

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

3(199)
2013

ТЕОРЕТИЧЕСКИЙ И ПРИКЛАДНОЙ НАУЧНО-ТЕХНИЧЕСКИЙ ЖУРНАЛ

Издается с ноября 1995 г.

УЧРЕДИТЕЛЬ
Издательство "Новые технологии"

СОДЕРЖАНИЕ

СИСТЕМЫ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ

- Стенников В. А., Барахтенко Е. А., Соколов Д. В. Применение онтологий при реализации программного комплекса для решения задач оптимального проектирования теплоснабжающих систем 2
- Черемисинова Л. Д., Кириенко Н. А. Синтез многоуровневых логических схем с учетом энергопотребления 8

МОДЕЛИРОВАНИЕ И ОПТИМИЗАЦИЯ

- Мамедов Р. К., Владимирский Э. И., Мустафаева С. Р. Аномальная диффузия и ее влияние на хаотические системы. 15
- Четырбоцкий А. Н. Приложение прямого метода решения системы линейных уравнений для численного моделирования сферического течения Куэтта несжимаемой жидкости в среде MATLAB 20
- Елесина С. И., Логинов А. А., Никифоров М. Б. Методика оценки качества кри- териальной функции совмещения изображений. 24

ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ И СЕТИ

- Речистов Г. С. Использование полноплатформенного имитационного моделиро- вания суперкомпьютерной системы для определения производительности счетных приложений. 29
- Саак А. Э. Полиномиальные алгоритмы диспетчеризации массивов заявок гипер- болического типа. 33

КОМПЬЮТЕРНАЯ ГРАФИКА И ОБРАБОТКА ИЗОБРАЖЕНИЙ

- Бобков В. А., Роньшин Ю. И., Машенцев В. Ю., Кудряшов А. П. Навигация ав- тономного подводного аппарата по видеопотоку 36
- Бимаков В. А., Касимов Т. Р. Использование воксельных вычислителей для ско- ростного преобразования 2D- и 3D-изображений высокого разрешения. 42

БАЗЫ ДАННЫХ

- Барский А. Б., Нгуен Ван Лой. Оценка среднего времени выполнения запроса к многосерверной базе данных с циркулирующими сегментами 48
- Власов С. В. Сущностно-ориентированный подход к разработке информаци- онного хранилища (в задачах транспорта газа). 51

Журнал в журнале НЕЙРОСЕТЕВЫЕ ТЕХНОЛОГИИ

- Артеменков С. Л. Иерархия процессов опознания в "перцептроне" Миракяна . . . 56
- Пекунов В. В. О классификации лиц методом голосования с нейросетевым ар- битром. Распараллеливание вычислений на многоядерных видеокартах 61
- Дорогов А. Ю., Абатуров В. С., Раков И. В. РММЛ-модели быстрых нейронных сетей. 65
- Contents 71
- Приложение. Бетелин В. Б., Галатенко В. А., Костюхин К. А. Основные понятия контролируемого выполнения сложных систем

Главный редактор
НОРЕНКОВ И. П.

Зам. гл. редактора
ФИЛИМОНОВ Н. Б.

Редакционная
коллегия:

- АВДОШИН С. М.
АНТОНОВ Б. И.
БАРСКИЙ А. Б.
БОЖКО А. Н.
ВАСЕНИН В. А.
ГАЛУШКИН А. И.
ГЛОРИОЗОВ Е. Л.
ДОМРАЧЕВ В. Г.
ЗАГИДУЛЛИН Р. Ш.
ЗАРУБИН В. С.
ИВАННИКОВ А. Д.
ИСАЕНКО Р. О.
КОЛИН К. К.
КУЛАГИН В. П.
КУРЕЙЧИК В. М.
ЛЬВОВИЧ Я. Е.
МАЛЬЦЕВ П. П.
МЕДВЕДЕВ Н. В.
МИХАЙЛОВ Б. М.
НЕЧАЕВ В. В.
ПАВЛОВ В. В.
ПУЗАНКОВ Д. В.
РЯБОВ Г. Г.
СОКОЛОВ Б. В.
СТЕМПКОВСКИЙ А. Л.
УСКОВ В. Л.
ФОМИЧЕВ В. А.
ЧЕРМОШЕНЦЕВ С. Ф.
ШИЛОВ В. В.
- Редакция:
БЕЗМЕНОВА М. Ю.
ГРИГОРИН-РЯБОВА Е. В.
ЛЫСЕНКО А. В.
ЧУГУНОВА А. В.

Информация о журнале доступна по сети Internet по адресу <http://novtex.ru/IT>.
Журнал включен в систему Российского индекса научного цитирования.
Журнал входит в Перечень научных журналов, в которых по рекомендации ВАК РФ должны быть опубликованы научные результаты диссертаций на соискание ученой степени доктора и кандидата наук.

УДК 621.311.001.57

В. А. Стенников,

д-р техн. наук, зам. директора по науке,

Е. А. Барактенко, канд. техн. наук, науч. сотр.,

Д. В. Соколов, вед. инж.,

Институт систем энергетики им. Л. А. Мелентьева

СО РАН (ИСЭМ СО РАН), г. Иркутск,

e-mail: barakhtenko@isem.sei.irk.ru

Применение онтологий при реализации программного комплекса для решения задач оптимального проектирования теплоснабжающих систем

Для решения задачи оптимального проектирования систем теплоснабжения предложена комплексная методика, основанная на принципах декомпозиции и дальнейшей увязки получаемых результатов в едином вычислительном процессе. Применявшиеся ранее подходы к построению программного обеспечения не обеспечивали полноценную реализацию этой методики в виде единого программного комплекса. Для преодоления возникающих в связи с этим трудностей был разработан методический подход к построению программного обеспечения на принципиально новой основе. Данный подход основан на применении концепции модельно-управляемой разработки, где для представления моделей используется онтологическая система. Принципиальные положения этого подхода использованы при разработке ПК СОСНА-М, предназначенного для решения задачи оптимального проектирования теплоснабжающих систем.

Ключевые слова: онтология, онтологическая система, программный комплекс, теплоснабжающая система, оптимальное проектирование, сценарий, метапрограммирование

Введение

Теплоснабжающие системы (ТСС) современных городов и промышленных центров представляют собой сложные по структуре и параметрам системы, снабжающие теплом множество потребителей, имеющие в своем составе разнообразное оборудование и работающие с несколькими источниками различного типа. Для таких систем все актуальнее становится своевременное решение вопросов оптимального проектирования, что обусловлено старением теплопроводов, оборудования и ежегодным

ростом присоединенной нагрузки потребителей. Необходимо обеспечить своевременный анализ и исследование состояния, свойств таких ТСС, которые определяются постоянным развитием систем, конъюнктурой рыночных условий, составом элементов, появлением новых энергоэффективных технологий и оборудования.

Комплексная методика решения задачи проектирования ТСС [2], предложенная в ИСЭМ СО РАН в рамках теории гидравлических цепей (ТГЦ) [1], основана на применении метода декомпозиции общей задачи на ряд подзадач: схемно-структурной оптимизации, схемно-параметрической оптимизации, расчета показателей надежности, расчета нормальных и аварийных режимов, расчета параметров источников тепла и расчета показателей экономической эффективности.

Особенность разработанной в ИСЭМ методики решения задачи оптимального проектирования ТСС состоит в том, что она предполагает формирование своего алгоритма для различного состава подзадач, индивидуального для множества рассматриваемых ТСС, с учетом их конкретных особенностей. Как правило, это сложный итерационный вычислительный процесс, в ходе которого перечисленные выше подзадачи для различных ТСС можно решать в разной последовательности и различными методами в зависимости от поставленной цели. Один из возможных вариантов алгоритма решения задачи оптимального проектирования ТСС приведен на рис. 1.

Недостатки существующего программного обеспечения

Ранее программное обеспечение (ПО) разрабатывали в соответствии с модульной парадигмой программирования. Роль модулей, как правило, играли библиотеки подпрограмм, реализующие некоторую функциональность и предоставляющие интерфейс к ней. Используемая парадигма построения ПО имеет целый набор ограничений, среди них можно выделить следующие:

- невозможность организации гибкой схемы управления вычислительным процессом, необходимой при реализации методики решения задачи оптимального проектирования ТСС;
- отсутствие средств, позволяющих инженеру-энергетику разрабатывать алгоритм решения задачи оптимального проектирования ТСС, организуя и изменяя единый итерационный вычислительный процесс;

Концепция модельно-управляемой разработки

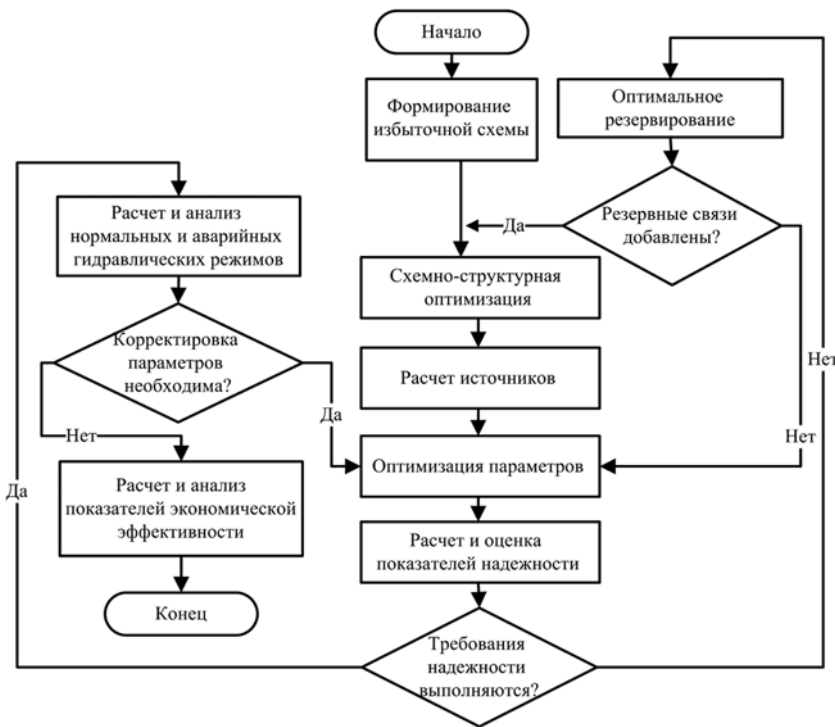


Рис. 1. Вариант алгоритма решения задачи оптимального проектирования ТСС

- схема взаимодействия между математическими модулями скрыта в программном коде управляющего модуля программной системы, что делает невозможным понимание и развитие алгоритма решения задачи для инженера-энергетика.

Математические модули не являются полноценными программными компонентами, которые можно многократно использовать при построении различных программных систем, и не обеспечивают реализации гибкой схемы управления вычислительным процессом. Имеются серьезные трудности, связанные с интеграцией отдельных вычислительных модулей, написанных на разных языках программирования, в единый программный комплекс. Отсутствие возможности для исследователя создания моделей элементов и интеграции их в ПО при проведении научных или инженерных расчетов.

Имеется ряд недостатков традиционной технологии организации доступа к базам данных, затрудняющих интеграцию ПО с базами данных различных разработчиков. Модули доступа к базам данных, как правило, разработаны таким образом, что могут взаимодействовать с СУБД только одного типа и соответствующей структурой, что затрудняет переход к базам данных других разработчиков и БД, имеющим другую структуру.

Рассмотренные недостатки традиционной архитектуры ПО привели к необходимости разработки нового методического подхода к построению сложных программных систем для решения задач оптимального проектирования ТСС.

В основе нового методического подхода лежит концепция разработки ПО, управляемой моделями (Model-Driven Development, MDD) [3]. Идея MDD состоит в том, что ПО автоматически строится на основе моделей для решения конкретной задачи. Модели, описывающие объекты решаемой задачи и связи между ними, формализуются с помощью специальных языков описания предметных областей. Формирование схемы вычислительного процесса происходит на основе этих моделей. В качестве средства представления модели использованы онтологии, которые предложено организовать в онтологическую систему.

Особенность предложенного методического подхода состоит в том, что он позволяет:

- осуществить разделение моделей элементов ТСС и методов, применяемых для решения задач;
- динамически формировать ПО при решении конкретной задачи, в том числе создавать часть программных компонентов по моделям, сформированным на основе онтологий;
- автоматически организовывать вычислительный процесс на основе схемы, составленной инженером-энергетиком в терминах его предметной области.

Разработанный методический подход включает:

- архитектуру универсальной инструментальной платформы для построения ПО, предназначенного для решения задач оптимального проектирования ТСС;
- методы разработки и применения программных компонентов;
- методику построения и применения онтологий;
- технологию автоматической настройки ПО на расчет различных ТСС, с разным составом оборудования;
- базовую архитектуру ПО для решения задач оптимального проектирования ТСС;
- технологию автоматической организации вычислительного процесса.

Структура онтологической системы

Развитие рынка оборудования, с одной стороны, приводит к повышению эффективности ТСС, с другой стороны, при решении задачи оптимального проектирования ТСС требуется учитывать широкий спектр оборудования с различными параметрами. ПК должен предоставлять возможность работы со всем множеством оборудования для ре-

шения всего комплекса задач оптимального проектирования ТСС. В свою очередь, для автоматизации вычислительного процесса, во-первых, необходимо иметь формализованное описание решаемых задач и, во-вторых, требуется осуществить формализованное представление связей между решаемыми задачами для автоматического составления иерархии подзадач. Для этих целей должно быть выполнено унифицированное описание множества объектов ТСС и решаемых задач таким образом, чтобы объекты предметной области и отношения между ними были представлены в виде формализованных знаний, которые могли бы быть использованы программными системами во время их работы. Решить данную задачу позволяют *онтологии* — базы знаний специального вида, формализующие знания предметных областей. В инженерии знаний под онтологией понимается описание некоторой предметной области, которое используется для формального и декларативного определения ее концептуализации [4, 5].

В рамках разработанного методического подхода предложена методика построения онтологической системы, которая состоит из следующих компонентов (рис. 2): метаонтологии, онтологии инженерных знаний, онтологии программного обеспечения и машины вывода. Онтологическую систему применяют для формализованного представления моделей объектов предметной области при реализации концепции MDD.

Метаонтология — это онтология, которая специфицирует множество онтологий. Она содержит базовые понятия предметной области и устанавливает отношения между ними, которые используются для построения онтологии инженерных знаний и онтологии программного обеспечения.

Онтология инженерных знаний состоит из онтологии теплоснабжающей системы, онтологии задач, ее строит с применением специального редактора онтологий инженер-энергетик, который является специалистом в данной области, при этом участия программиста не требуется.

Онтология теплоснабжающей системы (онтология ТСС) содержит описание свойств подсистем ТСС, типы и параметры оборудования, иерархию

структуры элементов сети, их свойства, связи между элементами, математическое описание (например, формулы расчета гидравлических сопротивлений, падений давлений и др.). Данная онтология предназначена для формирования моделей элементов, которые в процессе осуществления вычислений интегрируются с вычислительными алгоритмами, содержащимися в программных компонентах. Применение полученных моделей позволяет настраивать ПО на особенности оборудования, рассматриваемого при проектировании конкретной ТСС.

Онтология задач содержит формализованное описание задач, решаемых при проектировании ТСС, входящих в них подзадач и последовательность решения последних, параметров, являющихся исходными данными, и параметров, получаемых в результате решения задачи. Она ориентирована на организацию пошагового решения поставленных задач (вызов программных модулей и подпрограмм).

Онтология программного обеспечения (онтология ПО) состоит из онтологии программных компонентов и онтологии баз данных.

Онтология программных компонентов содержит описание программных компонентов и их свойств, входные и выходные параметры, форматы данных, технологии и интерфейсы доступа к компонентам системы. Данная онтология используется для организации доступа к программным компонентам. Управляющий компонент, предназначенный для организации вычислительного процесса, выполняет обход графа решения задачи и последовательно вызывает компоненты, соответствующие шагам решаемой задачи. Вызов компонентов происходит на основе информации, содержащейся в онтологии ПО.

Онтология баз данных (онтология БД) содержит описание структуры используемых баз данных (таблицы и связи между ними, представления, хранимые процедуры и т. д.). Онтология БД используется для организации взаимодействия ПО с различными БД при автоматической генерации конфигурации доступа к СУБД. Данный подход позволяет настраивать ПО на использование БД разных разработчиков, с произвольной структурой и предназначенных для работы под управлением различных СУБД (MS Access, MS SQL Server, Firebird и т. д.).

Машина логического вывода — это программа, моделирующая механизм рассуждений и оперирующая знаниями и данными в целях получения новых данных из знаний и других данных, имеющихся в рабочей памяти [4]. Обычно машина логического вывода использует программно реализованный механизм дедуктивного логического вывода или механизм поиска решения в сети



Рис. 2. Структура онтологической системы

фреймов или семантической сети. Так, в онтологии задач формализовано хранятся знания о задачах и о составе входящих в нее подзадач. Машина вывода на основе знаний и правил, которые их связывают, строит общее решение для конкретной ТСС.

Одним из важнейших преимуществ, получаемых за счет применения онтологий, является формализация знаний предметных областей, которые можно многократно использовать при решении различных задач. Такие знания формализуют при участии инженеров-энергетиков, являющихся специалистами в предметной области, и они могут быть использованы как самостоятельный информационный ресурс в качестве справочной информации.

В качестве средства формализации онтологической системы в рамках предложенного методического подхода выбран язык XML (англ. extensible markup language — расширяемый язык разметки). Особенностью разрабатываемого формализованного описания является использование собственного языка вместо стандартных средств и языков онтологий. Причиной разработки собственного языка служит то, что в данном случае онтология имеет прикладной характер и предназначена для использования в конкретном ПК. В связи с этим целесообразно использовать свой предметно-ориентированный язык, наиболее лаконичный и близкий к языку энергетиков и программистов. Для формализованного описания графических моделей объектов ТСС используют SVG (англ. scalable vector graphics — масштабируемая векторная графика), математических выражений — MathML (англ. mathematical markup language — язык математической разметки).

Архитектура программного обеспечения

Авторами разработана архитектура ПО, учитывающая особенности методики решения задач оптимального проектирования ТСС [3, 6, 7]. Ее особенностью является широкое применение технологий метапрограммирования [8]. Архитектура представляется в виде трех логических уровней (рис. 3), описание которых рассматривается ниже.

Супервизор. Он организует процесс решения задачи оптимального проектирования ТСС, который состоит из следующих шагов.

1. Планирование вычислительного процесса, в ходе которого строится граф, узлы которого соответствуют шагам решения задачи, дуги — связям между шагами. Этот граф строится автоматически на основе знаний, содержащихся в онтологии задач.

2. Создание алгоритма решения задачи на основе полученного графа. Алгоритм формализовано представляется в виде сценария, который является набором операторов языка программирования, причем эти операторы интерпретируются без проведения компиляции программного кода. В качестве языков сценариев выбраны JavaScript и Python.

3. Выполнение подготовленного сценария решения задачи осуществляется в ходе вычислительного процесса путем последовательного вызова вычислительных программных компонентов. Для интерпретации полученного сценария используется технология *Java Scripting* [9], которая предназначена для взаимодействия программ, написанных на языке *Java*, с программами, реализованными с использованием языков сценариев.

Динамическое подключение новых вычислительных компонентов к работающей программе позволяет под конкретную задачу собирать программную систему из готовых компонентов во время ее работы. Для этого используется подход, основанный на применении онтологии ПО и механизмов рефлексии языка программирования *Java*. Средства рефлексии языка *Java* обеспечивают получение информации о компонентах, загрузку их в память и манипуляцию ими во время работы программы.

Вычислительная подсистема. Генерация компонентов-моделей позволяет отделить реализацию вычислительных алгоритмов от программных реализаций математических моделей элементов ТСС [7]. Язык программирования *Java* имеет набор необходимых средств для генерации исходного *Java*-кода

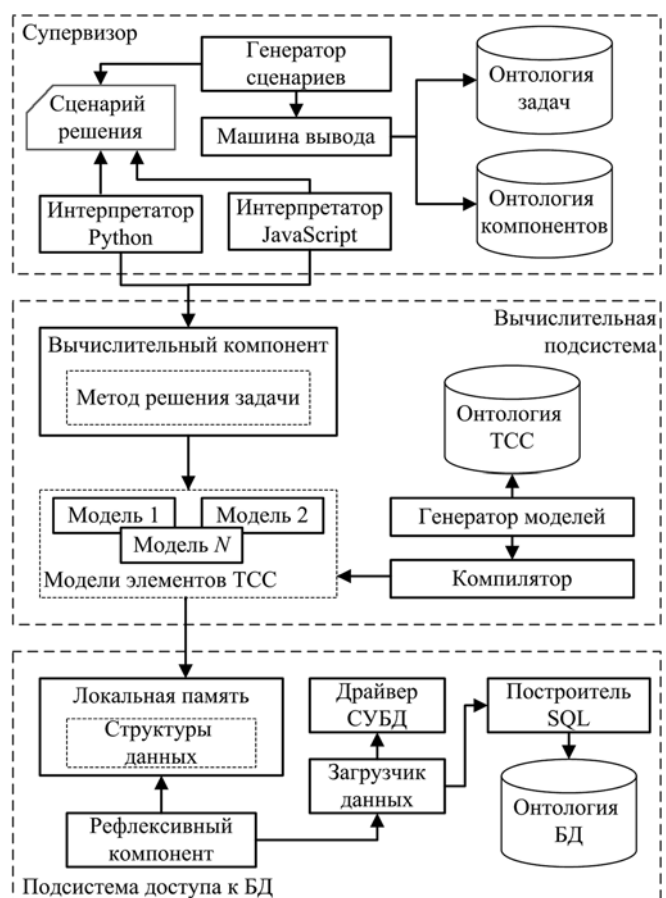


Рис. 3. Разработанная архитектура программного обеспечения

как результата работы самой программы, его компиляции в программные компоненты и подключения их к программной системе во время ее работы.

Подсистема доступа к БД. В целях организации гибкого доступа к БД различных разработчиков, реализованных под разными СУБД, происходит автоматическая генерация SQL-кода. В данной подсистеме используется механизм рефлексии, содержащийся в разработанном рефлексивном компоненте, для динамического заполнения структур данных, используемых вычислительными компонентами. В результате применения данных технологий компоненты доступа к БД приобретают важные свойства, такие как гибкость и универсальность использования.

ПК СОСНА-М и его применение

Разработанный методический подход использован при реализации ПК СОСНА-М, который позволяет находить оптимальные параметры системы как для разветвленных, так и кольцевых схем ТСС с множеством источников, сложным рельефом местности, произвольным числом узлов, участков и контуров [6]. С его помощью в проектной практике могут быть подготовлены рекомендации по реконструкции и развитию реальных ТСС, включающие предложения по выбору диаметров новых и реконструируемых участков тепловой сети, видам реконструкции существующих участков, местам установки насосных станций, имеющимся напорам теплоносителя на насосных станциях и источниках. Кроме того, ПК осуществляет проверку работоспособности системы и рассчитывает финансово-экономические показатели предлагаемого решения.

