-80%;
- определены необходимые величины снижения нагрузок, позволяющие сохранить ресурс изоляционных материалов.

Библиография

1. Копылов И.П., Клоков Б.К., Морозкин В.П., Токарев Б.Ф. Проектирование электрических машин. -3-е изд. испр. и доп. - М.: Высшая школа, 2002. - 757 с.
2. ГОСТ 13109 – 97. Межгосударственный стандарт. Электрическая энергия. Совместимость технических средств электромагнитная. Нормы качества электрической энергии в системах электроснабжения общего назначения. Введен в Украине с 01.01.2000.
3. Жежеленко И.В. Показатели качества электроэнергии и их контроль на промышленных предприятиях.– М: Энергоатомиздат. - 1986. – 174с.
4. Федоров М.М., Пинчук О.Г. Особенности теплового состояния асинхронных двигателей при несимметричном питающем напряжении. Вісник Кременчуцького державного політехнічного університету: – Кременчуг. – 2004. – Вип. 2(25). - С. 122-125.
5. Иванов-Смоленский А.В. Электрические машины. – М: Энергия. – 1980. – 928с.
6. Зевеке Г.В., Ионкин П.А., Нетушил А.В., Страхов С.В. Основы теории цепей: Учебник для вузов.–М.: Энергоатомиздат.–1989. – 528с.
7. Толочко О.І., Чекавський Г.С., Мірошник Д.М. Векторні моделі асинхронного двигуна у середовище пакета Matlab./ праці Кременчуцького державного політехнічного університету. – 2003, Том 1, №2 (19) – с.199 – 202.

Дата поступления в редколлегию 16.06.2007.

УДК 681.3

UML-ПОДХОД К РАЗРАБОТКЕ ДИСКРЕТНЫХ СИМУЛЯТОРОВ (НА ПРИМЕРЕ ОБЪЕКТОВ УГОЛЬНОЙ ПРОМЫШЛЕННОСТИ)

Чепцов А.А.
Донецкий национальный технический университет

Рассмотрена технология модельной поддержки разработок и эксплуатации сложных динамических систем на примере объектов угольной промышленности. Предложен алгоритм разработки симуляторов дискретных моделей. Предложена унифицированная UML-технология разработки дискретных симуляторов.

1. Введение

Компьютерное моделирование производственных процессов, технологических объектов и систем управления ими приобрело во всем мире новое качество: оставаясь мощным средством научных исследований, достоверные модели, реализованные в современной компьютерной технике как программные продукты (симуляторы), используются фирмами для решения производственных задач прогнозирования и планирования, оперативного управления с оптимизацией затрат и прибыли, предупреждения аварийных ситуаций и быстрейшего перевода объектов в безопасные режимы работы, модельной проверки правильности проектных решений по модернизации технологических процессов и производств, обучения персонала на тренажерах-симуляторах и др. Нами предлагается подход к информационно-программной поддержке разработки дискретных симуляторов для объектов угольной промышленности, основанный на унифицированном языке моделирования UML [1]. Полученные для объектов угольной промышленности технологии создания моделирующих программных компонентов в составе технологически-ориентированных распределённых моделирующих сред [2] могут быть перенесены также на другие предметные области.

2. Алгоритм разработки симуляторов для объектов предметной области

Основой достоверных моделей и симуляторов является физически правильное математическое описание моделируемых процессов, объектов и систем. ДонНТУ в содружестве с МакНИИ

разработаны модели аэрогазодинамических процессов в схемах проветривания выемочных участков (СПВУ) и в шахтных вентиляционных сетях (ШВС) [3], на базе которых построены системы диспетчерского контроля и управления проветриванием и безопасностью АТМОС и КАГИ, что позволяет реализовать в угольной отрасли Украины планомерное использование компьютерных моделей. При этом возникает задача внедрения современных подходов к информационно-программной поддержке разработки дискретных симуляторов.

В соответствии с требованиями предметной области [4], для разработки симуляторов объектов угольной промышленности предлагается унифицированный алгоритм, в соответствии с которым для объектов предметной области реализуется универсальный процесс разработки «модель (*model*) – модель в форме, удобной для использования численного метода (*simulation model*) – дискретная форма модели, пригодная для численного решения (*discrete simulation model*) – UML-модель (*UML-Model*) – симулятор», распространяемый на заданную предметную область [5]. Реализация алгоритма для предметной области «ШВС и безопасность труда» представлена на рис.1.