Основной частью данного ПК является оптимизатор, в котором реализован метод МКО. Схема работы оптимизатора показана на рис. 4. ПК предоставляет пользователю возможность задать конфигурацию процесса оптимизации, т. е. сформировать желаемый набор используемого оборудования, состав мероприятий по реконструкции существую-

щих участков сети, предложения по размещению насосных станций, ограничения на производительность теплоисточников и действующие напоры на них, ограничения по давлениям теплоносителя у потребителей и т. д. На основе заданной пользователем конфигурации автоматически формируется набор моделей для используемого в исследуемой ТСС оборудования, которые извлекаются из онтологии ТСС. Основным преимуществом предложенной технологии организации вычислительного процесса является возможность осуществления оптимизации параметров ТСС с дифференциацией допустимого набора трубопроводов, оборудования и видов реконструкции элементов конкретной теплоснабжающей системы с учетом ее особенностей и условий развития.

ПК СОСНА-М представляет собой универсальный вычислительный инструмент, используется для исследовательских целей и применяется для расчета реальных ТСС городов и населенных пунктов. На рис. 5 приведен фрагмент ТСС одного из населенных пунктов Иркутской области. В связи с ростом тепловых нагрузок, старением теплопроводов и оборудования возникла необходимость решения вопросов оптимального проектирования системы. С помощью разработанного ПК была проведена оптимизация параметров теплопроводов и насосных станций ТСС. На основе выполненных расчетов были определены:

- узкие места в системе с недостаточной пропускной способностью и значительными потерями давления теплоносителя;
- типы реконструкции перегруженных участков (перекладка теплопроводов существующих участков, докладка нового теплопровода);
- оптимальные давления теплоносителя в подающей и обратной магистралях (в источниках, узлах соединения участков и у потребителей);
- места расположения новых насосных станций и их параметры.

Основные решения по развитию ТСС показаны на рис. 5. Для удовлетворения требований по работоспособности системы необходимо сооружение двух насосных станций, осуществление перекладки на больший диаметр четырех участков сети и увеличение действующего напора на котельных. Новые насосные станции на рис. 5 отмечены соответствующими условными обозначениями, участки, требующие реконструкции в связи с недостаточной их пропускной способностью, выделены жирными линиями, определены типы их реконструкции и новые диаметры трубопроводов. Предложенные рекомендации по реконструкции

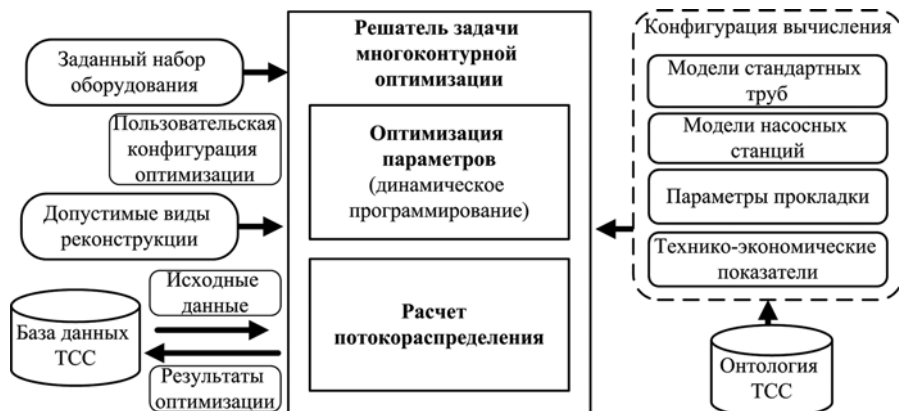


Рис. 4. Схема работы оптимизатора при решении задачи многоконтурной оптимизации

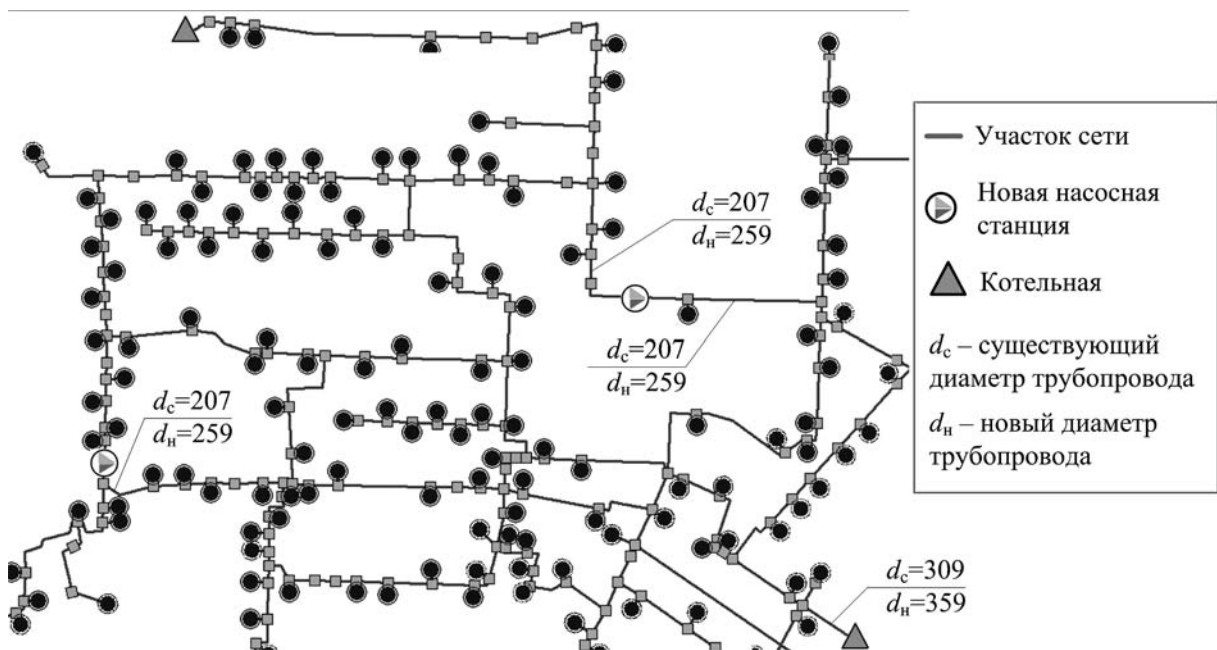


Рис. 5. Фрагмент схемы теплоснабжающей системы одного из населенных пунктов Иркутской области

ТСС обеспечивают ее адаптацию к возросшим перспективным тепловым нагрузкам и будут способствовать реализации оптимальных режимов ее функционирования.

Заключение

Предложен методический подход к построению ПО, который позволяет реализовать разработанную в ИСЭМ СО РАН комплексную методику решения задачи оптимального проектирования ТСС. Предложенный методический подход основан на применении MDD, где для формализованного описания моделей объектов предметной области используется онтологическая система.

Применение онтологий значительно расширило условия и увеличило возможности использования ПК, а также позволило: отделить реализации алгоритмов в виде программных компонентов от моделей элементов ТСС; гибко управлять процессом вычислений, формируя вычислительные итерационные последовательности из готовых программных компонентов; настраивать программное обеспечение для решения задач оптимального проектирования ТСС.

Предложенный методический подход имеет все условия для реализации открытых программных систем из готовых программных компонентов и обеспечивает многократное использование программных реализаций алгоритмов решения задач оптимального проектирования ТСС.

Разработанный методический подход использован при построении ПК СОСНА-М, который по-

зволяет подготавливать рекомендации по проектированию реальных ТСС с определением диаметров новых участков тепловой сети, видов реконструкции существующих участков, мест установки насосных станций, располагаемых напоров теплоносителя на насосных станциях и источниках, осуществлять проверку работоспособности системы и получать экономическую оценку предлагаемого решения.

Список литературы

1. Математическое моделирование и оптимизация систем тепло-, водо-, нефте- и газоснабжения / А. П. Меренков, Е. В. Сеннова, С. В. Сумароков, В. Г. Сидлер, Н. Н. Новицкий, В. А. Стенников, В. Р. Чупин. Новосибирск: ВО "Наука", Сибирская издательская фирма, 1992. 407 с.
2. Стенников В. А., Сеннова Е. В., Ощепкова Т. Б. Методы комплексной оптимизации развития теплоснабжающих систем // Энергетика. Изв. РАН. 2006. № 3. С. 44–54.
3. Стенников В. А., Барахтенко Е. А., Соколов Д. В. Методы комплексного развития и реконструкции теплоснабжающих систем с применением современных информационных технологий // Промышленная энергетика. 2012. № 4. С. 17–22.
4. Евгеньев Г. Б. Интеллектуальные системы проектирования: учеб. пособие. М.: Изд-во МГТУ им. Н. Э. Баумана, 2009. 334 с.
5. Гаврилова Г. А., Хорошевский В. Ф. Базы знаний интеллектуальных систем. СПб: Питер, 2000. 384 с.
6. Соколов Д. В., Стенников В. А., Ощепкова Т. Б., Барахтенко Е. А. Программный комплекс нового поколения для схемно-параметрической оптимизации многоконтурных теплоснабжающих систем // Теплоэнергетика. 2012. № 4. С. 1–6.
7. Стенников В. А., Барахтенко Е. А., Соколов Д. В. Применение метапрограммирования в программном комплексе для решения задач схемно-параметрической оптимизации теплоснабжающих систем // Программная инженерия. 2011. № 6. С. 31–35.
8. Bartlett J. The art of metaprogramming, Part 1: Introduction to metaprogramming. URL: <http://www-128.ibm.com/developerworks/linux/library/l-metaprogl.html>.
9. Монахов В. В. Язык программирования Java и среда NetBeans. 3-е изд. СПб: БХВ-Петербург, 2011. 704 с.

Л. Д. Черемисинова,

д-р техн. наук, гл. науч. сотр.,

Н. А. Кириенко,

канд. техн. наук, ст. науч. сотр.,

Объединенный институт проблем информатики

Национальной академии наук Беларуси,

cld@newman.bas-net.by

Синтез многоуровневых логических схем с учетом энергопотребления

Предлагается и исследуется метод построения многоуровневой логической схемы из вентилях, которая может служить хорошей основой для синтеза логических схем из библиотечных элементов, выполненных на основе КМОП-технологии. Метод ориентирован на минимизацию площади микросхемы, реализованной на кристалле КМОП СБИС, и среднего значения рассеиваемой ею мощности.

Ключевые слова: автоматизация проектирования, заказные КМОП СБИС, синтез схем с пониженным энергопотреблением

Введение

Успехи в области микроэлектроники сделали возможным размещение на одном кристалле СБИС несколько миллионов транзисторов, работающих на большой синхрочастоте. Однако наряду с огромными возможностями, которые открывают для электронных изделий эти успехи, они обуславливают также и серьезные проблемы, связанные с рассеянием мощности. Эти проблемы не могут быть проигнорированы, так как требуются все более эффективные изделия для различных сфер применения, способные работать длительное время без подзарядки батареи питания и имеющие наряду с этим низкую стоимость.

В последние годы фактор минимизации энергопотребления при проектировании интегральных схем стал играть такую же важную роль, как площадь и быстродействие. Обычно при проектировании электронных устройств приходится добиваться максимального сокращения площади кристалла при ограниченном потреблении электроэнергии. Основные из упоминаемых в литературе [1–4] возможностей сокращения рассеиваемой мощности достигаются на схемотехническом, системном и логическом уровнях. Сокращение энергопотребления микросхем на схемотехническом и системном уровнях достигается за счет выбора удачных технологических решений, на логическом уровне — за

счет построения удачной логической структуры. За счет этого можно достичь сокращения энергопотребления на 10–20 % без ущерба для быстродействия и сложности схемы [5].

Снижение энергопотребления проектируемой схемы может быть обеспечено на разных уровнях логического проектирования. При этом, чем более ранним является этап, тем более важно получать на нем более качественные решения. Одним из таких этапов является технологически независимая оптимизация многоуровневых представлений систем булевых функций, которая непосредственно следует за минимизацией булевых функций в классе дизъюнктивных нормальных форм (ДНФ) и обычно предшествует синтезу схем в технологическом базисе. Сложность и оценка энергопотребления синтезированной схемы в технологическом базисе определяются покрываемой многоуровневой схемой из вентилях. Существенное улучшение характеристик логической схемы в технологическом базисе требует, как правило, получения другого варианта схемы из вентилях.

В настоящей работе рассматривается задача построения и оптимизации многоуровневых представлений систем булевых функций в базисе простых вентилях (типа И, ИЛИ, И-НЕ, ИЛИ-НЕ), которые могли бы служить основой для синтеза логических схем из библиотечных элементов, выполненных на основе технологии статических КМОП-схем, которая в настоящее время является доминирующей технологией в области цифровых СБИС. Оптимизация логической структуры проводится в целях уменьшения интенсивности переключений сигналов на ее полюсах, что позволяет сократить динамическое рассеяние мощности в результате переключений сигналов на узлах КМОП-схем. За первый критерий качества синтезируемой логической схемы из вентилях принимается ее сложность (используется оценка по Квайну), за второй критерий — прогнозируемое энергопотребление соответствующей схемы в технологическом базисе (оцениваемое суммой переключательных активностей элементов схемы).

1. Оценка вариантов оптимизации схемы по энергопотреблению

Известно, что около 80 % энергии, рассеиваемой микросхемой, выполненной по КМОП-технологии, приходится на динамическую составляющую [2–4], порождаемую нестационарным поведением узлов микросхемы. Подавляющая часть этой энергии потребляется во время переключения значений сигналов на узлах схемы. Отсюда следует, что более активные в переключательном плане КМОП-схемы потребляют больший объем энергии. Таким образом, энергопотребление существенно зависит от

переключательной активности элементов схемы, а она, в свою очередь, определяется последовательностью подаваемых входных воздействий на КМОП-схемы, т. е. динамикой функционирования.

На логическом уровне, когда схемы еще нет и часто неизвестен даже технологический базис, в котором она будет реализована, энергопотребление схемы может быть снижено путем такого ее преобразования, которое обеспечивает уменьшение переключательной активности логической схемы без изменения ее функциональности [2, 9]. Для оценки предпочтительности вариантов оптимизации схемы на логическом уровне может быть использовано количественное изменение переключательной активности результирующей схемы при выборе этих вариантов. Такой подход к оценке энергопотребления дает возможность сравнивать варианты реализации схемы в процессе ее проектирования, что позволяет уже на логическом уровне проектировать схемы, потенциально имеющие низкое энергопотребление.

В основе методов оценки переключательной активности лежит подход, основанный на вероятностных характеристиках входных сигналов и функционально-структурных свойствах исследуемой схемы [10]. Подход предполагает задание вероятностей переключения сигналов на входе схемы, отражающих частоту смены их значений, и используется для вычисления вероятностей переключения сигналов на выходах узлов схемы. В литературе предлагается множество вероятностных методов оценки энергопотребления логических схем [3, 4, 9, 10]. Для оценки вариантов оптимизации схемы на логическом уровне, когда неизвестен целевой базис проектирования и проводится лишь сравнительная оценка вариантов, достаточным представляется использование простых оценок изменения переключательной активности при выборе вариантов оптимизации. Будем проводить эту оценку при следующих предположениях [2–4, 10]:

- изменения на входах схемы распространяются через все ее элементы мгновенно, а значит, все переходы в схеме происходят одновременно;
- для каждого входного полюса узла имеет место временная независимость, предполагающая, что значение сигнала в любом такте синхронизации не зависит от его значений в предшествующих тактах;
- входные полюсы узла пространственно независимы, что означает отсутствие корреляции значений сигналов на них (это может быть вызвано, например, наличием разветвлений на выходах элементов или обратных связей).

Различают [10] вероятность p_i появления сигнала 1 (0) на некотором i -м полюсе и вероятность смены сигнала на этом полюсе. Первая вероятность p_i называется сигнальной вероятностью (вероятностью появления сигнала 1) и определяется средней до-

лей тактов, на которых сигнал на i -м полюсе имеет единичное значение. Вторая вероятность $p_i^{1 \rightarrow 0}$ (или $p_i^{0 \rightarrow 1}$) есть вероятность смены значения сигнала с 1 на 0 (или с 0 на 1) и определяется средней долей тактов, на которых сигнал на i -м полюсе меняет свое значение по сравнению со значением в предшествующем такте. При сделанных предположениях вероятность $p_i^{1 \rightarrow 0}$ ($p_i^{0 \rightarrow 1}$) равна произведению вероятности появления на нем сигнала 1 (0) в одном такте на вероятность того, что в следующем такте на нем появится 0 (1). Соответственно переключательная активность i -го полюса схемы равна $E_i = p_i^{1 \rightarrow 0} + p_i^{0 \rightarrow 1}$ или (в предположении, что $0 < p_i < 1$)

$$E_i = 2p_i(1 - p_i). \quad (1)$$

Например, если сигнальная вероятность полюса у $p_y = 0,2$, то переключательная активность этого полюса $E_y = 0,32$, если $p_y = 0,5$, то $E_y = 0,5$.

Вероятность p_e появления сигнала 1 на выходе элемента e существенно зависит от вероятностных характеристик p_i сигналов на его входах и от функции, реализуемой этим элементом. Если сигналы на входах элемента не коррелируют в пространстве и времени, то сигнальные вероятности на выходе простых элементов типа инвертора, И и ИЛИ с $n(e)$ входными полюсами подсчитываются исходя из таблиц истинности реализуемых ими функций:

$$p_e^- = 1 - p_i; \quad p_e^+ = \prod_{i=1}^{n(e)} p_i; \quad p_e^\vee = 1 - \prod_{i=1}^{n(e)} (1 - p_i). \quad (2)$$

2. Распространение вероятностной информации через схему

Если заданы сигнальные вероятности входных сигналов схемы, то они могут быть распространены через всю схему на ее выходные полюсы, и таким образом подсчитаны переключательные активности всех полюсов схемы и, соответственно, переключательная активность схемы в целом.

Следует принимать во внимание, что даже, если требование пространственной (и временной) независимости выполняется для входных сигналов схемы, оно может не иметь места для входных сигналов внутренних элементов схемы (как результат наличия разветвлений на выходах элементов и линий обратной связи). В этом случае вычисленные по формулам (1), (2) вероятности имеют погрешности. Более точные методы подсчета сигнальной вероятности для некоторого узла схемы (основанные, например, на использовании бинарных диаграмм решений — BDD [11], алгебраических выражений [3], плотности вероятности [10]) достаточно трудно-

емки, и для оценки вариантов оптимизации достаточно, как правило, довольствоваться оценками, даваемыми с применением формул (1), (2).

3. Проблема синтеза многоуровневой сети

Библиотечный базис КМОП СБИС содержит достаточно широкий спектр различных логических элементов, среди которых центральное место занимает комбинационная логика: элементы И, ИЛИ, И-НЕ, ИЛИ-НЕ на разное число входов, обычно, как в базисе [12], 2, 3, 4 входа, а также элементы, представляемые древообразными схемами (из вентилях И, ИЛИ) с числом входов не более 4 и числом уровней 2—4.

За сложность (цену) библиотечного элемента можно принять число транзисторов его микросхемы, это число прямо связано с площадью, занимаемой элементом на кристалле. Логическую эффективность элемента количественно можно оценить ценой реализации одного полюса представляющей его древообразной структуры, измеряемой отношением числа транзисторов к числу полюсов этой структуры. Чем меньше значение этого коэффициента, тем выше функциональная эффективность элемента. Например, наиболее эффективными КМОП-элементами серии К1574 [8, 12] являются сильно структурированные логические сети, такие как 2-2И-2ИЛИ-НЕ и 2И-2ИЛИ-2И-НЕ (и двойственные им 2-2ИЛИ-2И-НЕ, 2ИЛИ-2И-2ИЛИ-НЕ), наименее эффективными — инвертор и двухвходовые вентили.

В большинстве систем проектирования процесс логического синтеза делится на два этапа: технологически независимая оптимизация и технологическое отображение [6]. Цель первого этапа заключается в построении и минимизации сложности многоуровневой сети из простых вентилях, измеряемой, как правило, числом вентилях и оценкой энергопотребления на логическом уровне. Цель второго этапа заключается в оптимальном переводе сети из вентилях в технологический базис по критериям площади и энергопотребления. Такой подход не предполагает кардинальную перестройку схемы, полученной на этапе технологически независимой оптимизации. Основной подход к решению задачи технологического отображения базируется на покрытии сети из вентилях подсхемами, реализующими библиотечные элементы и представленными в том же базисе, что и покрываемая сеть. При этом сеть должна состоять из вентилях с ограниченным числом входных полюсов [8].

4. Получение факторизованной формы системы ДНФ

В настоящее время неизвестны методы эффективной булевой декомпозиции, поэтому так же как и в известных системах автоматизации логического

проектирования [13—14], здесь используются методы алгебраической декомпозиции, в основе которых лежит построение факторизованных форм (или ДНФ) [6, 8]. Факторизованная форма, по сути, является алгебраической формой задания многоуровневого представления ДНФ. Специальный термин вводится для обозначения того факта, что заданная система логических уравнений подготовлена для многоуровневой реализации, т. е. удовлетворяет требованиям, накладываемым технологическим базисом, в котором предполагается реализовать описываемое ею устройство управления. Система факторизованных форм может представляться системой логических уравнений, задаваемых в виде ДНФ, в которых входные переменные и их инверсии (x и \bar{x}) рассматриваются как независимые и называются литералами. Наряду с входными и выходными переменными, которые встречаются либо в правых, либо в левых частях уравнений, имеются и внутренние переменные, которые встречаются в обеих частях уравнений.

Цель предлагаемого метода синтеза — построение сети из вентилях И и ИЛИ, допускающей наиболее эффективное покрытие библиотечными элементами. Эта сеть будет представлять лес деревьев, каждое дерево описывается факторизованной формой с заданными ограничениями на максимальные числа n_{\max}^k и m_{\max}^d аргументов многоместных операций конъюнкций и дизъюнкций: для рассматриваемого библиотечного базиса $n_{\max}^k = 4$ и $m_{\max}^d = 3$.

Для преобразования исходной минимизированной системы ДНФ в факторизованную форму можно воспользоваться процедурой алгебраической декомпозиции путем поиска факторов — общих частей конъюнкций или ДНФ системы. Факторизация выполняется отдельно (для каждой ДНФ) и совместно (для всей системы). В первом случае выполняется вынесение литералов за скобки представления ДНФ одной функции и получается сильно структурированная факторизационная форма. Во втором случае находятся факторы (части) для выражений сразу всех ДНФ системы, факторами служат части конъюнкций и дизъюнкций (ДНФ системы). Фактор реализуется однократно, а его литералы заменяются соответствующей внутренней переменной во всех выражениях, из которых он извлекается.

На этапе совместной факторизации (имея в виду последующую отдельную факторизацию) целесообразно выделять факторы, входящие в не менее, чем две конъюнкции или дизъюнкции, и имеющие длину, не менее n_{\min}^k и m_{\min}^d . Для рассматриваемого библиотечного базиса целесообразно принять $n_{\min}^k \in \{2, 3\}$ и $m_{\min}^d = 2$. Ограничение $n_{\min}^k = 3$ связано с тем, что двухвходовые вентили И, ИЛИ, И-НЕ, ИЛИ-НЕ являются наименее эффективными (наиболее дорогими), в то время как наиболее эффективны вентили, имеющие многоуровневую

структуру из двухходовых вентилях И, ИЛИ. Может оказаться более выгодным допускать покрытие вентилях сети с некоторым "перекрытием", для этого достаточно оставлять "мелкоблочную декомпозицию" на этап раздельной факторизации ДНФ системы, а для нескольких функций находить общие факторы большей длины.

За первый критерий качества искомой факторизационной формы принимается ее сложность по Квайну, за второй критерий — прогнозируемая суммарная переключаемая активность ее полюсов при заданных оценках сигнальных вероятностей на входах. Предлагаемый ниже метод наиболее эффективен в плане построения схем с меньшим энергопотреблением в том случае, когда переключаемые активности входных полюсов различны (есть и "плохие", близкие к 0,5, и "хорошие", близкие к 0). Суть метода в том, чтобы сигналы с наиболее "активных" входов, переключаемая активность которых близка к 0,5, снимались как можно ближе к выходу схемы и использовались в схеме реже (питали меньшее число элементов), а наиболее длинные цепочки сигналов начинались от наименее "активных" входов, переключаемая активность которых близка к 0.

Исходя из принятых критериев оптимальности и учитывая специфику КМОП-базиса, преобразование исходной системы ДНФ в факторизованную форму (которой соответствует многоуровневая реализация из вентилях с ограниченным числом входов) разбивается на следующие этапы.

1. Совместная минимизация системы булевых функций, представленных исходными логическими выражениями, с учетом их полярности [8], диктующей выбор ДНФ или ее инверсии для реализации каждой из функций в зависимости от их сравнительной сложности.

2. Нетривиальная совместная факторизация системы ДНФ [8]: выделение факторов (частей конъюнкций или ДНФ), имеющих длину в интервале

$$n_{\min}^k \leq r^k \leq n_{\max}^k \quad (\text{конъюнкции})$$

$$\text{или } m_{\min}^d \leq r^d \leq m_{\max}^d \quad (\text{дизъюнкции}) \quad (3)$$

и входящих в не менее, чем два выражения.

3. Раздельная факторизация каждой из ДНФ, полученной на втором этапе, путем вынесения за скобки литералов, а затем понижения рангов всех конъюнкций и дизъюнкций до n_{\max} , m_{\max} [7].

Преобразование на каждом из трех выделенных этапов выполняется с учетом минимизации переключаемой активности. Результатом выполнения этапов будет являться система факторизованных форм, представляемая в графическом виде лесом деревьев.

5. Совместная оптимизация факторизованных форм с учетом энергопотребления

Задача совместной нетривиальной факторизации множества выражений (конъюнкций или дизъюнкций) состоит в нахождении совокупности факторов, обеспечивающей:

1) выполнение ограничений (3) на длину факторов;

2) минимизацию стоимости схемной реализации факторизованных форм;

3) минимизацию переключаемой активности схемной реализации. Сначала факторизации подвергается множество выражений, соответствующих конъюнкциям всех ДНФ системы. После этого аналогичной процедуре подвергается множество выражений, соответствующих дизъюнкциям ДНФ системы. Разница только в том, что в первом случае литералами являлись переменные и их инверсии, а во втором случае — переменные, приписанные конъюнкциям.

В качестве методов поиска факторов, из которых выбирается наилучший, используют методы, описанные в работе [8]. Ключевым вопросом при выборе очередного фактора является вопрос оценки их качества: стоимостью и энергосберегающей. Доминирующей считается стоимостная оценка, так как от нее в значительной степени зависит и энергопотребление схемы. Стоимостную оценку качества фактора s с порождающим множеством U_s выражений (из которых он выделяется) можно упрощенно определить разницей числа литералов системы выражений до и после выделения фактора, т. е. разницей числа литералов, покрытых минором (представляющим фактор) булевой матрицы, задающей исходное множество выражений, и числа литералов, заменивших их после выделения фактора:

$$T_s = c(s)(|U_s| - 1),$$

где $c(s)$ — длина фактора s .

Аналогично энергосберегающее качество фактора можно также оценивать количественно, тем выигрышем в переключаемой активности искомой схемы, который дает выделение этого фактора. В факторизованном множестве выражений сигнальные вероятности и, соответственно, переключаемые активности всех выражений не изменятся по сравнению с их значениями в исходном множестве. Но изменится нагрузка полюсов схемы, соответствующих литералам, входящим в фактор s : она уменьшится на $(|U_s| - 1)$, соответственно, изменится и суммарная переключаемая активность полюсов схемы, реализующих факторизованное множество выражений. Энергосберегающее качество фактора $s = z_1 z_2 \dots z_l$ предлагается оценивать как

$$P_s = \sum_{z_i \in s} E(z_i)(|U_s| - 1).$$

6. Раздельная оптимизация факторизованных форм с учетом энергопотребления

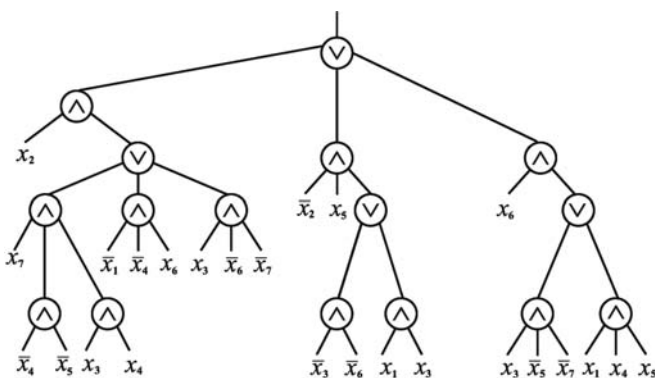
Раздельная факторизация основана на итеративном вынесении общих литералов конъюнкций каждой из ДНФ D за скобки, т. е. на декомпозиции [7]: $D = k(A) + B$, где D , A и B — некоторые ДНФ, а k — конъюнкция литералов, общих для всех конъюнкций из ДНФ A . Конъюнкция k выбирается следующим образом. Ее ядром (начальным значением) является некоторый "лучший" литерал x конъюнкций декомпозируемой ДНФ. Вместе с этим литералом за скобки выносятся и другие литералы, встречающиеся во всех тех конъюнкциях, из которых вынесен литерал x . Лучшим считается тот литерал x , который входит в максимальное число l конъюнкций из D , а из равноценных по этому критерию — тот, что имеет максимальное значение переключательной активности.

Выбор литерала с максимальным значением переключательной активности обосновывается следующими соображениями:

1) энергонагрузка на полюс схемы, имеющий переключательную активность E_x и соответствующий литералу x , выносимому из l выражений, уменьшается на величину $(l - 1)E_x$;

2) наиболее активный в переключательном плане сигнал будет подаваться на схему ближе к ее выходу, это позволит уменьшить суммарную переключательную активность древообразной схемы.

После вынесения литерала за скобки ДНФ D заменяется парой новых ДНФ A и B , для которых, в свою очередь, выполняются операции вынесения общих литералов конъюнкций за скобки. После окончания итеративного процесса вынесения общих литералов конъюнкций за скобки могут остаться конъюнкции u_k ранга $s(u_k)$, большего, чем n_{\max} . В этом случае каждая такая конъюнкция u_k разлагается независимо от других: из нее последовательно выделяются факторы — множества ее литералов — и выносятся за скобки. При этом в фактор включаются в первую очередь литералы, соответствующие полюсам с наименьшей переключательной активностью. Это обосновывается тем, что, во-первых,



Многоуровневая сеть в базе вентилей И, ИЛИ

число их вхождений в факторизованную форму не изменяется (равно 1 в обоих случаях), во-вторых, наиболее активные в переключательном плане сигналы желательно подавать на входы элементов схемы как можно ближе к ее выходу (следовательно, их лучше не включать в фактор, реализуемый элементом И).

После того как все конъюнкции скобочной формы будут иметь ранги не более n_{\max} , процедуре, аналогичной построению скобочного выражения для множества конъюнкций, подвергается множество дизъюнкций.

Например, на рисунке показана многоуровневая логическая схема в базе вентилей 2И, 3И, 4И, 2ИЛИ, 3ИЛИ, 4ИЛИ, реализующая ДНФ

$$D = (x_1 x_2 \bar{x}_4 x_6 \vee \bar{x}_2 \bar{x}_3 x_5 \bar{x}_6 \vee x_3 \bar{x}_5 x_6 \bar{x}_7 \vee x_2 \bar{x}_4 \bar{x}_5 x_7 \vee x_1 \bar{x}_2 x_3 x_5 \vee x_2 x_3 x_4 x_7 \vee x_1 x_4 x_5 x_6 \vee x_2 x_3 \bar{x}_6 \bar{x}_7),$$

при задании следующих значений сигнальных вероятностей:

$$p_{x1} = 0,3; p_{x2} = 0,5; p_{x3} = 0,3; p_{x4} = 0,2; \\ p_{x5} = 0,1; p_{x6} = 0,4; p_{x7} = 0,4.$$

7. Результаты экспериментального исследования

Предложенный метод получения многоуровневого представления системы ДНФ, ориентированный на оптимизацию сложности и энергопотребления, был реализован в качестве проектной процедуры "Синтез схем из многоходовых вентилей" в рамках программного комплекса проектирования цифровых интегральных КМОП-микросхем с пониженным энергопотреблением, разработанного в лаборатории логического проектирования Объединенного института проблем информатики Национальной академии наук Беларуси. Исходными данными для процедуры является функциональное описание схемы в виде системы ДНФ и сигнальные вероятности входных сигналов схемы. Настраиваемыми параметрами процедуры являются ограничения на ранги конъюнкций, дизъюнкций (3) и минимальное значение n_{dl} мощности порождающего множества факторов.

В табл. 1 и 2 приведены имена тестовых систем ДНФ (столбец 1) и их параметры: число аргументов, функций и конъюнкций (через запятую в столбце 2)). Описания 1—12 (табл. 1, 2) взяты из набора тестовых примеров [15], описания 13—25 сгенерированы случайным образом. Для каждого тестового примера были созданы файлы сигнальных вероятностей входных сигналов, значения которых выбирались случайным образом в интервале (0, 1).

В целях оценки влияния введенных эвристик учета переключательной активности на энергопотребление многоуровневой структуры из вентилей, соответствующей полученной системе факторизо-

Таблица 1

Исследование процедур синтеза при $n_{\max} = m_{\max} = 4, n_{dl} = 2$

	Имя	Параметры	Синтез схем без учета энергопотребления			Синтез схем с учетом энергопотребления			Выигрыш, %	
			Перекл. активн.	Время	Слож- ность	Перекл. активн.	Время	Слож- ность	Перекл. активн.	Слож- ность
			1	2	3	4	5	6	7	8
1	b12	15, 9, 431	47,09	0,16	299	47,09	0,19	299	0,00	0,00
2	b9	16, 5, 123	107,76	0,2	433	107,76	0,22	433	0,00	0,00
3	br1	12, 8, 34	34,79	0,05	266	33,51	0,05	250	3,68	6,02
4	br2	12, 8, 35	25,8	0,05	187	25,66	0,05	192	0,54	-2,67
5	dc2	8, 7, 58	59,65	0,06	293	59,13	0,06	293	0,87	0,00
6	frg1	28, 3, 119	97,9	0,91	427	97,9	0,94	427	0,00	0,00
7	in0	15, 11, 138	154,9	0,3	994	153,5	0,36	1017	0,90	-2,31
8	in2	19, 10, 137	188,93	0,31	1005	186,15	0,36	995	1,47	1,00
9	mlp4	8, 8, 256	327,15	0,61	1615	323,8	0,7	1625	1,02	-0,62
10	root	8, 5, 256	230,3	0,42	1067	195,46	0,52	998	15,13	6,47
11	tms	8, 16, 30	87,07	3,98	481	85,99	1,8	477	1,24	0,83
12	z9sym	9, 1, 420	97,69	1,55	476	97,69	2,22	476	0,00	0,00
13	GenP1	20, 4, 50	126,87	0,22	636	126,79	0,24	636	0,06	0,00
14	GenP2	30, 10, 100	396,8	0,92	1910	392,51	1,11	1884	1,08	1,36
15	GenP3	30, 10, 300	991,88	6,97	4782	976,22	8,92	4696	1,58	1,80
16	GenP4	30, 8, 400	1104,85	6,91	4794	1093,48	8,5	4780	1,03	0,29
17	GenP5	20, 6, 400	697,63	3,17	2991	690,72	4	3003	0,99	-0,40
18	GenP7	30, 12, 50	162,7	0,13	679	162,26	0,13	679	0,27	0,00
19	GenP8	30, 12, 50	208,09	0,16	889	206,49	0,16	886	0,77	0,34
20	GenP9	30, 12, 400	1182,83	8,78	5214	1164,93	9,72	5129	1,51	1,63
21	GenP14	30, 5, 500	1008,71	6,3	3967	992,62	7,72	3951	1,60	0,40
22	GenP17	30, 5, 400	1034,83	6,08	4348	1027,78	6,94	4345	0,68	0,07
23	GenP18	30, 12, 300	695,87	2	2821	688,52	2,13	2828	1,06	-0,25
24	GenP23	30, 10, 490	1258,83	10,2	5674	1238,54	11,22	5600	1,61	1,30
25	GenP24	30, 6, 550	880,05	2,45	3435	839,02	2,42	3386	4,66	1,43
	Среднее								1,67	0,67

Таблица 2

Исследование процедур синтеза при $n_{\max} = m_{\max} = 3, n_{dl} = 2$

	Имя	Параметры	Синтез схем без учета энергопотребления			Синтез схем с учетом энергопотребления			Выигрыш, %	
			Перекл. активн.	Время	Слож- ность	Перекл. активн.	Время	Слож- ность	Перекл. активн.	Слож- ность
			1	2	3	4	5	6	7	8
26	b12	15, 9, 431	78,43	0,45	436	78,43	0,48	436	0,00	0,00
27	b9	16, 5, 123	85,79	0,45	460	85,79	0,55	460	0,00	0,00
28	br1	12, 8, 34	43	0,06	258	34,49	0,06	213	19,79	17,44
29	br2	12, 8, 35	59,22	0,06	249	57,74	0,06	253	2,50	-1,61
30	dc2	8, 7, 58	46,76	0,06	266	46,28	0,07	266	1,03	0,00
31	frg1	28, 3, 119	91,09	11,3	443	91,09	11,3	443	0,00	0,00
32	in0	15, 11, 138	139,11	0,42	985	138,96	0,5	969	0,11	1,62
33	in2	19, 10, 137	188,38	0,36	1008	182,34	0,4	972	3,21	3,57
34	mlp4	8, 8, 256	242,24	0,78	1433	241,93	0,94	1404	0,13	2,02
35	root	8, 5, 256	149,02	0,78	1090	145,67	0,56	1084	2,25	0,55
36	tms	8, 16, 30	108,82	0,08	467	108,09	0,08	461	0,67	1,28
37	z9sym	9, 1, 420	174,24	3,08	777	174,24	4,17	777	0,00	0,00
38	GenP1	20, 4, 50	126,56	0,16	678	122,07	0,17	647	3,55	4,57
39	GenP2	30, 10, 100	426,67	1,3	2178	416,3	1,55	2152	2,43	1,19
40	GenP3	30, 10, 300	967,66	10,03	5065	957,58	11,92	5006	1,04	1,16
41	GenP4	30, 8, 400	1058,53	9,14	4945	1011,36	9,22	4945	4,46	0,00
42	GenP5	20, 6, 400	653,24	4	3073	642,02	4,78	3061	1,72	0,39
43	GenP7	30, 12, 50	153,15	0,14	709	150,17	0,15	709	1,95	0,00
44	GenP8	30, 12, 50	193,47	0,17	911	191,49	0,19	913	1,02	-0,22
45	GenP9	30, 12, 400	1060,46	11,61	5202	1068,94	13,69	5245	-0,80	-0,83
46	GenP14	30, 5, 500	877,55	8,27	4183	865,71	10,22	4184	1,35	-0,02
47	GenP17	30, 5, 400	1012,66	8,44	4465	991,29	9,58	4403	2,11	1,39
48	GenP18	30, 12, 300	712,31	2,55	2895	684,31	2,48	2863	3,93	1,11
49	GenP23	30, 10, 490	1228,83	13,39	5751	1222,67	15,42	5779	0,50	-0,49
50	GenP24	30, 6, 550	868,71	4,33	3517	828,02	4,67	3483	4,68	0,97
	Среднее								2,30	1,36

ванных форм, была разработана альтернативная процедура синтеза схем из многовыходных вентилях, отличающаяся от основной только тем, что в процессе выполнения совместной и раздельной факторизации систем ДНФ (разделы 5 и 6) не учитывается энергосберегающий критерий качества факторов.

Эксперименты проводили для обеих вышеупомянутых процедур синтеза схем из многовыходных вентилях с двумя наборами параметров (3). Результаты исследования для значений параметров $2 \leq r^k \leq 4$; $2 \leq r^d \leq 4$, $n_{dl} = 2$ приведены в табл. 1, для значений $2 \leq r^k \leq 3$; $2 \leq r^d \leq 3$, $n_{dl} = 2$ — в табл. 2. В столбцах 3—5 обеих таблиц для синтеза схем из многовыходных вентилях без учета энергопотребления, в столбцах 6—8 с учетом энергопотребления приведены следующие результаты эксперимента: переключательная активность (столбцы 3, 6) и сложность по Квайну (столбцы 5, 8) схемных реализаций; время выполнения процедур (столбцы 4, 7).

В столбцах 9 и 10 приведены сравнительные оценки выигрыша (в процентном отношении) процедуры синтеза схем с учетом энергопотребления по сравнению с процедурой синтеза без учета энергопотребления по переключательной активности и сложности полученного решения. В нижней строке таблиц представлены средние значения сравнительных оценок выигрыша.

Результаты эксперимента позволили сделать следующие заключения:

1) меньшие значения параметров $n_{\max} = m_{\max} = 3$ дали возможность получить большее среднее значение выигрыша по переключательной активности и сложности по сравнению с $n_{\max} = m_{\max} = 4$;

2) метод синтеза многоуровневых схем с учетом энергопотребления позволяет получить выигрыш, хотя и небольшой, как по переключательной активности, так и по сложности схемных реализаций;

3) учет энергопотребления в процессе оптимизации многоуровневых представлений позволил не проиграть в сложности результирующих схем;

4) времена выполнения процедур синтеза с учетом и без учета энергопотребления отличаются незначительно.

В целом, выигрыш по переключательной активности сильно зависит от сложности исходного функционального описания и от наперед заданных значений переключательных активностей входных полюсов схемы. Полученный выигрыш по переключательной активности можно считать удовлетворительным, так как он не привел к проигрышу по сложности. Кроме того, синтез многоуровневых схем из вентилях является только одним из этапов синтеза схемы в заданном технологическом базисе, и соответственно, полученный выигрыш является частью общей интегральной оценки степени влияния направленного проектирования на энергопотребление целевой схемы.

Предложен метод построения многоуровневых представлений систем булевых функций, которые призваны служить основой для синтеза логических схем из библиотечных элементов, выполненных на основе КМОП-технологии. Метод ориентирован на минимизацию площади микросхемы, реализованной на кристалле КМОП СБИС, и среднего значения рассеиваемой ею мощности. Реализована процедура синтеза схем из многовыходных вентилях в рамках программного комплекса проектирования цифровых интегральных КМОП-микросхем с пониженным энергопотреблением. Проведенные экспериментальные исследования показали, что предложенный метод синтеза позволяет получить достаточно стабильный выигрыш в оценке энергопотребления проектируемой схемы.

Список литературы

1. Рабаи Ж. М. Цифровые интегральные схемы. Методология проектирования. М.: Вильямс, 2007. 912 с.
2. Pedram M. Power Minimization in IC Design: Principles and Applications // ACM Trans. Design Automation Electronic Systems. 1996. Vol. 1. P. 3—56.
3. Roy K., Prasad S. C. Low Power CMOS VLSI Circuit Design. New York: John Wiley and Sons Inc., 2000. 376 p.
4. Benini L., Micheli G. De. Logic Synthesis for Low Power / Logic Synthesis and Verification / Eds. S. Hassoun, T. Sasao, R. K. Brayton. — Boston, Dordrecht, London: Kluwer Academic Publishers, 2002. P. 197—223.
5. Power Compiler. Automatic Power Management within Galaxy™ Implementation Platform: URL: http://www.synopsys.com/Tools/Implementation/RTL_Synthesis/Documents/power_compiler_ds.pdf.
6. Брейтон Р. К., Хэтчел Г. Д., Санджованни-Винченцелли А. Л. Синтез многоуровневых комбинационных логических схем // ТИИЭР. 1990. Т. 78, № 2. С. 38—83.
7. Торопов Н. Р., Черемисинова Л. Д. Синтез комбинационных многоуровневых сетей, отображаемых в базис базовых матричных кристаллов // Вестник Томского гос. университета. 2004. № 9 (1). С. 249—254.
8. Черемисинова Л. Д. Синтез и оптимизация комбинационных структур СБИС. Минск: ОИПИ НАН Беларуси, 2005. 236 с.
9. Черемисинова Л. Д. Оценка энергопотребления КМОП-схем на логическом уровне // Информационные технологии. 2010. № 8. С. 27—35.
10. Najm F. N. A survey of Power Estimation Techniques in VLSI Circuits // IEEE Trans. on VLSI. 1994. № 12. P. 446—455.
11. Najm F., Burch R., Yang P., Hajj I. Probabilistic simulation for reliability analysis of CMOS VLSI circuits // IEEE Trans. on Computer-Aided Design. — 1990. Vol. 9. N 4. P. 439—450.
12. Лукошко Г., Коннов Е. КМОП базовые матричные кристаллы серии K1574 // Радиолюбитель. 1997. № 9. С. 39—40.
13. Berkeley Logic Synthesis and Verification Group. ABC: A System for Sequential Synthesis and Verification. — Release 00127p. URL: <http://www-cad.eecs.berkeley.edu/~alanmi/abc>
14. Mishchenko A., Eén N., Brayton R., Jang S., Ciesielski M., Daniel T. Magic: An Industrial-Strength Logic Optimization, Technology Mapping, and Formal Verification Tool // Proc. of IWLS'10, 2010. P. 124—127.
15. Berkley PLA test set. URL: <http://www1.cs.columbia.edu/~cs4861/sis/espresso-examples/>.

УДК 519.246.8

Р. К. Мамедов,

д-р техн. наук, проф., зав. каф.,
e-mail: Rahim1951@mail.ru,

Э. И. Владимирский,

канд. техн. наук, ст. науч. сотр.,
e-mail: Eduard.Vladimirsky@hotmail.com,

С. Р. Мустафаева, докторант,

e-mail: mustafayeva_81@mail.ru
Азербайджанская государственная
нефтяная академия, г. Баку

Аномальная диффузия и ее влияние на хаотические системы

Представлены новые структуры взаимодействующих систем типов: Лоренц — Чен — аномальная диффузия, Рабинович — Фабрикант — аномальная диффузия. Получены рекуррентные диаграммы, характеризующие особенности структур.