Модель (*model*) – это представление топологии и математическое описание динамических процессов, протекающих в объектах исследований. Алгоритмом предусматриваются модели для динамических систем с сосредоточенными (ДССП-модели) и распределёнными (ДСРП-модели) параметрами [6]. В общем случае ДССП-модель содержит нелинейные и линейные дифференциальные уравнения различных порядков, алгебраические уравнения. ДСРП-модель представляется дифференциальными уравнениями в частных производных. Получена модель СПВУ шахтной вентиляционной сети [5] (предметная область «Шахтные вентиляционные системы (ШВС) и безопасность труда»), состоящая из уравнений откаточного штрека, аппроксимированного параллельными воздушными потоками выработанного пространства лавы, вентиляционного штрека и граничных условий. Модель приводится к системе уравнений в виде *simulation model*. Для ДСРП при этом существенной является аппроксимация по пространственным координатам. Модели сетевых динамических объектов (например ШВС) для заданной предметной области состоят из m уравнений для ветвей и n уравнений для узлов сетевого объекта [7]. Для ШВС при аппроксимации ветвей по методу прямых наряду с базовой топологией в форме графа строится вторичная топология, содержащая последовательно связанные

элементы ветвей и фрагменты типа схем проветривания участка [3], что требует разработку топологического анализатора и генераторов

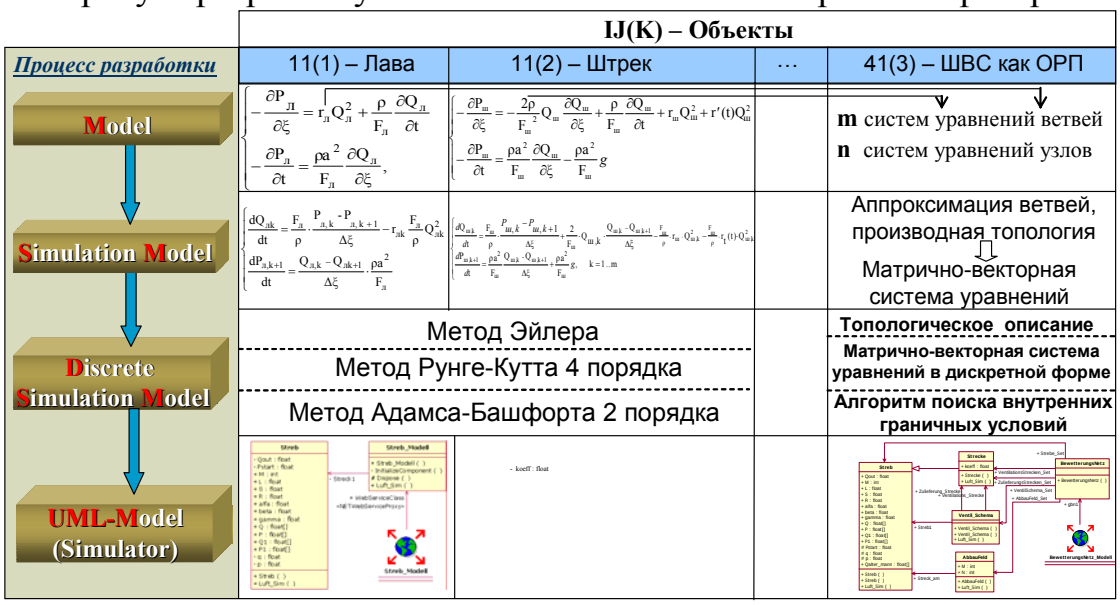


Рис.1. Алгоритм разработки симуляторов для объектов предметной области «Шахтные вентиляционные системы и безопасность труда».

уравнений ШВС как объектов с распределенными параметрами. Путем выбора одного из численных методов решения дифференциальных уравнений *simulation model* трансформируется в дискретную форму.