Ключевые слова: наблюдаемая, система Лоренца — Чена, система Рабиновича — Фабриканта, аномальная диффузия, нелинейный рекуррентный анализ

Динамический хаос подобно случайному процессу требует статистического описания. Обычно рассчитываются или измеряются вероятностные характеристики, такие как стационарное распределение вероятности по аттрактору, корреляционные функции, спектры мощности и др. Хаотические колебания, математическим образом которых являются разные типы хаотических аттракторов, характеризуются различными статистическими свойствами и степенью чувствительности к воздействию шума.

В связи с глубокими различиями между случайными и хаотическими системами [1], а также с учетом гетерогенности информационных потоков, циркулирующих в этих структурах, возникает проблема их *взаимодействия*. То есть возникает вопрос, как влияет стохастическая составляющая на хаотический процесс.

Оказывается, что результирующее хаотическое поведение динамической системы существенно объяснено своим возникновением не только действием динамических (детерминистских) законов, но и наличием статистических факторов [2].

Кроме того, в последнее время в связи с развитием синергетической концепции выяснилось интересное свойство таких систем: нарастание интенсивности стохастических слагаемых может приводить не только к росту беспорядка, но и образованию упорядоченных структур, т. е. самоорганизации системы, уменьшению ее энтропии [3].

Очевидно, последнее связано с неконсервативностью объема фазового пространства, поскольку в полной замкнутой системе энтропия не может уменьшаться.

Этот аспект важен в контексте установления природы экспериментально полученной наблюдаемой, что дает возможность прогнозировать состояние аттрактора системы.

Здесь важно отметить, что методы параметрического анализа взаимодействия стохастических и хаотических систем носят сложный характер, т. е. требуют либо достаточно данных, либо стационарных последовательностей, которые не всегда можно получить на практике. Использование нелинейного рекуррентного анализа позволяет разрешить эту проблему [4].

В работе рассматривается взаимодействие аномальной диффузии с хаосоподобными и хаотическими системами в контексте получения новых структур.

Прежде чем перейти к постановке задачи взаимодействия систем, кратко остановимся на элементах теории.

Нелинейный рекуррентный анализ

Фундаментальным свойством динамических диссипативных систем является рекуррентность (повторяемость) состояний в смысле прохождения последующей траектории достаточно близко к предыдущей. Это свойство было отмечено в 80-х гг. XIX века французским математиком Пуанкаре (Poincaré) и сформировано в виде "теоремы рекуррентности" [4, 5].

Теорема. Пусть T — сохраняющее меру преобразование пространства с конечной мерой (x, μ) и пусть $A \subset X$ — измеримое множество. Тогда для любого натурального $N \subset \mathbb{N}$

$$A(\{x \in A : \{T^n(x)\}_{n \geq N} \subset (X \setminus A)\}) = 0,$$

где T — рекуррентное время; X — произвольное измеримое множество; $\mu(\cdot)$ — вероятностная мера; x — параметр нормализованного ряда; N — длина ряда; \mathbb{N} — множество натуральных чисел.

Однако возможности такого анализа сильно ограничены, так размерность фазового пространства сложной динамической системы, как правило, более 3. В связи с этим в работе [4] был предложен способ отображения m -мерной фазовой траектории состояния системы $x(t)$ длиной N на двумерную квадратную двоичную матрицу размерности $N \times N$, в которой 1 (черная точка) соответствует повторению состояния при некотором времени i в некоторое другое время j , а обе координатные оси являются осями времени. Такое представление было названо *рекуррентной диаграммой* (recurrence plot, RP) и формально выглядит как [4]

$$R_{i,j}^{m,\varepsilon_i} = \theta(\varepsilon_i - \|x_i - x_j\|), \quad x \in R^m, \quad i, j = \overline{1, N}, \quad (1)$$

где N — число рассматриваемых состояний x_i ; ε_i — размер окрестности точки x в момент i ; $\|\cdot\|$ — норма; $\theta(\cdot)$ — функция Хэвисайда.

Как отмечено в работе [5], невозможно обнаружить полную рекуррентность в смысле $x_i = x_j$ (состояние динамической, а особенно — хаотической системы не повторяется полностью эквивалентно начальному состоянию, а подходит к нему сколь угодно близко). Таким образом, *рекуррентность определяется как достаточная близость состояния x_i состоянию x_j* , т. е. рекуррентными являются состояния x_j , попадающие в m -мерную окрестность с радиусом ε_i и центром в x_i . Эти точки x_j называются *рекуррентными точками* (recurrence plots).

Так как $R_{i=j} = 1, i = \overline{1, N}$, то рекуррентная диаграмма всегда содержит черную диагональную линию — *линию идентичности* (line of identity, LOI), под углом $\pi/4$ к осям координат [4].

Здесь важно отметить, что произвольно взятая рекуррентная точка (i, j) не несет какой-либо полезной информации о состояниях во времена i и j . Только вся совокупность рекуррентных точек позволяет восстановить свойства системы.

В контексте использования нелинейного рекуррентного анализа для реализации различных прикладных задач необходимо учесть работу Л. Ю. Колесова и Н. Х. Розова [6], в которой дается новое определение хаотического инвариантного множества для непрерывного полупотока в метрическом пространстве. Отмечено, что предлагаемое определение обобщает известное определение Девани [6] и позволяет учесть специфическую особенность, возникающую в некомпактном и бесконечномерном случае, — так называемый турбулентный хаос.

Новое определение хаоса

В работе [6] было предложено определение хаотичности динамической системы, которое помимо чувствительной зависимости от начальных условий включает также требование *сложности траекторий*. Под сложностью авторы понимают отсутствие свойства рекуррентности. Для придания смысла термину

"сложность" напомним понятие рекуррентности по Биркгофу [6].

Определение 1. Траектория $\gamma(x_0) \subset A$ полупотока φ^t называется рекуррентной, если для любого $\varepsilon > 0$ найдется такое $T = T(\varepsilon) > 0$, что при каждом $t_0 \geq 0$ справедливо включение $\gamma(x_0) \subset O_\varepsilon(\gamma[t, t_0 + T](x_0))$. Здесь $\gamma[t, t_0 + t](x_0)$ — отрезок траектории $\gamma(x_0)$, отвечающий значениям $t \in [t, t_0 + T]$, а символом $O_\varepsilon(\cdot)$ обозначается ε -окрестность соответствующего множества.

Отсюда вытекает следующее определение.

Определение 2 [6]. Траектория $\gamma(x_0)$ полупотока φ^t , принадлежащая инвариантному множеству A , называется *сложной*, если для нее выполняется одна из следующих двух возможностей:

- ω — предельное множество $\omega(x_0)$ этой траектории пусто;
- справедливо равенство $\omega(x_0) = A$, но сама траектория $\gamma(x_0)$ не является *рекуррентной*.

Для оценки содержательности этого определения заметим, что не всякая траектория, не обладающая свойством рекуррентности, оказывается сложной.

Так, из определения 2 с необходимостью следует, что для сложной траектории, принадлежащей компактному множеству, должно выполняться включение $\gamma(x_0) \subset \omega(x_0)$.

Замечание. ω — *предельное множество движения*. Рассмотрим на конечномерном многообразии M динамическую систему

$$\dot{x} = F(x), \quad (2)$$

где $F(\cdot)$ — гладкое векторное поле на M . Решение $x(t)$ этого уравнения называется *движением* на M , а множество принадлежащих M точек

$$\tilde{\gamma} = \{x(t) \forall t\} \quad (3)$$

называется *траекторией* (или *орбитой*) этого движения. Предположим, что задача с начальными данными для уравнения (3) корректно поставлена, и для любого x_0 из M и всех $t \geq 0$ обозначим через $\varphi(x_0, t)$ решение, которое начинается в точке x_0 ; таким образом, если $x(t)$ — какое-либо решение, то

$$x(t) = \varphi(x(0), t). \quad (4)$$

При фиксированном x функция $\varphi(x, t)$ — это движение, тогда как при фиксированном $t \geq 0$ — это взаимно однозначное отображение $x \rightarrow \varphi(x, t)$ многообразия M на себя. Функция $\varphi(x, t)$ называется *полупотоком*; она обладает полугрупповым свойством, а именно, для неотрицательных t и s

$$\varphi(\varphi(x, t), s) = \varphi(x, t + s). \quad (5)$$

При $t = 0$ φ совпадает с тождественным отображением: $\varphi(x, 0) \equiv x$. Будем предполагать, что функция $\varphi(x, t)$ — непрерывная по x и t .

Точка ξ из M называется ω -*предельной точкой* движения $x(t)$, если для произвольно больших вре-

мен $x(t)$ сколь угодно близко подходит к ξ , т. е. если существует такая последовательность $\{t_n\}_1^\infty$, что

$$\left. \begin{aligned} t_n \rightarrow \infty, \\ |x(t_n) - \xi| \rightarrow 0 \end{aligned} \right\} \text{ при } n \rightarrow \infty. \quad (6)$$

Множество всех ω -предельных точек движения называется его ω -предельным множеством и обозначается Ω_x , где x — начальная точка движения; это множество точек замкнуто. Для любой другой точки y той же самой траектории $\Omega_y = \Omega_x$.

Определение 3 [6]. Полупоток φ^t называем хаотическим на инвариантном множестве A при выполнении следующих ограничений:

- $\varphi^t|A$ обладает свойством существенной зависимости от начальных условий;
- найдется такое подмножество $A_0 \subset A$, $\overline{A_0} = A$, что любая траектория $\gamma(x_0)$ с начальным условием $x_0 \in A_0$ является сложной. Отсюда напрашивается вывод, что при формировании математических моделей взаимодействия систем по наблюдаемой в контексте использования нелинейного рекуррентного анализа необходимо учитывать ограничения на сложность траектории хаотических процессов.

В связи с этим имеет место следующее замечание.

Замечание. Полупотоки. Базовый аспект. Пусть W — полное метрическое пространство с метрикой $d(u, v) = d_W(u, v)$.

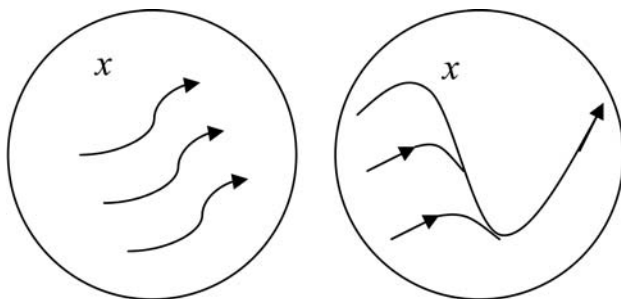
Пусть M — подмножество W , $M \subset W$ и пусть $R = (-\infty, \infty)$, $R^+ = [0, \infty)$. Говорят, что отображение $\tau = \tau(u, t)$, где $\tau : M \times [0, \infty) \rightarrow M$ есть полупоток на M , если:

- 1) $\tau(\omega, 0) = \omega, \forall \omega \in M$;
- 2) оно отвечает полугрупповому свойству, т. е. $\tau(\tau(\omega, s), t) = \tau(\omega, t + s), \omega \in M, \forall s, t \in R^+$;
- 3) отображение $\tau : M \times [0, \infty) \rightarrow M$ непрерывно.

Существует дискретный вариант этих понятий. При фиксировании $\tau > 0$ имеем

$$\tau L^+ = \{n\tau : n = 0, 1, \dots\} \text{ и } \tau L^* = \{n\tau : n = 0, \pm 1, \dots\}.$$

Отображение $\tau(\cdot, t) : M \rightarrow M$ для $t \in \tau L^+$ представляет собой дискретный полупоток на M при непрерывном τ , $\tau(\omega, 0) = \omega, \forall \omega \in M$ и согласно п. 3 $\forall \omega \in M, \forall st \in \tau L^+$.



Поток

Полупоток

Рис. 1

Таким образом, дискретные потоки определяются аналогично, но уже с заменой τL на τL^+ .

В общем случае отображение $\sigma(u, n\tau)$ есть дискретный полупоток на M , если и только если непрерывное отображение $T : M \rightarrow M$ такое, что

$$\sigma(u, n\tau) = T^n(u), \forall u \in M \text{ и } \forall n \in Z^+.$$

На рис. 1 дано схематическое отображение потока и полупотока.

Негауссовы законы случайных блужданий

Броуновским движением, или процессом Винера — Башелье, называют начинающийся в начале координат процесс с независимыми приращениями, имеющими нормальное распределение с нулевым математическим ожиданием и дисперсией, пропорциональной t (рис. 2, см. третью сторону обложки). Винером и Леви было показано, что траектории такого процесса с вероятностью 1 непрерывны и что это свойство выделяет данный процесс из более широкого класса безгранично делимых процессов [7].

Определение 4 [7]. Винеровский процесс (W -процесс) есть однородный процесс с конечной дисперсией, если:

- а) при $0 < t_1 < t_2 < \dots$ случайные величины $X(t_1), X(t_2) - X(t_1), \dots$ — независимы;
- б) $X(t + \tau) - X(t)$ не зависит от t ;
- в) $\lim P(|X(t + \tau) - x(t)| \geq \varepsilon/\tau) = 0$ при $t \rightarrow 0$ и всех $\varepsilon > 0$ (условие Линдберга) — непрерывность траекторий.

Отсюда W -процесс есть однородный процесс с конечной дисперсией.

Леви-процесс

На содержательном уровне Леви-процесс ("полеты Леви") — это движение, состоящее из серий коротких перемещений, причем в промежутках между ними совершаются длинные перемещения. Если прочертить траекторию такого движения, то получится большая фигура, состоящая из маленьких, которые по форме напоминают большую (рис. 3, см. третью сторону обложки).

Здесь важно отметить, что полеты Леви имеют отношение к фракталам, так как в них фрагменты являются подобием целого, иначе говоря, это самоподобные структуры.

Перейдем к формальной стороне Леви-процесса (L -процесс). Определим L -процесс как однородный процесс с автомодельным одномерным распределением

$$p(x, t) = t^{-1/\alpha} g^{(\alpha)}(xt^{-1/\alpha}),$$

где $\alpha = 1/H$ (H — функция Херста), а $g^{(\alpha)}(x)$ — неизвестная пока плотность распределения. Таким образом, переход от W -процесса к L -процессу осуществляется заменой условия конечности дисперсии требованием автомодельности.

Определение 5 [7]. Случайный процесс называется α -устойчивым (стандартным) движением Леви с параметрами $0 < \alpha \leq 2$, $-1 \leq \beta \leq 1$, если:

а) $X(0) \approx 0$;
 б) $X(t)$ является процессом с независимыми приращениями;

в) $X(t + \tau) - X(t) \stackrel{d}{=} \tau^{1/\alpha} S(\alpha, \beta)$ при любых t и τ .

Знак d над равенством означает асимптотическое равенство распределений случайных величин, стоящих по обе стороны от этого знака.

Математическая модель системы 1

Представим модель взаимодействий стохастического процесса φ_{i+1} с хаосоподобным типа Лоренц — Чен в виде [8]

$$\begin{aligned} \dot{x} &= (25\alpha + 10)(y - x) + D\varphi_{i+1}; \\ \dot{y} &= (28 - 35\alpha)x + (29\alpha - 1)y - xz; \\ \dot{z} &= xy - \frac{8 + \alpha}{3}z. \end{aligned} \quad (7)$$

Здесь x — величина, пропорциональная скорости конвекции; y — разность температур между восходящими и нисходящими потоками; z — отклонение вертикального профиля температуры от линейного параметра; D — интенсивность стохастических слагаемых; $\alpha \in [0, 1]$, φ_{i+1} — дискретное стохастическое уравнение вида [9]

$$\varphi_{i+1} = \sqrt{r}\zeta_i + [(1 - \gamma\tau) + \sqrt{r}\xi_i]\varphi_i, \quad (8)$$

где ζ_i и ξ_i — аддитивный и мультипликативный стохастические источники, нормированные условиями белого шума $\langle \zeta_i \zeta_j \rangle = \langle \xi_i \xi_j \rangle = \delta_{ij}$; коэффициент трения γ задает параметр $V = \gamma(1 + \gamma)$, значения которого определяют режим диффузии.

Параметрическое разрешение математических моделей наталкиваются на некоторые трудности. Поэтому изучение таких моделей, особенно при экспериментальных исследованиях, часто реализуется посредством обработки регистрируемых сигналов, произведенных системами.

Определение 6 [1]. Последовательность измерений

$\hat{x} = \{x_n\}_{n=0}^N$ называется детерминировано порожденной, если выполнены следующие условия:

- существуют конечномерная динамическая система f , точка y_n и функция $\varphi(f^n(y_0)) = x_n$;
- расстояние $\text{dist}(f^k x, f^k x') \leq \text{const} \cdot e^{\lambda k} \text{dist}(x, x')$, т. е. максимальный ляпуновский показатель должен быть ограничен;
- функция φ (функция Липшица) непрерывна, т. е. $|\varphi(x) - \varphi(x')| \leq L|x - x'|$ для некоторой величины L , называемой константой Липшица.

Таким образом, исследование наблюдаемого сигнала может дать ответ на принципиальный вопрос о конечности и детерминированности исследуемого сигнала.

Математическая модель системы 2

Представим модель взаимодействия стохастического процесса φ_{i+1} с хаотической системой Рабинович — Фабрикант в виде [10]

$$\begin{aligned} \dot{x}_1 &= -\alpha x_1 + hx_2 + x_2 x_3 + D\varphi_{i+1}; \\ \dot{x}_2 &= -bx_2 + hx_1 - x_1 x_2; \\ \dot{x}_3 &= -dx_3 + x_1 x_2, \end{aligned} \quad (9)$$

где x_1, x_2, x_3 — переменные; a, b, d, h — положительные константы.

Постановка задачи для моделей систем 1 и 2

Пусть заданы системы уравнений (7), (9).

Требуется:

- определить взаимодействующие наблюдаемые, произведенные системами (7), (9);
- определить рекуррентные диаграммы в контексте взаимодействующих наблюдаемых;
- дать топологический и текстурный анализы рекуррентных диаграмм, в результате которых принимается решение о принадлежности классу процессов.

Алгоритм моделирования системы 1

Шаг 1. Проводим моделирование хаосоподобной системы типа Лоренц — Чен в программной среде MATLAB/Simulink [8, 11], $N = 10^5$ точек (реально $5 \cdot 10^3$), представленной схемой (рис. 4, а, см. третью сторону обложки) с аттрактором в плоскости $x - y$ (рис. 4, б, см. третью сторону обложки).

Шаг 2. Реализуем итерационный алгоритм процедуры (8) на множестве, состоящем из $N = 8 \cdot 10^3$ точек (реально $2 \cdot 10^3$), приводящий к временному ряду, характеризующему аномальную диффузию.

Шаг 3. Отображаем аддитивную составляющую наблюдаемой процессов Лоренц — Чен — аномальная диффузия $\{\hat{x}_n\}_{n=0}^N$ (рис. 5, а, см. четвертую сторону обложки) — на квадратную матрицу $N \times N$, $M = [N \times N]$ как

$$G_{i,j} = \{\hat{x}_n\}_{n=0}^N \Rightarrow M, G_{i,j} \in R^2. \quad (10)$$

Шаг 4. Получаем рекуррентные диаграммы PD (рис. 5, б, в, см. четвертую сторону обложки), используя выражение

$$\begin{aligned} PD_{i,j}^{m,\varepsilon} &= \theta(\varepsilon - \|x_i - x_j\|) \gamma(x_0) \cap \omega(x_0) \neq \emptyset, \\ x &\in R^m, i, j = 1, N, \end{aligned} \quad (11)$$

где N — число рассматриваемых состояний x ; ε — размер окрестности точек x в момент i ; $\|\bullet\|$ — норма; $\theta(\bullet)$ — функция Хевисайда; $\gamma(x_0) \cap \omega(x_0) \neq \emptyset$ — ограничение на сложность траектории (см. Определение 2).

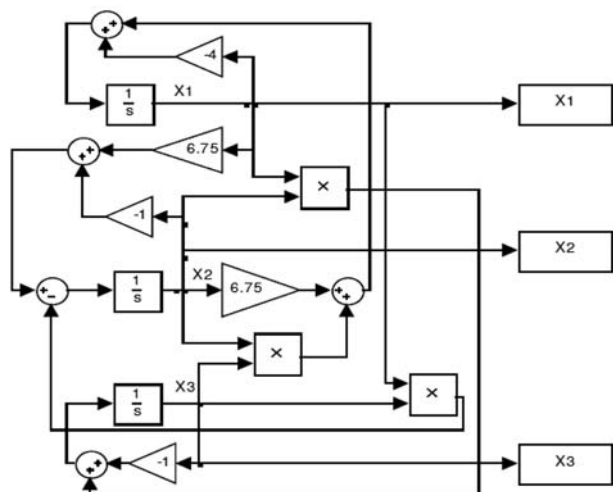
Алгоритм моделирования системы 2

Шаг 1. Проводим моделирование хаотической системы типа Рабинович — Фабрикант в программной среде MATLAB/Simulink [10, 11], $N = 2,5 \cdot 10^4$ точек (реально $2 \cdot 10^3$), представленной схемой (рис. 6, а) с аттрактором в плоскости $x_2 - x_3$ (рис. 6, б) при $a = 4$, $b = d = 1$, $h = 6,75$.

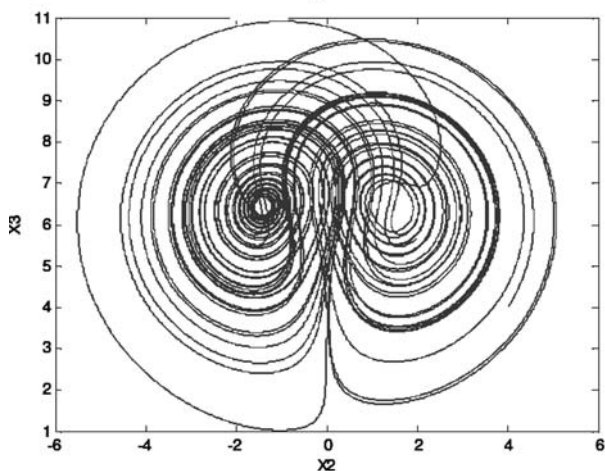
Шаг 2. Реализуем итерационный алгоритм процедуры (8) на множестве, состоящем из $N = 8 \cdot 10^3$ точек (реально $2 \cdot 10^3$), приводящей к временному ряду, характеризующему аномальную диффузию.

Шаг 3. Отображаем аддитивную составляющую наблюдаемой процессов Рабинович — Фабрикант — аномальная диффузия $\{\hat{y}_n\}_{n=0}^N$ (рис. 7, а, см. четвертую сторону обложки) на квадратную матрицу $N \times N$, $M = [N \times N]$:

$$G_{i,j} = \{\hat{y}_n\}_{n=0}^N \Rightarrow M, G_{i,j} \in R^2. \quad (12)$$



а)



б)

Рис. 6

Шаг 4. Получаем рекуррентные диаграммы (рис. 7, б, в, см. четвертую сторону обложки) [12], используя процедуру (11).

Таким образом, в результате реализованных алгоритмов оказалось, что визуальный анализ рекуррентных диаграмм (см. рис. 5, б, в) однозначно показал текстуру, характеризующую хаотический процесс (густое множество мелких точек), т. е. сформировалась новая хаотическая система Лоренц — Чен — аномальная диффузия.

Что касается рекуррентных диаграмм, представленных на рис. 7, б, в, то выяснилось, что интенсификация резонансного возбуждения стохастической составляющей (8) активизировало формирование квазирегулярной системы ($\lambda_1 = \lambda_2 = 0$) типа Рабинович — Фабрикант — аномальная диффузия.

Заключение

На основании проведенных исследований был сделан вывод, что резонансные возбуждения непосредственно не регулируют поведение взаимодействующих систем, а лишь формируют механизм их самоорганизации, т. е. формирование новых структур.

Список литературы

1. Лоскутов А. Ю. Проблемы нелинейной динамики II. Подавление хаоса и управление динамическими системами // Вестник МГУ им. М. В. Ломоносова. Сер. 3. Физика. Астрономия. 2001. № 3. С. 3—21.
2. Шарыпов О. В. Детерминированный хаос и случайность. URL: <http://filosof.historic.ru/books/item/f00/soo/z0000242T>.
3. Олемской А. И. Теория стохастических систем с сингулярным мультипликативным шумом // УФН. 1998. Т. 168, № 3. С. 287—321.
4. Eckmann Y. P., Oliffson S., Kamphorst O., Ruelle D. Recurrence plots of dynamical systems // Europhys. Lett. 1987. № 4. P. 973—977.
5. Владимирский Э. И., Исмаилов Б. И. Нелинейный рекуррентный анализ как математическая модель управления хаотическими процессами // Информационные технологии. 2011. № 5. С. 42—45.
6. Колесов А. Ю., Розов Н. Х. Турбулентный хаос // Современная математика и ее приложения. 2009. Т. 64. С. 39—53.
7. Учайкин В. В. Автомодельная аномальная диффузия и устойчивые законы // УФН. 2003. Т. 43, № 8. С. 847—876.
8. Pehlivan I., Uyaroglu Y. A new chaotic attractor from general Lorenz system family and electronic experimental implementation // J. Elec Eng & Comp. Sci. 2010. Vol. 18, N 2. P. 171—184.
9. Олемской А. И., Борисюк В. Н., Шудо И. А. Мультифрактальный анализ временных рядов // Вісник СумДУ. Сер. Фізика, математика, механіка. 2008. № 2. С. 70—81.
10. Emiroglu S. and Uyaroglu Y. Control of Rabinovich chaotic system based on a passive control // Scientific Research and Essays. 2010. Vol. 5 (21). P. 3298—3305.
11. Дьяконов В., Круглов В. Математические расширения MATLAB: Специальный справочник. СПб.: Питер, 2001. 480 с.
12. Владимирский Э. И., Исмаилов Б. И. Синхронизация в управлении хаотическими системами // Информационные технологии. 2012. № 1. С. 16—19.

А. Н. Четырбоцкий,
д-р физ.-мат. наук, вед. науч. сотр.,
Дальневосточный
геологический институт ДВО РАН,
Дальневосточный федеральный университет,
e-mail: Chetyrbotsky@yandex.ru

Приложение прямого метода решения системы линейных уравнений для численного моделирования сферического течения Куэтта несжимаемой жидкости в среде MATLAB

Рассматриваются вопросы организации вычислений составляющих скоростей сферического течения Куэтта, которое соответствует модельному представлению многообразных реальных процессов конвективного переноса. Приводится алгоритм вычислений и обоснование эффективности его реализации в рамках среды MATLAB. На основании серии вычислительных экспериментов показана работоспособность предлагаемого способа численного моделирования сферического течения Куэтта ньютоновской вязкой несжимаемой жидкости.

Ключевые слова: вязкая жидкость, сферическое течение Куэтта, методы решения системы линейных уравнений, разреженная матрица большой размерности, инструментарий графического представления результатов вычислительных экспериментов

Введение

Многообразные природные явления на Земле (отчасти и на планетах земной группы) в определенной степени связаны с вращением ее оболочек. В частности, во внешней оболочке вращение индуцирует крупномасштабную циркуляцию атмосферы [14] и последовательность нелинейно взаимосвязанных явлений планетарного масштаба (встречаются также случаи более мелких масштабов), динамическое взаимодействие между геосферами, мантийные конвективные течения и ряд других зависящих от вращения Земли процессов. В перечисленных здесь случаях движение субстанции (в случае атмосферы — масс воздуха, внутренних оболочек Земли — ее вещества) происходит между вращающимися коаксиальными или соосными сферами соответствующих радиусов. Простой моделью такого процесса служит сферическое течение Куэтта (СТК), где в качестве субстанции выступает некоторая обобщенная вязкая несжимаемая модельная жидкость. Допускается, что вращение граничных сфер происходит с разными угловыми скоростями вдоль их общей оси.

Изучению случаев СТК, когда в системе имеется малый параметр и можно упростить модельные урав-

нения, посвящено большое число работ [4, 15—17]. Представляется необходимым заметить отсутствие в большинстве из них сколь-нибудь конструктивных сведений о приемах увеличения эффективности используемого прикладного программного обеспечения. В связи с чем многочисленные приложения СТК обуславливают актуальность рассмотрения СТК с этой точки зрения. Подобное рассмотрение становится возможным только в рамках численных методов. Применяемые здесь методы основаны на так называемых прямых методах численного решения системы линейных уравнений (СЛАУ), при использовании которых точное решение уравнений находится за конечное число заранее определенных операций [1, 10].

Модель сферического течения Куэтта

Течение вязкой несжимаемой жидкости определяется из закона изменения импульса (баланс сил инерции, градиента давления в движущей среде и вязкости) в системе нелинейных нестационарных уравнений Навье — Стокса и условия несжимаемости. В тензорных обозначениях их запись принимает следующий вид:

$$\partial_t \mathbf{V} + (\mathbf{V} \cdot \nabla) \mathbf{V} = -\rho^{-1}(\nabla P + \nabla \cdot \mathbf{T}), \quad \nabla \cdot \mathbf{V} = 0, \quad (1)$$

где $\mathbf{V} = (V_r, V_\theta, V_\varphi)$ — вектор скорости течения (нижний индекс указывает на соответствующую ее составляющую — r — радиальную, θ — меридиональную и φ — азимутальную); t — текущее время; ρ — плотность жидкости; ∇P — градиент давления P (∇ — трехмерный оператор "набла" или дифференциальный оператор Гамильтона); $\nabla \cdot \mathbf{T}$ — дивергенция тензора напряжений \mathbf{T} . Здесь рассматривается случай осесимметричных (радиальная V_r и меридиональная V_θ скорости не зависят от азимута φ , что соответствует условиям обращения в нуль $\partial_\varphi = 0$ частных производных по этой переменной) движений ньютоновской несжимаемой вязкой жидкости в кольце $r_1 \leq r \leq r_2$ между сферами (рис. 1). Каждая из сфер вращается с собственной угловой скоростью (внутренняя сфера со скоростью ω_1 , а внешняя сфера с угловой скоростью ω_2). Независимость V от φ указывает на тот факт, что течение жидкости одинаково в любой меридиональной плоскости $\varphi = \text{const}$.

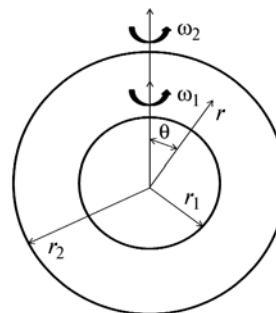


Рис. 1. Сферическое кольцо области течения

Согласно технике векторного исчисления для несжимаемой жидкости $(\mathbf{V} \cdot \nabla)\mathbf{V} = \nabla \times \mathbf{V} \times \mathbf{V}$ и $\nabla^2 V = -\nabla \times \nabla \times V$. Поскольку полагается случай $\rho = \text{const}$ и $\mu = \text{const}$, то в форме Громеки—Ламба векторное уравнение (1) и условие несжимаемости принимают следующий вид:

$$\begin{cases} \partial_t \mathbf{V} + \nabla \times \mathbf{V} \times \mathbf{V} = -\nabla \left(\frac{P}{\rho} \right) - \nu \nabla \times \nabla \times \mathbf{V}, \\ \frac{1}{r^2} \partial_r (r^2 V_r) + \frac{1}{r \sin \theta} \partial_\theta (\sin \theta V_\theta) = 0, \end{cases} \quad (2)$$

где "×" знак векторного произведения; $\nu = \mu/\rho$ — коэффициент кинематической вязкости; а второе уравнение есть запись условия несжимаемости модельной жидкости.

Коэффициенты Ламе в сферической системе координат не зависят от φ , что позволяет выполнить процедуру скаляризации, состоящей в представлении векторного поля \mathbf{V} посредством единственной скалярной функции — функции тока [12]. Процедура скаляризации выполняется векторными операторами, которые посредством введения этого объекта (функции тока) ψ проецируют векторное поле на скалярное. Здесь ψ вводится следующей записью [7]:

$$\mathbf{V} = \nabla \times (\psi \nabla \varphi),$$

следуя которому, составляющие векторного поля определяются выражениями

$$V_r = \frac{\Psi_\theta}{r^2 \sin \theta}, \quad V_\theta = -\frac{\Psi_r}{r \sin \theta}, \quad V_\varphi = \frac{\omega}{r \sin \theta}. \quad (3)$$

В выражениях (3) для простоты изложения принимается обозначение: нижний индекс у ψ указывает на частное дифференцирование по соответствующей переменной. Применение к обеим частям (2) оператора ротора (формально умножение слева обеих его частей на множитель $\nabla \times$) и последующее затем "обезразмеривание" приводит к следующим соотношениям:

$$\begin{cases} \partial_t \xi + P(\xi, \omega, \psi) = \frac{1}{\text{Re}} D^2 \xi, \\ \partial_t \omega - H(\omega, \xi) = \frac{1}{\text{Re}} D^2 \omega, \\ D^2 \psi = \xi, \\ P(\xi, \omega, \psi) \equiv \frac{2}{r^3 \sin \theta} [\omega G(\omega) + \xi G(\xi)] - H(\psi, \xi), \\ G(Y) \equiv r \cos \theta \partial_r Y - \partial_\theta Y, \\ H(a, b) \equiv \frac{1}{r^2 \sin \theta} (\partial_r a \partial_\theta b - \partial_\theta a \partial_r b), \\ D^2 \equiv \frac{\partial^2}{\partial r^2} + \frac{1}{r^2} \left(\frac{\partial^2}{\partial \theta^2} - \text{ctg} \theta \frac{\partial}{\partial \theta} \right). \end{cases} \quad (4)$$

Безразмерный комплекс Re (так называемое число Рейнольдса) определяется следующим выражением [8]:

$$\text{Re} = \frac{\omega_0 r_0^2}{\nu}, \quad (5)$$

где ω_0 и r_0 — зависимые от конкретной ситуации масштабы (характерные значения) угловой скорости и радиуса сферы.

В начальный момент времени $t = 0$ полагается

$$\psi = 0, \quad \xi = 0. \quad (6)$$

Граничные условия состоят в их сферичности, непроницаемости для жидкости, на границах выполняется также условие скольжения жидкости. При этом функция тока на оси вращения равняется нулю. Тогда запись граничных условий принимает вид

$$\begin{cases} \theta = 0 \text{ и } \theta = \pi: \psi = 0, \quad \xi = 0, \quad \omega = 0 \\ r = \frac{r_1}{r_0}: \psi = 0, \quad \psi_r = 0, \quad \xi = \frac{\Psi(\theta, r + \Delta r)}{\Delta r^2}, \quad \omega = \frac{\omega_1 r_1^2}{\omega_0 r_0^2} \sin^2 \theta, \\ r = \frac{r_2}{r_0}: \psi = 0, \quad \psi_r = 0, \quad \xi = \frac{\Psi(\theta, r - \Delta r)}{\Delta r^2}, \quad \omega = \frac{\omega_2 r_2^2}{\omega_0 r_0^2} \sin^2 \theta, \end{cases} \quad (7)$$

где Δr — шаг используемой вычислительной сетки для радиальной переменной. При записи (7) учитывается тот факт, что решение задачи рассматривается в полосе $\theta \in [0, \pi]$. Выбор такого диапазона изменения θ обусловлен спецификой СТК, а именно, симметрией течения объема модельной жидкости вокруг оси его вращения. Граничные условия для ξ следуют из 3-го уравнения соотношений (4) после подстановки в него нулевых граничных значений ψ и конечно-разностной записи оператора $\partial^2/\partial r^2$ в точках границы вычислительной области.

Уравнения (4) с начальными условиями (6) и граничными условиями (7), вследствие их нелинейной природы, решаются численным образом на основании приведенной ниже вычислительной процедуры.

Вычислительная процедура

На первом этапе вычислительной процедуры следует выполнить обезразмеривание переменных

$$t \omega_0 \rightarrow t, \quad \frac{r}{r_0} \rightarrow r, \quad \frac{\Psi}{\omega_0 r_0^2} \rightarrow \psi, \quad \frac{\omega}{\omega_0 r_0^2} \rightarrow \omega, \quad (8)$$

где стрелка характеризует способ преобразования переменной, каким образом это преобразование выполняется.

Для решения задачи (4) при начальном условии (7) и граничных условиях (8) область ее вычислений первоначально покрывается регулярной сеткой узлов $\{(r_i, \theta_j, t_k) : 1 \leq i \leq N_r, 1 \leq j \leq N_\theta, 1 \leq k \leq N_t\}$, где N_r, N_θ, N_t — заданное число узлов для каждой независимой переменной. Соответствующие переменные здесь масштабированы значениями r_1 и ω_1 , т. е. $r_0 \equiv r_1$ и $\omega_0 \equiv \omega_1$. Тогда

$$\Delta r = \frac{(r_2 - r_1)}{r_1 N_r}; \quad \Delta \theta = \frac{\pi}{N_\theta}; \quad \Delta t = \frac{T_1 \omega_1}{N_t}, \quad (9)$$

где T_1 — период вращения внутренней сферы.

В рассматриваемом случае для дискретизации дифференциальных уравнений используется пяти-точечный шаблон узлов указанной регулярной сетки. А сама процедура дискретизации выполняется таким образом, что:

1) линейные члены уравнений содержат весовой множитель, посредством которого можно регулировать степень неясности результирующей вычислительной схемы;

2) нелинейные члены с противоположным знаком составляют правую часть уравнений и в последующих вычислениях записываются в форме членов многошаговой линейной схемы (в данном случае это двухшаговая явная схема второго порядка Адамса—Бэшфорта [9]).

Согласно сказанному, запись дискретной аппроксимации уравнений на k -м временном слое принимает вид

$$\left\{ \begin{aligned} \xi_{ij}^{(k+1)} &= \xi_{ij}^{(k)} + \frac{\Delta t}{\text{Re}} \Lambda(\alpha_1 \xi_{ij}^{(k+1)} + \\ &+ (1 - \alpha_1) \xi_{ij}^{(k)}) - \hat{P}_{ij}^{(k)}, \\ \omega_{ij}^{(k+1)} &= \omega_{ij}^{(k)} + \frac{\Delta t}{\text{Re}} \Lambda(\alpha_2 \omega_{ij}^{(k+1)} + \\ &+ (1 - \alpha_2) \omega_{ij}^{(k)}) - \hat{H}_{ij}^{(k)}, \\ \Lambda \psi_{ij}^{(k+1)} &= \xi_{ij}^{(k+1)}, \\ \hat{P}_{ij}^{(k)} &\equiv \frac{\Delta t}{2} [3P(\xi_{ij}^{(k)}, \omega_{ij}^{(k)}, \psi_{ij}^{(k)}) - \\ &- P(\xi_{ij}^{(k-1)}, \omega_{ij}^{(k-1)}, \psi_{ij}^{(k-1)})], \\ \hat{H}_{ij}^{(k)} &\equiv \frac{\Delta t}{2} [3H(\omega_{ij}^{(k)}, \xi_{ij}^{(k)}) - H(\omega_{ij}^{(k-1)}, \xi_{ij}^{(k-1)})], \end{aligned} \right. \quad (10)$$

где линейный оператор Λ представляет конечно-разностную аппроксимацию второго порядка оператора D^2 из (4):

$$\Lambda(q_{ij}) \equiv \frac{1}{\Delta r^2} q_{i-1j} - 2 \left(\frac{1}{\Delta r^2} + \frac{1}{r_i^2 \Delta \theta^2} \right) q_{ij} + \frac{1}{\Delta r^2} q_{i+1j} + \\ + \frac{1}{r_i^2} \left(\frac{1}{\Delta \theta^2} + \frac{\text{ctg} \theta_j}{2 \Delta \theta} \right) q_{ij-1} + \frac{1}{r_i^2} \left(\frac{1}{\Delta \theta^2} - \frac{\text{ctg} \theta_j}{2 \Delta \theta} \right) q_{ij+1}. \quad (11)$$

Многообразие конечно-разностных методов решения дифференциальных уравнений состоит не столько в способах построения конечно-разностной аппроксимации уравнений, сколько в методах решения соответствующей этой аппроксимации системы линейных алгебраических уравнений. Выбор эффективного способа решения (10) здесь определяется ее структурой. Так, в простейшем случае на каждом временном слое для нахождения решений во всех внутренних узлах сетки ($2 \leq i \leq N_r - 1$, $2 \leq j \leq N_\theta - 1$) следует $2(N_r - 2)(N_\theta - 2)$ раза вычислять значения (11). Что представляется крайне неэффективным по причине значительных вычислительных затрат.

Для решения (10) предлагается следующий алгоритм. Сначала вводится матрица A размерности

$N \times N$, где $N = (N_r - 2)(N_\theta - 2)$, каждая строка которой содержит только пять ненулевых элементов. Каждому внутреннему узлу сетки (i, j) соответствует диагональный элемент матрицы с индексом (n, n) , где $n = (j - 2)(N_r - 2) + i - 1$. Значение диагонального элемента равно коэффициенту при q_{ij} в представлении (11), а справа и слева от диагонального элемента — число $1/\Delta r^2$; элемент $n - (N_r - 2)$ столбца данной строки есть коэффициент при q_{ij-1} , а $n + (N_r - 2)$ столбца — коэффициент при $q_{i,j+1}$. В этих обозначениях совокупность (11) записывается в виде N -мерного вектор-столбца $A \cdot Q$. Последовательность его элементов соответствует упорядоченным по столбцам элементам матрицы $\{q_{ij} : 2 \leq i \leq N_r - 1, 2 \leq j \leq N_\theta - 1\}$, т. е. (i, j) элемент этой матрицы есть $n = (j - 2)(N_r - 2) + i - 1$ элемент вектора

$$Q = (q_{22}, q_{32}, q_{42}, \dots, q_{NN}).$$

В рамках введенных понятий запись (10) принимает вид

$$\left\{ \begin{aligned} \left(E - \alpha \frac{\Delta t}{\text{Re}} A \right) \Xi^{(k+1)} &= \left[E + (1 - \alpha) \frac{\Delta t}{\text{Re}} A \right] \Xi^{(k)} - \hat{P}^{(k)}, \\ \left(E - \alpha \frac{\Delta t}{\text{Re}} A \right) \Omega^{(k+1)} &= \left[E + (1 - \alpha) \frac{\Delta t}{\text{Re}} A \right] \Omega^{(k)} - \hat{H}^{(k)}, \\ A \Psi^{(k+1)} &= \Xi^{(k+1)}, \end{aligned} \right. \quad (12)$$

где E — единичная матрица $N \times N$ размерности; $\Xi = (\xi_{22}, \xi_{32}, \xi_{42}, \dots, \xi_{NN})$ и $\Omega = (\omega_{22}, \omega_{32}, \omega_{42}, \dots, \omega_{NN})$ — N -мерные вектор-столбцы. В (12) и везде в дальнейшем полагается равенство $\alpha_1 = \alpha_2$.

Матрица $E - \alpha A$ относится к разреженному типу, где под этим термином понимается совокупность схемы хранения данных в сочетании с соответствующим алгоритмом для выполнения требуемой операции [2]. Структура этой матрицы и алгоритм работы с ней позволяют получить выигрыш по памяти и времени. Действительно, она насчитывает только $500/N$ % отличных от нуля элементов (например, при $N_r = 20$ и $N_\theta = 20$ только около 1,5 % элементов, отличных от нуля). И именно только их следует учитывать при различных линейных операциях с самой матрицей элементов (далее станет ясно, что в рассматриваемом случае изучения СТК используются только линейные операции с матрицами). Как раз такой способ работы автоматизирован в программной среде MATLAB. Все алгоритмы MATLAB для разреженных матриц, хранящихся в массивах типа **sparse array**, работают с учетом структуры матрицы, обеспечивая значительное ускорение вычислений по сравнению с полными матрицами [3]. В этом легко убедиться запуском профайлера MATLAB (программа для получения сведений о временных затратах на выполнение команд).

В рамках среды MATLAB решение (12) целесообразно провести следующим образом. Сначала выполняется LU -факторизация разреженной матрицы $E - \alpha A$, которая состоит в ее представлении как произведения нижней треугольной матрицы L

и верхней треугольной матрицы U (процедура LU [13]). В такой ситуации (12) принимает вид

$$\begin{cases} L \cdot U \cdot \Xi^{(k+1)} = F(\Xi^{(k)}, \hat{P}^{(k)}), \\ L \cdot U \cdot \Omega^{(k+1)} = F(\Omega^{(k)}, \hat{H}^{(k)}), \\ L \cdot U \cdot \Psi^{(k+1)} = \Xi^{(k+1)}, \\ F(X, Y) = \left[E + (1 - \alpha) \frac{\Delta t}{\text{Re}} A \right] X - Y. \end{cases} \quad (13)$$

Для СТК матрицы L и U также являются разреженными и поэтому в процедуре численного моделирования используются только их ненулевые элементы. Допустимость LU -факторизации матрицы $E - \alpha A$ обусловлена ее невырожденностью [10]. Понятно, что такое разложение при работе вычислительной процедуры достаточно выполнить только один раз, сразу же после формирования матрицы факторизации. Далее матрицы разложения используются на каждом временном шаге решения (12). Само же решение каждого уравнения (12) выполняется в два этапа. На первом этапе решается уравнение $L \cdot X = F$, где F — правая часть соответствующего уравнения. На втором этапе уже решается уравнение $U \cdot Y = X$, где Y — искомое решение. Такая процедура относится к прямым методам решения системы линейных уравнений [1, 13].

Вычислительные эксперименты

Основная цель экспериментов состояла в изучении влияния ω_2 на динамику функции тока, что представляет особый интерес в изучении геодинамики Земли. Для их проведения были использованы сведения о глубинной динамике мантии Земли [11]. В данном случае радиус внутренней сферы (радиус жидкого ядра Земли) $r_1 = 3000$ км, а радиус внешней сферы (радиус Земли) $r_2 = 6370$ км. Число Рейнольдса полагалось равным 10, а $\omega_1 = 3$. Параметр α из (12) полагается равным 0,9, а число временных циклов 1000. Для построения начального распределения ω внутри расчетной области использовалась удовлетворяющая граничным точкам линейная функция.