3. UML-поддержка разработки моделирующих программных компонентов

Разработанные дискретные модели подлежат программной реализации в качестве моделирующих программных компонентов. Для разработки моделирующих программных компонентов и интеграции проектных решений предложен подход, представляющий собой *RUP (Rational Unified Process)*-объектно-ориентированную стратегию, тесно интегрированную с унифицированным языком моделирования UML [1]. Применение UML-технологий позволило охватить такие аспекты разработки и практического использования симуляторов, как представление объектов исследований (описание топологии и параметров), компонентная реализация симуляторов (администрирование компонентной организации моделирующего программного обеспечения), использование симуляторов (документирование функциональности симуляторов, взаимодействие с другими компонентами), системная организация (поддержка в актуальном состоянии структуры и состава ресурсов), реализация

алгоритмов моделирования в симуляторах. Разработанная технология создания проблемно-ориентированных моделирующих компонентов, включает фазы постановки задачи, системного анализа, проектирование и реализации [8]. При этом актуальной задачей являлась адаптация фаз разработки к особенностям задач моделирования и специфике заданной предметной области, что предлагается решать путем разработки специализированного UML-инструментария полнофункционального и дружественного к пользователям отображения взаимосвязей между формальным описанием сложных динамических систем и UML-базированным представлением проблемно-ориентированных компонентов (рис.2).

Рис.2. Формальное и UML-базирование описание объектов моделирования.

В рамках предложенной стратегии разработаны *UML*-диаграммы предметной области, симуляторов данных. Модифицированы UML-диаграммы деятельности (Deployment diagram), времени (Timing diagram), развёртывания (Deployment diagram), последовательности (Sequence diagram), классов (Classes diagram) [9].

4. Заключение

Разработанные технологии использованы при построении моделирующего сервисного центра для угольной промышленности [4,7], в рамках которого программно реализованы модели объектов технологического направления «Техника безопасности и проветривание» угольной промышленности: лавы, откаточного и вентиляционного штрека, воздухоподающего и воздухоотводящего ствола, схемы проветривания выемочного участка, шахтной вентиляционной сети как объектов с распределёнными параметрами.

Дальнейшие разработки концентрируются на реализации дискретных симуляторов для предметной области «Шахтные вентиляционные системы и безопасность труда горняков», а также апробации на шахте „Южно-Донбасская №3" (Донецкая область, Украина).

Литература

1. Фаулер М., Скотт К. UML в кратком изложении. Применение стандартного языка моделирования.: Пер. с англ. – М.: Мир, 1999. – 191с.
2. K. Bondareva, V. Svjatnyj. Verfahrenstechnisch orientierte parallele Simulationsumgebung. Наукові праці ДонНТУ, Серія «Проблеми моделювання та автоматизації проектування динамічних систем», вип. 5(116), 2006, с. 28-35.
3. Абрамов Ф.А., Фельдман Л.П., Святный В.А. Моделирование динамических процессов рудничной аэрологии. – К.: Наукова думка, 1981. – 284с.
4. Чепцов О.О., Бєляєв О.Н. Розробка моделюючого та інформаційно-сервісного центру (МСЦ) для вугільних шахт – Збірник праць науково-практичної конференції „Донбасс-2020", 30-31 травня 2006р., Донецьк.
5. Чепцов А.А. Алгоритм разработки объектов иерархии моделей заданной предметной области в моделирующем сервисном центре. Наукові праці ДонДТУ, Серія:"Інформатика, кібернетика та обчислювальна техніка", вип. 8(120), 2007 – с. 309-316.
6. Святний В.А.: Паралельне моделювання складних динамічних систем / Сборник трудов конференции «Моделирование-2006», Институт проблем моделирования в энергетике им. Г.Е.Пухова НАН Украины, Киев 2006. – с.83-90.
7. Cheptsov, O. "The system organization and basic algorithms of the Simulation- and Servicecentre for the coal industry", in: IEEE Proceedings International Conference "Modern Problems of Radio Engineering, Telecommunications and Computer Science TCSET'2006", pp. 205-207.
8. Якобсон А., Буч Г., Рембо Дж. Унифицированный процесс разработки программного обеспечения. СПб.: Питер, 2002.
9. Чепцов О.О. Системна організація та розробка компонентів проблемно-орієнтованих моделюючих та інформаційно-сервісних центрів – Моделирование-2006, Институт проблем моделирования в энергетике им Г.Е.Пухова НАН Украины, Киев 2006. – с.437-442.

Дата поступления в редакцию 10.06.2007