На рис. 2 представлены результаты экспериментов, выполненные для различных угловых скоростей внешней сферы ($\omega_2 = 1$, $\omega_2 = 1,5$ и $\omega_2 = 2$).

Для всех случаев вследствие независимости скорости от азимутального угла поверхности тока являются коаксиальными поверхностями вращения. Анализ этих распределений показывает следующее. При низких значениях ω_2 в каждом отдельном октанте области вычислений присутствует только одна структура вложенных друг в друга замкнутых линий тока.

По мере увеличения ω_2 в октанте появляется дополнительная структура, наличие которой обусловлено расслоением течения или его бифуркацией. В результате образуется вторичное течение, которое вследствие разных знаков у соответствующих

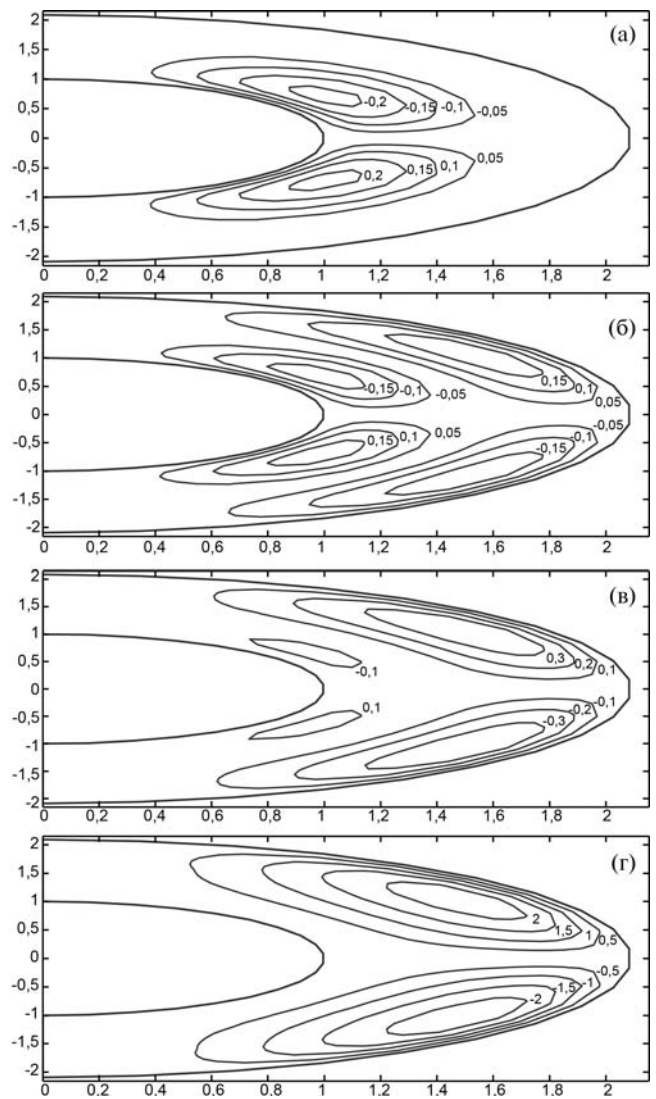


Рис. 2. Контур функции тока для различных значений угловой скорости внешней сферы:

$a - \omega_2 = 1$; $b - \omega_2 = 1,5$; $v - \omega_2 = 2$; $r - \omega_2 = 4,5$ (для удобства значения ψ умножены на 100)

кривых направлено противоположно основному течению. Представляется полезным отметить подобие между такой ситуацией и формированием экмановских слоев противотечений в океане и атмосфере [5, 6]. Появление этого течения становится возможным только при определенном сочетании сил внутреннего трения (вязкости), угловых скоростей и характера отношения инерционных сил к силам вязкости (число Рейнольдса).

Заключение

Рассмотрено течение во вращающемся с постоянной угловой скоростью сферическом кольце однородной несжимаемой вязкой жидкости. На границах области имеет место условие прилипания. Разработан алгоритм решения алгебраической системы уравнений, которая есть разностная аппроксимация уравнений движения со вторым порядком

точности. Программная реализация алгоритма выполнена в среде MATLAB. Результаты серии вычислительных экспериментов показывают, что даже в такой относительно простой системе модельных уравнений возможны нетривиальные эффекты дифференциального вращения. Данное обстоятельство представляется особенно полезным фактом настоящего рассмотрения, поскольку для изучения таких нетривиальных режимов течения обычно полагают неоднородность жидкости и к модельным уравнениям привлекают еще уравнение динамики температуры.

Список литературы

1. Андерсон Д., Таннехил Дж., Плетчер Р. Вычислительная гидромеханика и теплообмен. Т. 1. М.: Мир, 1990. 384 с.
2. Ануфриев И. Е., Корнеев В. Г. Численное тестирование метода декомпозиции для р-версии МКЭ // Известия вузов. Математика. 2005. № 512 (1). С. 10—24.
3. Ануфриев И. Самоучитель MATLAB 5.3/6.x. СПб.: БХВ-Петербург, 2002. 712 с.
4. Астафьева Н. М., Браиловская И. Ю., Яворская И. М. Нестационарное движение сжимаемой вязкой жидкости в шаровом слое // Изв. АН СССР. Механика жидкости и газа. 1972. № 3. С. 3—10.
5. Астафьева Н. М. Структуры, формирующиеся во вращающемся сферическом слое под влиянием условий, имити-

рующих глобальные потоки тепла в атмосфере // Современные проблемы дистанционного зондирования Земли из космоса. 2006. Т. 2. С. 345—356.

6. Гилл А. Динамика атмосферы и океана. Т. 2. М.: Мир, 1986. 415 с.
7. Джозеф Д. Устойчивость движения жидкости. М.: Мир, 1981. 638 с.
8. Ландау Л. Д., Лифшиц Е. М. Гидродинамика. М.: Наука, 1988. 736 с.
9. Оран Э., Борис Дж. Численное моделирование реагирующих потоков. М.: Мир, 1990. 660 с.
10. Флетчер К. Вычислительные методы в динамике жидкостей. Т. 1. М.: Мир, 1991. 502 с.
11. Сеидов Д. Г., Сорохтин О. Г., Сорохтин Ю. О. Численное моделирование мантийной конвекции // Докл. АН СССР. 1987. Т. 294, № 1. С. 58—63.
12. Ширяева С. О., Григорьев А. И. Скаляризация векторных краевых задач гидродинамики. Ярославль: ЯрГУ, 2010. 180 с.
13. Anderson E., Bai Z., Bischof C., Blackford S., Demmel J., Dongarra J., Du Croz J., Greenbaum A., Hammarling S., McKenney A., Sorensen D. LAPACK User's Guide. URL: http://www.netlib.org/lapack/lug/lapack_lug.html, Philadelphia: Third Edition, SIAM, 1999.
14. Munson B. R., Joseph D. D. Viscous incompressible flow between concentric spheres. Part 1. Basic flow // Journal of fluid mechanics. 1971. Vol. 49. P. 289—303.
15. Proudman I. The almost rigid rotation of viscous fluid between concentric spheres // Journal of fluid mechanics. 1956. Vol. 1. P. 505—516.
16. Stewartson K. On almost rigid rotations // Journal of fluid mechanics. 1957. Vol. 3. P. 17—26.

УДК 519.248:681.51

С. И. Елесина, канд. техн. наук, доц.,
e-mail: elesinas@mail.ru,

А. А. Логинов, канд. техн. наук, доц.,

М. Б. Никифоров, канд. техн. наук, доц.,
Рязанский государственный
радиотехнический университет

Методика оценки качества критерияльной функции совмещения изображений

Приводится методика оценки качества критерияльной функции, полученной при совмещении изображений в корреляционно-экстремальной навигационной системе. Рассматриваются корреляционные и разностные критерияльные функции. Сформировано множество показателей качества критерияльной функции. Проведено ранжирование показателей качества критерияльной функции по приоритетам и реализован многокритерияльный выбор наиболее подходящих функций.

Ключевые слова: критерияльная функция, глобальный экстремум, многокритерияльная оптимизация, целевая функция, корреляционно-экстремальная навигационная система, показатели качества

Корреляционно-экстремальные навигационные системы (КЭНС) базируются на автоматическом совмещении текущего изображения (ТИ) поверхности, полученного от радиолокационной станции (РЛС), и эталонного (ЭИ), хранящегося в базе данных бортовой ЭВМ. По итогам совмещения осуществляется привязка координат летательного аппарата. Задача усложняется требованиями реального времени и заданной высокой точностью совмещения. Вычислительная сложность совмещения в значительной степени зависит не только от алгоритма поиска, но и от выбранного функционала сравнения или типа критерияльной функции F , т. е. меры, с помощью которой будет выноситься решение о степени сходства сравниваемых изображений. В данной статье предлагается методика выбора наиболее рационального критерия.

1. Формирование исходного множества критерияльных функций

Можно выделить пять групп критерияльных функций: корреляционные, разностные, парные, ранговые и спектральные [1—4]. Рассмотрим некоторые из них, наиболее пригодные для совмещения радиолокационных изображений [5, 6].

1.1. Корреляционная критериальная функция

Под классическим алгоритмом корреляционной обработки изображений понимается вычисление функции взаимной корреляции или интеграла типа свертки с последующим поиском максимума этой функции:

$$F(x, y) = \frac{1}{MN} \sum_{i=0}^{N-1M-1} \sum_{j=0}^{M-1} f_{\text{ЭИ}}(i, j) f_{\text{ТИ}}(i+x, j+y), \quad (1)$$

где $0 \leq x < (K-N)$, $0 \leq y < (L-M)$, $K \times L$ — размер ТИ, $N \times M$ — размер ЭИ; x, y — смещение фрагмента изображения $f_{\text{ЭИ}}$ относительно $f_{\text{ТИ}}$. Считается, что x и y отсчитываются от левого верхнего угла $f_{\text{ТИ}}$, для которого $x = 0, y = 0$.

Кроме того, в качестве меры сходства двух фрагментов изображений можно использовать коэффициент корреляции:

$$\rho(x, y) = \frac{1}{\sigma(f_{\text{ЭИ}})\sigma(f_{\text{ТИ}})} \times \left[\frac{1}{MN} \sum_{i=0}^{N-1M-1} \sum_{j=0}^{M-1} f_{\text{ЭИ}}(i, j) f_{\text{ТИ}}(i+x, j+y) - \bar{f}_{\text{ЭИ}} \bar{f}_{\text{ТИ}} \right], \quad (2)$$

где $\sigma(f_{\text{ЭИ}})$, $\sigma(f_{\text{ТИ}})$ — среднеквадратические отклонения яркости фрагментов $f_{\text{ЭИ}}$ и $f_{\text{ТИ}}$; $\bar{f}_{\text{ЭИ}}$, $\bar{f}_{\text{ТИ}}$ — средние значения отсчетов яркости изображений.

Обычно корреляционную критериальную функцию нормируют для того, чтобы число ложных максимумов, возникающих при различии уровней интенсивности изображений, было бы по возможности минимальным. Кроме этого, по значению нормированной корреляционной функции в найденном глобальном экстремуме (ГЭ) можно судить о том, является он истинным или ложным. В работе [2] показано, что использование площади нормированного корреляционного пика повышает устойчивость алгоритма. Тогда на основании [2] выражение (1) можно представить в виде

$$F^2(x, y) = \frac{\left[\sum_{i=0}^{N-1M-1} \sum_{j=0}^{M-1} f_{\text{ЭИ}}(i, j) f_{\text{ТИ}}(i+x, j+y) \right]^2}{\left[\sum_{i=0}^{N-1M-1} \sum_{j=0}^{M-1} f_{\text{ЭИ}}^2(i, j) \right] \times \left[\sum_{i=0}^{N-1M-1} \sum_{j=0}^{M-1} f_{\text{ТИ}}^2(i+x, j+y) \right]} \quad (3)$$

или

$$F(x, y) = \frac{\sum_{i=0}^{N-1M-1} \sum_{j=0}^{M-1} f_{\text{ЭИ}}(i, j) f_{\text{ТИ}}(i+x, j+y)}{\sqrt{\left[\sum_{i=0}^{N-1M-1} \sum_{j=0}^{M-1} f_{\text{ЭИ}}^2(i, j) \right] \times \left[\sum_{i=0}^{N-1M-1} \sum_{j=0}^{M-1} f_{\text{ТИ}}^2(i+x, j+y) \right]}} \quad (4)$$

1.2. Разностные критериальные функции

Разностные критериальные функции основаны на поэлементном вычислении разностей интенсивности ТИ и ЭИ.

Общее выражение $F(x, y)$ для разностных алгоритмов имеет вид

$$F(x, y) = \sum_{i=0}^{N-1M-1} \sum_{j=0}^{M-1} |f_{\text{ЭИ}}(i, j) - f_{\text{ТИ}}(i+x, j+y)|^n, \quad (5)$$

где $n = 0, 1, 2, \dots$. При увеличении параметра n точностные характеристики функции улучшаются, однако при $n \geq 3$ это улучшение становится незначительным.

В КЭНС распространение получили следующие разностные критериальные функции:

функция среднего модуля разности

$$F_1(x, y) = \frac{1}{MN} \sum_{i=0}^{N-1M-1} \sum_{j=0}^{M-1} |f_{\text{ЭИ}}(i, j) - f_{\text{ТИ}}(i+x, j+y)|; \quad (6)$$

функция среднего квадрата разности

$$F_2(x, y) = \frac{1}{MN} \sum_{i=0}^{N-1M-1} \sum_{j=0}^{M-1} |f_{\text{ЭИ}}(i, j) - f_{\text{ТИ}}(i+x, j+y)|^2. \quad (7)$$

Разностные критериальные функции равны нулю при полном совмещении одинаковых изображений и имеют большие значения при неточном совмещении изображений. Таким образом, в точке наилучшего совмещения функция $F(x, y)$ достигает минимума, поэтому целесообразно, если ищется максимум, данную функцию брать с отрицательным знаком. Кроме того, различие в средних уровнях яркости радиолокационных изображений может вызвать ложные экстремумы. В этом случае используют центрированную разностную функцию

$$F(x, y) = \frac{1}{MN} \sum_{i=0}^{N-1M-1} \sum_{j=0}^{M-1} |f_{\text{ЭИ}}(i, j) - \bar{f}_{\text{ЭИ}} - f_{\text{ТИ}}(i+x, j+y) + \bar{f}_{\text{ТИ}}|, \quad (8)$$

где $\bar{f}_{\text{ЭИ}}$, $\bar{f}_{\text{ТИ}}$ — средние яркости РЛИ.

Преимуществом разностных критериальных функций перед корреляционными является то, что в них отсутствует операция умножения. Это позволяет снизить вычислительные затраты в 4...10 раз. В то же время, при малых отношениях сигнал/шум характеристики разностных функций уступают корреляционным [1].

2. Формирование множества показателей качества критериальной функции

Чтобы сравнить данные функции, необходимо определить множество показателей качества целевой функции (ЦФ) $Q_F = \{g_{F1}, g_{F2}, \dots, g_{FN}\}$. К показателям качества можно отнести следующие характеристики функции: трудоемкость, число экстремумов, расстояние между ГЭ и ближайшим по значению и по расстоянию локальным экстремумом (ЛЭ), среднее расстояние между ближайшими ЛЭ, среднее значение ЦФ, минимальное и максимальное значения функции, среднее квадратическое отклонение значений (СКО) функций, СКО экстремумов функции, СКО расстояний между экстремумами и т. д.

Сравнивать корреляционные и разностные функции в абсолютных величинах нельзя, потому что они имеют разный диапазон изменения своих значений. Поэтому в дальнейших исследованиях для сравнения показателей качества g_{Fi} различных критериальных функций будем использовать приведение их к относительной величине.

3. Проведение многокритериальной оптимизации для отбора лучших критериальных функций

Оно может быть представлено в виде процедуры, состоящей из четырех этапов.

3.1. Разбиение показателей качества критериальных функций на группы по приоритету

В множестве $Q_F = \{g_{F1}, g_{F2}, \dots, g_{FN}\}$ не все показатели качества равнозначны, поэтому априорно они делятся на группы, каждой из которых назначается свой приоритет. Первый (наивысший) приоритет:

- трудоемкость вычисления ЦФ T ;
 - число экстремумов K ;
 - постоянная Липшица L .
- Второй приоритет:
- СКО значений функции $\sigma_{ЦФ}$;
 - СКО экстремумов функции σ_{F^*} ;
 - среднее расстояние между ближайшими экстремумами \bar{d}_{mean} ;
- Третий (низший) приоритет:
- СКО расстояний между экстремумами функции σ_d ;
 - минимальное расстояние между экстремумами d_{min} .

Трудоемкость ЦФ определяется числом операций сложения (вычитания) и умножения (деления):

$$T = nt_{add} + mt_{mul}$$

где t_{add} — трудоемкость операции сложения; t_{mul} — трудоемкость операции умножения; n и m — число операций сложения и умножения, которое зависит от размера ЭИ и вида критериальной функции.

Число экстремумов K определяется с помощью матрицы значений ЦФ $\|F_{ij}\|$, размерность которой

равна размеру области поиска ГЭ $M \times N$. В результате получаем массив экстремумов $F^*(r)$, $r = \overline{1, K}$, каждый элемент которого содержит значение ЦФ в экстремуме $f^*(x, y)$, где (x, y) — координаты экстремума в матрице $\|F_{ij}\|$, K — число элементов в массиве. Отсортировав массив по убыванию значений ЦФ, можно определить другие характеристики целевой функции, такие как максимальное значение функции, которое совпадает со значением ЦФ в ГЭ: $f_{max} = \max_{1 \leq r \leq K} F^*(r)$. Минимальное значение

функции: $f_{min} = \min_{\substack{1 \leq i \leq M \\ 1 \leq j \leq M}} F(i, j)$. Разность между

максимальным и минимальным значением ЦФ: $\Delta f = |f_{max} - f_{min}|$. Относительная разность между значениями ГЭ и ближайшим по значению ЛЭ: $\delta f = \frac{F^*(1) - F^*(2)}{f_{max} - f_{min}}$.

Для оценки постоянной Липшица будем использовать простейшую гармоническую модель критериальной функции, приближенно представляющей ее поведение на произвольном вертикальном срезе относительно плоскости xOy :

$$f(x) = Ax \sin(\omega x), \quad (9)$$

где $\omega = \frac{2\pi}{l_{min}}$; l_{min} — расстояние между ГЭ и ЛЭ:

$l_{min} = \min\{d_{mean}, d_{gmin}\}$, где d_{mean} — среднее расстояние между локальными экстремумами; d_{gmin} — минимальное расстояние между ГЭ и ЛЭ; $\Delta f = \delta f_{gmin}$ — относительная разность между значениями ГЭ и ближайшим по координатам ЛЭ.

Считая, что $x_1 \approx \frac{\pi}{2} \omega$, а $x_2 \approx \frac{5\pi}{2} \omega$ из формулы (9)

$$\text{определим значение } A = \frac{2\Delta f}{\pi \left(5 \sin\left(\frac{5\pi^2}{l_{min}^2}\right) - \sin\left(\frac{\pi^2}{l_{min}^2}\right) \right)}$$

Определяем производную функции (9):

$$f'(x) = A \sin\left(\frac{2\pi}{l_{min}} x\right) + A \frac{2\pi}{l_{min}} x \cos\left(\frac{2\pi}{l_{min}} x\right).$$

Постоянная Липшица $L = \max_{l_{min}} f'(x)$.

Для определения среднего расстояния между локальными экстремумами d_{mean} необходимо получить матрицу расстояний $\|d_{ij}\|$ где d_{ij} — расстояние между i -м и j -м ЛЭ:

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2};$$

(x_i, y_i) и (x_j, y_j) — координаты i -го и j -го ЛЭ.

Полученная матрица симметрична относительно главной диагонали. Строки и столбцы в матрице упорядочены по убыванию значений ЦФ в локаль-

ных экстремумах. Следовательно, первая строка матрицы содержит значения расстояний между ГЭ и всеми ЛЭ.

Среднее расстояние между локальными экстремумами d_{mean} :

$$d_{mean} = \frac{\sum_{i=1}^K \sum_{j=1}^K d_{ij}}{(K-1)^2}.$$

Минимальное расстояние между ГЭ и ЛЭ:

$$d_{glmin} = \min_{2 \leq i \leq K} d_{1i}.$$

В результате определяем i^* — номер ЛЭ в упорядоченном по значениям ЦФ массиве F^* . Значение ЛЭ ближайшего по координатам к ГЭ:

$$f_{glmin} = F^*(i^*).$$

Относительная разность между значениями ГЭ и ближайшим по координатам ЛЭ:

$$\delta f_{glmin} = \frac{|f_{\max}^* - f_{glmin}|}{|f_{\max} - f_{\min}|}.$$

СКО значений ЦФ:

$$\sigma_{\text{ЦФ}} = \sqrt{\frac{\sum_{i=1}^N \sum_{j=1}^M (\bar{F}_{ij} - \bar{F}_{mean})^2}{N \times M}},$$

где $\bar{F}_{ij} = \frac{F_{ij}}{\Delta f}$ — относительное значение ЦФ;

$\bar{F}_{mean} = \frac{F_{mean}}{\Delta f}$ — относительное среднее значение

ЦФ; $F_{mean} = \frac{\sum_{i=1}^N \sum_{j=1}^M F_{ij}}{N \times M}$ — среднее значение ЦФ.

СКО экстремумов ЦФ:

$$\sigma_{F^*} = \sqrt{\frac{\sum_{i=1}^K (\bar{F}_i^* - \bar{F}_{mean}^*)^2}{K}},$$

где $\bar{F}_i^* = \frac{F_i^*}{\Delta f_{extr}}$ — относительное значение экстремума ЦФ;

$\Delta f_{extr} = |f_{\max}^* - f_{\min}^*|$ — разность между максимальным и минимальным значением экстремума ЦФ;

$\bar{F}_{mean}^* = \frac{F_{mean}^*}{\Delta f_{extr}}$ — относительное среднее значение экстремума ЦФ;

$F_{mean}^* = \frac{\sum_{i=1}^K F_i^*}{K}$ — среднее значение экстремума ЦФ.

Для определения *среднего расстояния между ближайшими экстремумами* \tilde{d}_{mean} необходимо получить матрицу расстояний $\|\tilde{d}_{ij}\|$, где все элементы каждой строки матрицы упорядочены по возрастанию значений расстояний между экстремумами. Таким образом, среднее расстояние между i -м экстремумом и l ближайшими экстремумами определяется как

$$\tilde{d}_{mean_i}(l) = \frac{\sum_{j=2}^{l+1} d_{ij}}{l},$$

где $i = \overline{1, K}$, причем $j = \overline{2, l+1}$, так как $\tilde{d}_{i1} = 0$. Таким образом, *среднее расстояние между ближай-*

шими экстремумами: $\tilde{d}_{mean}(l) = \frac{\sum_{i=1}^K \tilde{d}_{mean_i}}{K}$.

СКО расстояний между экстремумами:

$$\sigma_d = \sqrt{\frac{\sum_{i=1}^K \sum_{j=1}^K (d_{ij} - d_{mean})^2}{K-1}}.$$

Минимальное расстояние между экстремумами:

$$d_{\min} = \min_{\substack{1 \leq i \leq K \\ 1 \leq j \leq K}} d_{ij}.$$

3.2. Проверка адекватности выбора и приоритетной группировки критериев критериальной функции с помощью оценки влияния степени предобработки изображения

Для проверки адекватности предложенных критериев качества ЦФ и разбиения их на группы по приоритетам проведено сравнение показателей ЦФ в зависимости от степени предварительной обработки ТИ. Рассмотрены исходное радиолокационное изображение (РЛИ) и ТИ на различных этапах предобработки: отфильтрованное РЛИ с помощью медианного фильтра (вид маски — крест, размер — 7 пиксел) и РЛИ с кластеризацией протяженных объектов.

Исследования выявили, что показатели качества первого приоритета (число экстремумов ЦФ K и постоянная Липшица L) за счет предобработки значительно улучшаются (в 5...7 раз). Показатели качества второго приоритета (СКО значений функции $\sigma_{\text{ЦФ}}$, СКО экстремумов функции σ_{F^*} , среднее расстояние между ближайшими экстремумами d_{mean}) за счет предобработки увеличивают свое значение на (15...50) %. Это происходит вследствие того, что фильтрация шума и выделения сущностей в исходном ТИ приводит к увеличению степени разброса значений критериальной функции относительно среднего значения. Показатели ка-

чества третьего приоритета изменяются незначительно (порядка 10 %).

Таким образом, исследование влияния степени предобработки ТИ на показатели качества ЦФ подтвердили правильность выбора предложенных показателей и их классификацию по приоритетам.

Лучшей будет та функция, которая имеет наименьшие показатели, относящиеся к первому приоритету. Чем больше показатели второго и третьего приоритета, тем лучше критериальная функция, так как большой разброс значений ЦФ и большие расстояния между координатами экстремумов облегчают поиск истинного положения ГЭ.

3.3. Первичный отбор критериальной функции по критериям первого приоритета

На первом этапе из дальнейшего рассмотрения была исключена функция взаимной корреляции (1), так как на исследуемых типах изображений даже при полном переборе не обеспечивается гарантированный поиск ГЭ.

Далее оставшиеся шесть функций рассматривались по показателям качества функции первого приоритета. Результаты сравнительной оценки представлены на рис. 1. На рисунке используются следующие обозначения: 1 — функция коэффициента корреляции; 2 — квадрат нормированной корреляционной функции; 3 — нормированная корреляционная функция; 4 — функция среднего модуля разности; 5 — функция среднего квадрата разности; 6 — центрированная разностная функция. По трудоемкости наилучшее значение, естественно, имеют

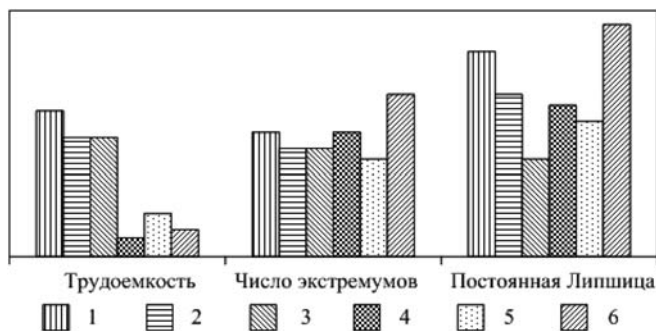


Рис. 1. Сравнение показателей качества первого приоритета критериальной функции

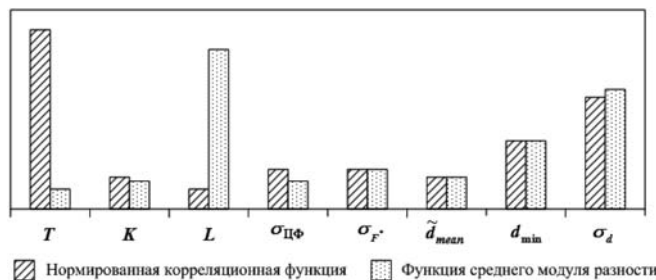


Рис. 2. Сравнение показателей качества нормированной корреляционной функции и функции среднего модуля разности

разностные функции и лучшая из них — функция среднего модуля разности (6). По числу экстремумов лучшие показатели у нормированных корреляционных функций (3) и (4) и у разностных (6) и (7). По постоянной Липшица в большинстве случаев лидирует нормированная корреляционная функция (4).

3.4. Отбор по критериям второго и третьего приоритетов

Таким образом, для сравнения ЦФ по показателям качества второго и третьего приоритетов выбираем две функции: нормированную корреляционную функцию (4) и функцию среднего модуля разности (6).

По показателям качества второго и третьего приоритетов явного лидера не выявлено. Лишь при больших размерах ЭИ (50 % от ТИ) небольшое преимущество по большинству показателей имеет нормированная корреляционная функция (4). Сравнение показателей для обеих критериальных функций представлено на рис. 2.

Ранжирование показателей качества ЦФ по приоритетам позволило реализовать многокритериальный выбор наиболее подходящих функций. Функция среднего модуля разности имеет самую низкую трудоемкость, и это в условиях реального времени, несомненно, очень важный показатель. Однако нормированная корреляционная функция тоже имеет свои преимущества. Данная функция принимает значения в диапазоне от 0 до 1, поэтому по найденному значению ГЭ можно судить о близости найденного значения к истинному ГЭ. Такие показатели ЦФ, как число экстремумов и постоянная Липшица, у нормированной корреляционной функции лучше, чем у разностной. Это приводит к уменьшению трудоемкости метода поиска ГЭ, и алгоритм поиска будет сходиться быстрее.

В результате применения данной методики были определены наиболее рациональные критериальные функции F , т. е. меры, с помощью которой можно выносить решения о степени сходства сравниваемых радиолокационных изображений в КЭНС.

Список литературы

1. Баблицкий В. К., Бочкарев А. М., Мусьяков М. П. Методы фильтрации сигналов в корреляционно-экстремальных системах навигации. М.: Радио и связь, 1986. 216 с.
2. Андреев Г. А., Потапов А. А. Алгоритмы обработки навигационной пространственно-временной информации. Часть I // Зарубежная радиоэлектроника. 1989. № 3. С. 3—18.
3. Андреев Г. А., Потапов А. А. Алгоритмы обработки навигационной пространственно-временной информации. Часть II // Зарубежная радиоэлектроника. 1989. № 4. С. 3—21.
4. Ким Н. В. Обработка и анализ изображений в системах технического зрения: учеб. пособие. М.: МАИ, 2001. 164 с.
5. Елесина С. И., Никифоров М. Б. Повышение эффективности генетического алгоритма // Информационные технологии. № 3, 2012. С. 49—55.
6. Бабаев С. И., Елесина С. И., Костров Б. В. Сравнение модифицированного поискового и генетического алгоритмов нахождения глобального экстремума в системах навигации // Вопросы радиоэлектроники. Сер. ОТ. 2010, июль. Вып. 1. С. 145—152.

УДК 004.383.4

Г. С. Речистов, мл. науч. сотр.,
инженер по программному обеспечению,
Лаборатория суперкомпьютерных технологий
для биомедицины, фармакологии
и малоразмерных структур,
факультет радиотехники и кибернетики,
Московский физико-технический институт,
ЗАО "Интел А/О",
e-mail: grigory.rechistov@phystech.edu

Использование полноплатформенного имитационного моделирования суперкомпьютерной системы для определения производительности счетных приложений

Предлагается расширение двух существующих моделей описания производительности многопроцессорных систем в целях их применения к кластерным суперкомпьютерам, состоящим из нескольких многопроцессорных узлов. На основе этих моделей предлагается способ предсказания производительности и поведения научных приложений на будущих аппаратных вычислительных платформах с использованием компьютерной симуляции для получения исходных данных.

Ключевые слова: моделирование, суперкомпьютеры, *Simics*, анализ CPI, анализ узких мест

Введение

Одним из основных принципов наращивания доступной производительности вычислений для научных и коммерческих задач является их параллельное исполнение. Этот принцип проявляется на всех уровнях организации ЭВМ, начиная от микроархитектуры в виде суперскалярных процессоров, многоядерных узлов и заканчивая объединением в сеть отдельных компьютеров для организации кластера. К сожалению, простое увеличение числа элементов (АЛУ, ядер, машин) практически никогда не приводит к пропорциональному росту полной вычислительной производительности. Причин этому много: необходимость синхронизации удаленных процессов; обращение к ограниченным общим ре-

сурсам и др. Предсказание поведения приложений на будущих, еще не построенных системах, представляется крайне важным, так как позволяет заранее обнаружить слабые места проекта и перераспределить средства для получения оптимальной системы. Для этого используется компьютерное моделирование (полноплатформенная симуляция) больших систем на доступных кластерах меньшего масштаба.

В работе описывается расширение двух существующих методик описания производительности, изначально разработанных для многопроцессорных систем, на системы типа "кластер" (группа идентичных ЭВМ, объединенных в сеть) для научных приложений, использующих библиотеку *MPI* [1]. Затем описывается решение задачи моделирования с помощью *Simics* [2], позволяющего сбалансировать точность и скорость исследования. Далее приводятся результаты практического применения методики для анализа существующего проекта серии суперкомпьютеров, развиваемых в Московском физико-техническом институте.

Модель

Описание характеристик моделируемого кластера дано в таблице. Для построения и исследования модели был задействован программный симулятор *Wind River Simics*. Это приложение предоставляет ряд уникальных инструментов, являвшихся жизненно необходимыми для нашего исследования: параллельное исполнение модели на всех процессорах системы; распределение модели на все узлы кластера; скорость исполнения моделируемого приложения, близкая к его скорости на реальной

Характеристики реальной и моделируемой вычислительных систем

Характеристика	Реальная система	Моделируемая система
Процессор	Intel Xeon E5680 (Westmere) 3,3 ГГц	Intel Xeon E5 (Sandy Bridge) 2,8 ГГц
Число ядер в процессоре	6	8
Число процессоров в узле	2	2
Число узлов	16	112
Объем ОЗУ на узел, Гбайт	24	48
Сеть	Infiniband QDR 10 Гбит/с	Infiniband QDR 10 Гбит/с
Общее число ядер	192	1792
Общий объем ОЗУ, Гбайт	384	5376

системе; изучение влияния немодифицированной операционной системы.

К сожалению, включение в модель всех деталей работы ядер микропроцессора, устройств оперативной памяти, сетевых маршрутизаторов привело бы к существенному замедлению ее работы и невозможности проведения сколько-нибудь существенного объема исследований в разумное время. Поэтому нами был использован комбинированный подход, в котором моделирование используется для получения значений ключевых параметров, характеризующих работу приложения внутри симулируемой системы. Эти данные затем используются в аналитических формулах описания производительности многопроцессорных ЭВМ, адаптированных для кластерных систем.

Два способа описания производительности приложения

Существует большое число работ, посвященных аналитическому описанию производительности микропроцессоров и многоядерных систем. В них используются два подхода: приведение производительности к усредненному времени исполнения одной инструкции [3–5] и анализ узких мест в системах передачи данных [6]. Ниже мы приводим адаптацию обоих подходов для кластерных систем.

Анализ среднего времени выполнения одной инструкции. Ключевой метрикой производительности является величина CPI (*cycles per instruction*), показывающая, сколько в среднем тактов процессора необходимо для исполнения одной инструкции. При анализе CPI разбивается на сумму факторов, независимо влияющих на производительность [3, 7]:

$$CPI = CPI_{core} + CPI_{caches} + CPI_{memory} + CPI_{MPI}$$

где CPI_{core} — число тактов, затрачиваемых на исполнение непосредственно в ядре процессора; CPI_{caches} — задержки при обращениях в кэш-память; CPI_{memory} — задержки при доступе в оперативную память; CPI_{MPI} — коммуникации с помощью интерфейса MPI . Все члены, кроме CPI_{core} , равны математическому ожиданию некоторого класса событий, прерывающих вычисления в ядре. Так,

$$CPI_{caches} = \sum P\{i\}L_i$$

где $P\{i\}$ — вероятность i -го типа промаха кэшей (при чтении, записи, для различных уровней кэша); L_i — задержка в тактах для данного события. По сравнению с оригинальной моделью, разработанной для одноядерной системы, описанный подход учитывает эффекты от взаимодействия нескольких ядер через совместное использование памяти и кэшей. Кроме того, новый член — CPI_{MPI} учитывает взаимодействие, специфичное для распределенных приложений, использующих библиотеку MPI .

Следующий шаг методики — это учет того факта, что CPI характеризует быстроту исполнения инст-

рукций безотносительно их природы: в эту величину включаются операции загрузки данных, условных переходов и т. п., непосредственно не связанные с вычислительным алгоритмом приложения. В конечном счете, нас интересует скорость исполнения операций с числами с плавающей запятой — $FLOPS$ (*floating operations per second*). Не все исполненные инструкции выполняют собственно вычисления, кроме того, современные векторные инструкции содержат операции над двумя (в случае набора SSE) или четырьмя (для набора AVX) парами операндов. Для получения $FLOPS$ необходимо знать еще одну характеристику приложения — коэффициент α , показывающий, сколько операций с плавающей запятой в среднем приходится на одну инструкцию. Значение $FLOPS$ получается из следующего соотношения:

$$FLOPS^{-1} = CPI_{core}/\alpha + CPI_{caches} + CPI_{memory} + CPI_{MPI}$$

Анализ узких мест. Эта методика исходит из наблюдения, что производительность систем определяется "узким местом" — одной из подсистем передачи данных с наибольшей утилизацией при исполнении конкретной задачи. Для различных задач и конфигураций аппаратуры лимитирующими факторами будут разные подсистемы. Ниже изложена адаптация методики нахождения пиковой производительности вычислений с плавающей запятой ($FLOPS$) [5].

Мы выделяем и анализируем три возможные точки образования "узкого места": кэш-память, оперативную память и сеть передачи сообщений между узлами. Для каждой из них введем характеристику c — удельную интенсивность операций, равную среднему числу операций над числами с плавающей запятой, отнесенному к одному переданному байту. Ее значение будет различно для кэша, памяти и сети,

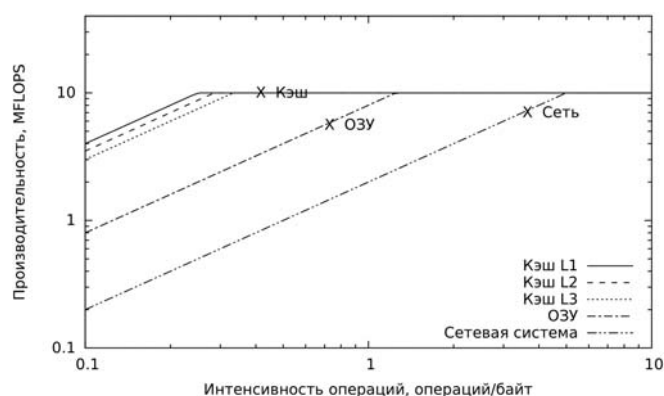


Рис. 1. Зависимости производительности приложения от интенсивности использования для кэш-памяти, оперативной памяти и сети. Наклон графиков в их левой части определяется пропускной способностью соответствующего канала; общая "крыша" графика определена максимальной производительностью ядра. Итоговая скорость системы характеризуется наименьшей из трех величин (в данном примере — производительностью ОЗУ)

например, в ОЗУ попадают только те запросы данных, которые не были удовлетворены системой кэшей. Вторая характеристика — это пропускная способность каждой подсистемы X , измеряемая в байт/с. Последняя величина — это пиковая производительность вычислительного ядра T в случае, если ни одна из подсистем ее не ограничивает. Производительность приложения при условии, что узким местом является подсистема i , определяется из соотношения

$$FLOPS_i = \min(cX, T).$$

Если в системе несколько потенциальных узких мест, то необходимо выбрать минимальное из них (рис. 1)

$$FLOPS = \min(FLOPS_1, FLOPS_2, FLOPS_3).$$

Описание эксперимента

Выше в таблице приведены характеристики системы, использовавшейся для проведения исследований. Как видно из сравнения систем в таблице, мы имели примерно в 9 раз меньше физических вычислительных ядер по сравнению с числом моделируемых. Тем не менее скорость моделирования оказалась достаточной для проведения серии экспериментов (наблюдаемое замедление скорости загрузки операционной системы было 10...40 раз). Были исследованы конфигурации, включающие в себя от 16 до 112 узлов модели. В качестве изучаемых приложений были использованы пакеты молекулярной динамики Gromacs [8] и Amber [9], компилируемые с помощью GCC 4.4.5. В качестве библиотеки MPI использовалась MPICH2-1.4.pl.

Источники численных значений, входящих в формулы

Для получения модельных значений CPI_{caches} , CPI_{memory} , CPI_{MPI} в симуляторе были использованы следующие механизмы: подключаемая трехуровневая иерархия кэшей; модель неоднородной (*non uniform memory access*, NUMA) памяти; инструментация приложения для обнаружения исполнения и учета задержек всех функций MPI. Значение CPI_{core} затруднительно получить без использования медленной потактовой симуляции, поэтому его брали из экспериментов на реальной аппаратуре [10]. В этом эксперименте также определялось значение коэффициента α .

Значение T ("потолок" производительности) находим из результатов запуска программы High Performance Unpack [11]. Значения пропускных способностей X берутся из тестов: для ОЗУ — STREAM [12]; для кэшей — LMBench [13]; для сетевой подсистемы — Netperfmeter [14]. Значения c находим из экспериментов на модели путем подсчета числа соответствующих событий передачи данных.

Результаты

Ввиду ограничений объема ниже приведены только результаты экспериментов по определению характеристик системы памяти и анализа хода исполнения изучаемых приложений.

Часть результатов была получена и описана в предыдущих работах. Порядок нахождения и результаты измерения CPI_{core} описаны в работе [7]. Результаты измерения производительности сети и пиковой вычислительной производительности High Performance Linpack, полученные для изучаемой модели и для используемой физической системы, даны в работе [10].

На рис. 2 показаны результаты измерения пропускной способности иерархии кэшей и оперативной памяти, выполненные на реальной аппаратуре и используемые затем в методе анализа узких мест.

На рис. 3 приведен результат одного из экспериментов по определению средней задержки передачи MPI сообщения L_p , используемый в подходе, связанном с анализом длительности средней инструкции. На рис. 4 показано снятое на модели распределение относительных частот (вероятностей $P\{i\}$) для отдельных вызовов MPI программы **mdrun**

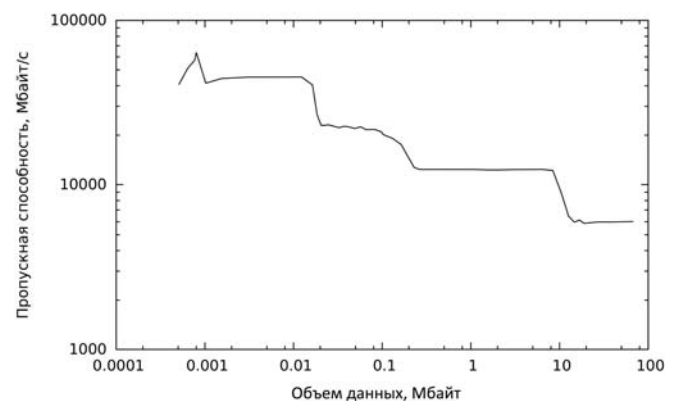


Рис. 2. Результаты измерения пропускной способности компонентов подсистемы памяти (кэшей L1, L2, L3, и ОЗУ) для реальной аппаратуры, полученные с помощью бенчмарка *Lmbench* на реальной аппаратуре. Каждый горизонтальный участок зависимости соответствует своему уровню иерархии памяти

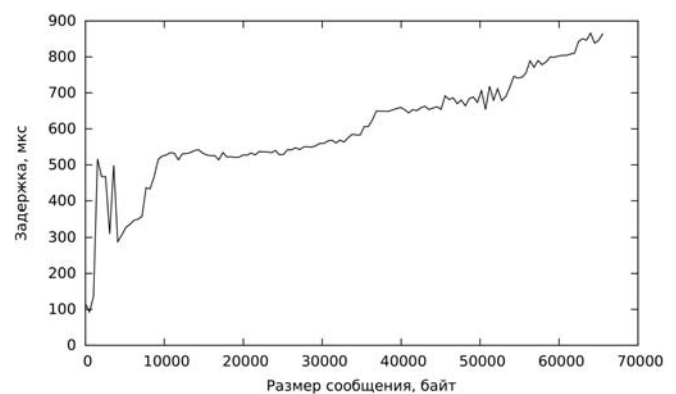


Рис. 3. Зависимость задержки сетевых коммуникаций MPI типа "точка-точка" от размера пакета для изучаемого кластера

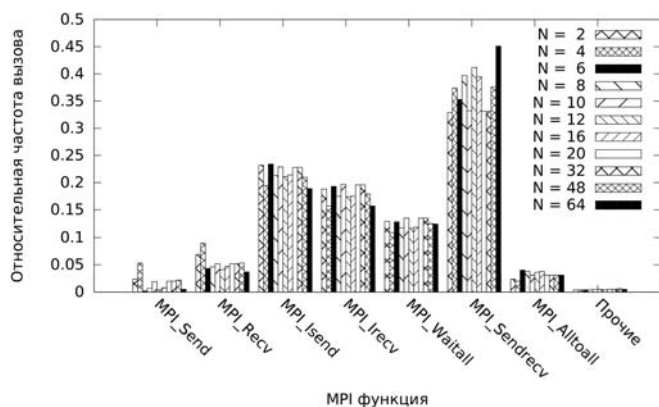


Рис. 4. Распределение частот отдельных *MPI*-вызовов для разного числа моделируемых узлов N для приложения *mdrun* из пакета *Gromacs*

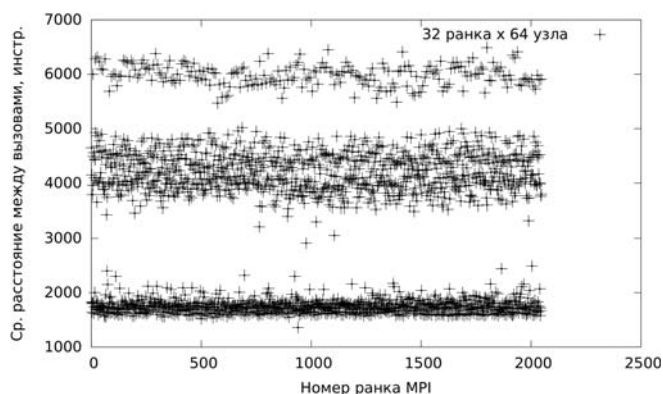


Рис. 5. Зависимость среднего расстояния между вызовами двух *MPI*-функций от порядкового номера подпроцесса (*rank*, ранк) при симуляции исполнения *mdrun* на $N = 64$ узлах (2048 потоков *MPI*)

из пакета *Gromacs*. Видно, что в своей активной фазе это приложение активно использует лишь небольшое число типов коммуникаций. На рис. 5 показаны средние расстояния между двумя *MPI*-вызовами для одного из запусков *mdrun*.

Заключение

Использование компьютерного моделирования для получения сведений о характере работы приложения на еще не построенной вычислительной системе предоставляет исследователю возможность поиска оптимальной конфигурации аппаратуры и наилучшего режима работы приложения. При этом решающим фактором является возможность баланса усилий и времени, потраченных на конфигурирование модели, на проведение непосредственно экспериментов и измерений и на точность получаемых из нее данных. Вследствие масштаба задачи сам процесс моделирования не является тривиальным, однако современные программные решения позволяют выполнить его на доступном оборудовании.

Дальнейшая работа включает в себя следующие направления.

- Сбор данных для большого числа прикладных приложений на симуляции в целях предсказания их производительности.
- Уточнение существующих методик путем введения поправок, учитывающих при работе программ второстепенные процессы. Одним из подобных эффектов является неоднородность скорости исполнения программы на разных потоках/узлах. Этот факт был обнаружен при анализе распределения, показанного на рис. 5, из которого видно, что все процессы делятся на три группы по характерной средней длине интервала между двумя *MPI*-вызовами.
- Верификация результатов, полученных на моделях, с помощью измерений, проведенных на реальной системе, определение достигнутой точности и причин расхождений.

Список литературы

1. **MPI: A Message-Passing Interface Standard. Version 2.2.** // URL: <http://www.mpi-forum.org/docs/docs.html> (дата обращения 16.04.2012).
2. **Magnusson P. S., Christensson M., Eskilson J.** et al. Simics: A Full System Simulation Platform // *Computer* 2002. Т. 35-2. P. 50–58.
3. **Simonson L. J., He L.** Micro-architecture Performance Estimation by Formula // *Proc. of SAMOS'05*. 2005. P. 192–201.
4. **Eyerman S., Eeckhout L., Karkhanis T., Smith J. E.** A mechanistic performance model for superscalar out-of-order processors // *ACM Transactions on Computer Systems*. 2009. Т. 27 (2). P. 3:1–3:37.
5. **Sorin D. J., Lemon J. L., Eager D. L., Vernon M. K.** Analytic Evaluation of Shared-Memory Architectures // *IEEE Transactions on Parallel and Distributed Systems*. 2003. Т. 14. P. 180.
6. **Williams S. W., Waterman A., Patterson D. A.** Roofline: An Insightful Visual Performance Model for Floating-Point Programs and Multicore Architectures // *Tech. rep. UCB/EECS-2008-134*. EECS Department, University of California, Berkeley. 2008. URL: <http://www.eecs.berkeley.edu/Pubs/TechRpts/2008/EECS-2008-134.html> (дата обращения 16.04.2012).
7. **Речистов Г. С., Иванов А. А., Шишпор П. Л., Пентковский В. М.** Симуляционный подход для нахождения производительности параллельных *MPI*-приложений на вычислительном кластере // *Труды 54 научной конференции МФТИ "Проблемы фундаментальных и прикладных естественных и технических наук в современном информационном обществе"*. 2011. С. 82–83.
8. **Van Der Spoel D., Lindahl E., Hesse B.** et al. GROMACS: Fast, Flexible, and Free // *Journal of Computational Chemistry*. 2005. Т. 26, N 16. P. 1701–1718.
9. **Case D. A., Darden T. A., Cheatham T. E.** et al. Amber 11 Users' Manual // University of California. 2010.
10. **Речистов Г. С., Иванов А. А., Шишпор П. Л., Пентковский В. М.** Моделирование компьютерного кластера на распределенном симуляторе. Верификация моделей вычислительных узлов и сети кластера // *Материалы конференции "Разработка ПО 2011" СЕЕ-SECR*. URL: <http://2011.secr.ru/lang/ru-ru/talks/modeling-of-a-computer-cluster-on-a-distributed-simulator> (дата обращения 18.04.2012).
11. **Dongarra J. J.** Performance of Various Computers Using Standard Linear Equations Software. URL: <ftp://netlib2.cs.utk.edu/benchmark/performance.pdf> (дата обращения 18.04.2012).
12. **McCalpin J. D.** STREAM: Sustainable Memory Bandwidth in High Performance Computers. URL: <http://www.cs.virginia.edu/stream> (дата обращения 18.04.2012).
13. **McVoy L., Staelin C.** Lmbench: portable tools for performance analysis // *Proc. of the 1996 annual conference on USENIX Annual Technical Conference*. San Diego, CA: USENIX Association, 1996. С. 23–28.
14. **Dreibholz T.** Netperf: A TCP/UDP/SCTP/DCCP Network Performance Meter Tool. URL: <http://www.iem.uni-due.de/~dreihb/netperf> (дата обращения 18.04.2012).

А. Э. Саак, канд. техн. наук, доц.,
 Технологический институт Южного
 федерального университета в г. Таганроге,
 e-mail: saak@tti.sfedu.ru

Полиномиальные алгоритмы диспетчеризации массивов заявок гиперболического типа

Рассмотрен гиперболический тип массива заявок пользователей на компьютерное обслуживание в Grid-системах, многопроцессорных вычислительных системах. Предложены и исследованы уровневые полиномиальные алгоритмы назначения заявок гиперболического квадратичного типа и даны рекомендации о возможности их использования в диспетчере как МВС, так и Центра Grid-технологий.

Ключевые слова: Grid-система, многопроцессорная вычислительная система, диспетчеризация, гиперболический квадратичный тип массива требований пользователей, уровневый полиномиальный алгоритм

1. Постановка задачи

В работах [1–3] определена квадратичная типизация массивов заявок пользователей на компьютерное обслуживание в Grid-системах, многопроцессорных вычислительных системах (МВС) [5–11]. В работах [12–15] рассмотрена оптимальная укладка последовательности квадратов $1 \times 1, 2 \times 2, \dots, k \times k$, а в работе [16] — оптимальная укладка последовательности прямоугольников с постоянным периметром $1 \times k, 2 \times (k - 1), \dots, (k - 1) \times 2, k \times 1$. Эти исследования дают дополнительную мотивацию рассматриваемому кругу вопросов. В работе [4] для требований кругового квадратичного типа проведен сравнительный анализ предложенных полиномиальных алгоритмов распределения ресурсов. В настоящей статье предлагаются и исследуются уровневые алгоритмы назначения на обслуживание заявок гиперболического квадратичного типа. Вводится среднегиперболический уровень в качестве наиболее важной числовой характеристики гиперболических массивов. Приводится аналитическое определение эвристической меры.

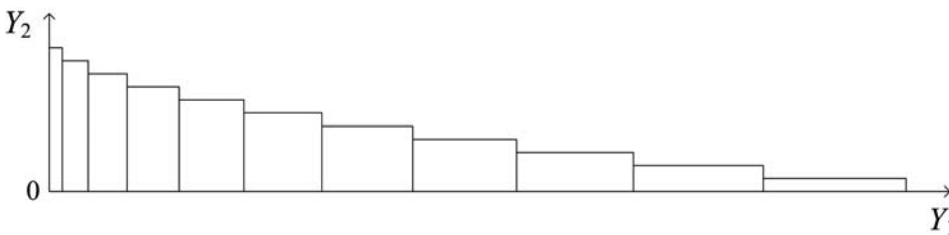


Рис. 1. Гиперболическая линейная полиэдральная оболочка ресурсов

2. Уровневые алгоритмы диспетчеризации заявок гиперболического типа

При представлении заявки пользователя для обслуживания диспетчером Центра Grid-технологий или операционной системы МВС координатным ресурсным прямоугольником горизонтальное и вертикальное измерения, соответственно, принимают равными числу единиц ресурса процессоров и времени, требуемому для обработки. Символом $a(j_1) \times b(j_1)$ или $[(a(j_1), b(j_1))]$ обозначается j_1 -я заявка, требующая $a(j_1)$ единиц процессоров и $b(j_1)$ единиц времени.

Приведем и исследуем уровневый алгоритм диспетчеризации линейными гиперболическими полиэдрами координатных ресурсных прямоугольников $\bigcup_{j_1=0}^{k-1} [(a(j_1), b(j_1))]$ (рис. 1) с убывающими высотами и растущими основаниями $b(j_1) \downarrow, a(j_1) \uparrow, j_1 \uparrow$.

Обозначим через H уровень горизонтальной полосы $0 \leq Y_2 \leq H$. На первом шаге вдоль линии $Y_1 = 0$ вертикально суперпозируются ресурсные прямоугольники

$$\bigcup_{j_1=k-1}^{k-q_1} [(a(j_1), b(j_1))]$$

до наилучшего приближения уровня с недостатком $\sum_{j_1=k-1}^{k-q_1} b(j_1) = H - 0$,

где q_1 — мощность ресурсных прямоугольников, суперпозированных в первом слое. Строится начальная ресурсная оболочка (рис. 2).

На втором шаге вдоль правой стороны достигнутой ресурсной оболочки $Y_1 = a(k - 1)$ вертикально суперпозируются ресурсные прямоугольники $\bigcup_{j_1=k-q_1-1}^{k-q_1-q_2} [(a(j_1), b(j_1))]$ до наилучшего приближения уровня с недостатком

$$\sum_{j_1=k-q_1-1}^{k-q_1-q_2} b(j_1) = H - 0,$$

где q_2 — мощность ресурсных прямоугольников, суперпозированных во втором слое. Строится новая ресурсная оболочка (рис. 3).

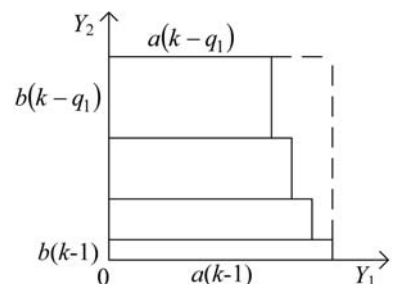


Рис. 2. Начальная ресурсная оболочка

На следующем шаге вдоль правой стороны достигнутой ресурсной оболочки $Y_1 = a(k-1) + a(k-q_1-1)$ вертикально суперпозируются последующие ресурс-

ные прямоугольники $\bigcup_{j_1=k-q_1-q_2-1}^{k-q_1-q_2-q_3} [(a(j_1), b(j_1))]$ до наилучшего приближения уровня с недостатком $\sum_{j_1=k-q_1-q_2-1}^{k-q_1-q_2-q_3} b(j_1) = H - 0$, где q_3 — мощность ресурсных прямоугольников, суперпозированных в третьем слое. Строится достигнутая ресурсная оболочка (рис. 4).

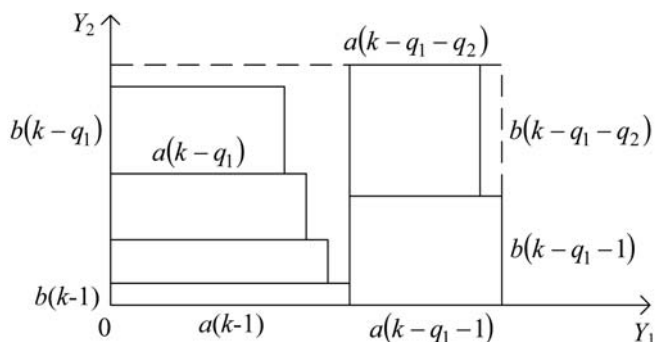


Рис. 3. Первая ресурсная оболочка

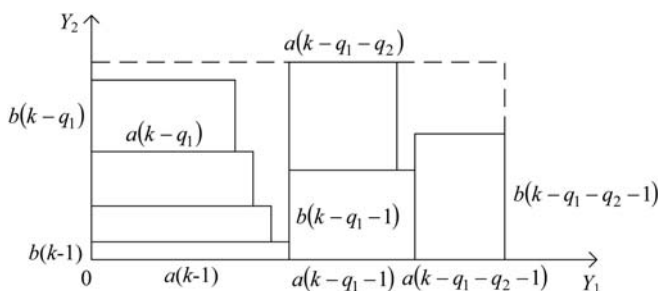


Рис. 4. Вторая ресурсная оболочка

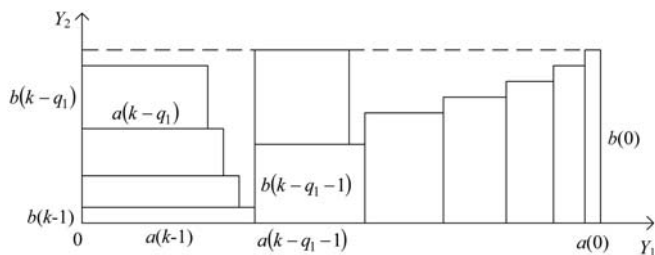


Рис. 5. Конечная ресурсная оболочка

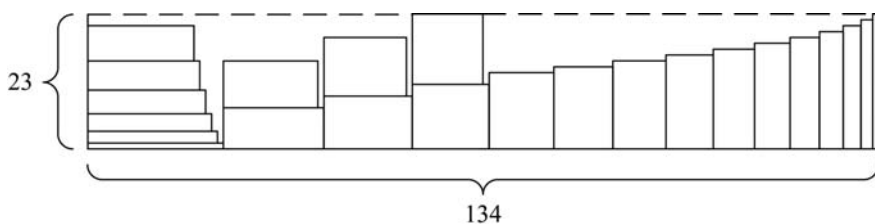


Рис. 6. Укладка гиперболической полиэдрала начальным уровневый алгоритмом

Введенный таким образом уровневый алгоритм повторяем до полного исчерпания ресурсных прямоугольников массива (рис. 5).

Отметим, что число операций работы алгоритма составляет k операций сложения и k операций сравнения, т. е. алгоритм является полиномиальным.

Рассмотрим случай, когда за уровень принимаем высоту начального ресурсного прямоугольника $H = b(0)$. В частности, для линейной полиэдрала гиперболических ресурсных прямоугольников $j_1 \times (k - j_1)$, $j_1 \in [1, k - 1] \subset Z^1$ при $k = 24$ соответствующие построения начальным уровневый алгоритмом приведены на рис. 6.

Время работы оптимального алгоритма [16] приведено в табл. 1.

Сравним качество приведенного начального уровневый алгоритма с оптимальной укладкой в объемлющий прямоугольник минимальной площади, полученной в работе [16]. Приведем в табл. 2 результаты размещения начальным уровневый алгоритмом для последовательности гиперболических ресурсных прямоугольников $1 \times (k - 1)$, $2 \times (k - 2)$, ..., $(k - 1) \times 1$. Здесь L — горизонтальное и H — вертикальное измерения объемлющего прямоугольника начального уровневый алгоритма; Δ — погрешность площади ресурсной оболочки в процентах относительно оптимального значения.

Видим, что погрешность не превосходит 47,4 %, что является подтверждением целесообразности использования предложенного начального уровневый алгоритма при диспетчировании процессорно-временными ресурсами.

Рассмотрим случай, когда за уровень принимаем среднюю гиперболическую величину $H = \sqrt{\sum_{j_1=0}^{k-1} a(j_1)b(j_1)}$. В частности, для линейной поли-

эдрала гиперболических ресурсных прямоугольников $j_1 \times (k - j_1)$, $j_1 \in [1, k - 1] \subset Z^1$ при $k = 24$ соответствующие построения алгоритмом со средним гиперболическим уровнем приведены на рис. 7.

Сравним качество приведенного алгоритма со средним гиперболическим уровнем с оптимальной укладкой в объемлющий прямоугольник минимальной площади, полученной в работе [16]. Приведем в табл. 3 результаты размещения алгоритмом со средним гиперболическим уровнем для последовательности гиперболических ресурсных прямо-

Таблица 1

k	Время работы оптимального алгоритма, 2,93 GHz Intel Core 2 Duo E7500			
	дни	часы	минуты	секунды
20	—	—	02	15
21	—	—	07	51
22	—	—	11	20
23	—	9	12	37
24	3	22	50	38

угольников $1 \times (k - 1)$, $2 \times (k - 2)$, ..., $(k - 1) \times 1$. Здесь L — горизонтальное и H — вертикальное измерение объемлющего прямоугольника начального уровня алгоритма; Δ — погрешность площади ресурсной оболочки в процентах относительно оптимального значения.

Видим, что погрешность не превосходит 40,3 %, что является подтверждением целесообразности использования предложенного алгоритма со средним гиперболическим уровнем при диспетчировании процессорно-временными ресурсами.

3. Эвристическая мера ресурсных оболочек полиномиальных алгоритмов

Управление ресурсами, наряду с проблемой оптимальности, требующей сложных по трудоемкости алгоритмов распределения, ставит проблему полиномиальных алгоритмов диспетчирования с оценкой качества посредством эвристической меры.

Так, эвристической мерой координатного ресурсного прямоугольника $L \times H$ на Z^2 -плоскости с аддитивными подмножествами — координатными ресурсными прямоугольниками $a \times b$, $\lambda \times \beta$ (рис. 8) называем величину

$$\frac{1}{2} \left(\frac{L \cdot H}{a \cdot b + \lambda \cdot \beta} + \frac{(L - H)^2}{a \cdot b + \lambda \cdot \beta} \right)$$

и ставим вопрос о минимизации данного целевого критерия надлежащей диспетчированием координатных ресурсных прямоугольников в ресурсную оболочку.

Для большего числа ресурсных прямоугольников эвристическая мера определяется полусуммой отношений ресурсных мер оболочки аддитивной графики граней к суммарной мере охватываемых планарных элементов и мерой асимметрии измерений оболочки к тому же значению суммарной меры. Приведем в табл. 4 эвристические меры ресурсных оболочек начального уровня алгоритма и алгоритма со средним гиперболическим уровнем для последовательности гиперболических ресурсных прямоугольников $1 \times (k - 1)$, $2 \times (k - 2)$, ..., $(k - 1) \times 1$.

Видим, что эвристические меры ресурсных оболочек алгоритма со средним гиперболическим уровнем меньше. Следовательно, более предпочтительным при диспетчировании процессорно-временными ресурсами является использование именно этого алгоритма.

Заключение

В статье для гиперболического типа массива заявок пользователей рассмотрены уровневые полиномиальные алгоритмы диспетчирования. Введено аналитическое определение эвристической меры. Проведен сравнительный анализ уровневых полиномиальных алгоритмов распределения вычислительных ресурсов и даны рекомендации о возможности их использования в диспетчере как МВС, так и Центра Grid-технологий.

Таблица 2

Результаты сравнения начального уровня и оптимального алгоритмов

k	$L \times H$	$\Delta, \%$
14	50 × 13	40,0
15	60 × 14	47,4
16	61 × 15	31,5
17	72 × 16	39,1
18	75 × 17	29,6
19	87 × 18	35,9
20	100 × 19	41,4
21	114 × 20	46,7
22	115 × 21	35,3
23	130 × 22	40,2
24	134 × 23	32,9

Таблица 3

Результаты сравнения алгоритмов со средним гиперболическим уровнем и оптимального

k	$L \times H$	$\Delta, \%$
14	31 × 21	40,3
15	33 × 24	38,9
16	37 × 25	32,9
17	38 × 28	28,5
18	43 × 31	35,5
19	50 × 31	34,5
20	51 × 36	36,6
21	53 × 39	33,0
22	55 × 42	29,4
23	61 × 45	34,6
24	63 × 46	25,0

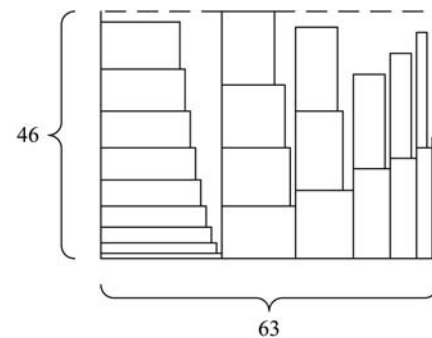


Рис. 7. Укладка алгоритмом со средним гиперболическим уровнем

Таблица 4

Сравнение эвристических мер ресурсных оболочек

k	Эвристическая мера ресурсной оболочки	
	начального уровня алгоритма	алгоритма со средним гиперболическим уровнем
14	2,219	0,825
15	2,639	0,779
16	2,229	0,786
17	2,627	0,713
18	2,394	0,762
19	2,775	0,838
20	3,181	0,775
21	3,609	0,735
22	3,176	0,700
23	3,588	0,741
24	3,348	0,693

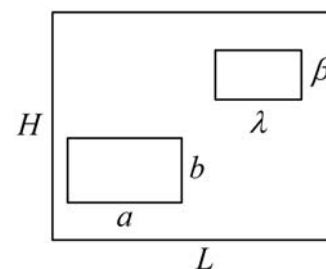


Рис. 8. Ресурсный прямоугольник с аддитивными подмножествами

Список литературы

1. Саак А. Э. Локально-оптимальные ресурсные распределения // Информационные технологии. 2011. № 2. С. 28–34.
2. Саак А. Э. Алгоритмы диспетчеризации в Grid-системах на основе квадратичной типизации массивов заявок // Информационные технологии. 2011. № 11. С. 9–13.
3. Саак А. Э. Диспетчеризация в Grid-системах на основе однородной квадратичной типизации массивов заявок пользователей // Информационные технологии. 2012. № 4. С. 32–36.
4. Саак А. Э. Сравнительный анализ полиномиальных алгоритмов диспетчеризации в Grid-системах // Информационные технологии. 2012. № 9. С. 28–32.
5. Барский А. Б. Параллельные информационные технологии. М.: ИНТУИТ; БИНОМ. Лаборатория знаний, 2007. 503 с.
6. Барский А. Б. Параллельные информационные технологии в основе Grid-системы // Информационные технологии. 2006. № 12. С. 54–60.
7. Васенин В. А., Шундеев А. С. Эволюция технологии Grid // Информационные технологии. 2012. № 1. С. 2–9.
8. Васенин В. А., Инохин А. В., Шевелев М. В. Вычислительный Grid-полигон: состояние, идеи, решения // Информационные технологии. 2009. № 7. Приложение. 32 с.
9. Хорошевский В. Г. Архитектура вычислительных систем. М.: Изд-во МГТУ им. Н. Э. Баумана, 2005. 512 с.
10. Воеводин В. В., Воеводин Вл. В. Параллельные вычисления. СПб.: БХВ-Петербург, 2002. 608 с.
11. Каляев И. А., Левин И. И., Семерников Е. А., Шмойлов В. И. Реконфигурируемые мультимедийные вычислительные структуры / Изд. 2-е, перераб. и доп. / Под общ. ред. И. А. Каляева. Ростов н/Д: Изд-во ЮНЦ РАН, 2009. 344 с.
12. Korf R. Optimal rectangle packing: Initial results // Proc. of the thirteenth international conference on automated planning and scheduling (ICAPS 2003). Trento, Italy, June 9–13, 2003. P. 287–295.
13. Korf R. Optimal rectangle packing: New results // In Proc. of the fourteenth international conference on automated planning and scheduling (ICAPS 2004). Whistler, British Columbia, Canada, June 3–7, 2004. P. 142–149.
14. Korf R., Moffitt M., Pollack M. Optimal rectangle packing // Annals of Operations Research. 2010. Vol. 179, N 1. P. 261–295.
15. Korf R., Huang E. New Improvements in Optimal Rectangle Packing // Proc. of the 21st International Joint Conference on Artificial Intelligence (IJCAI 2009). Pasadena, California, USA, July 11–17, 2009. P. 511–516.
16. Korf R., Huang E. Optimal Rectangle Packing on Non-Square Benchmarks // Proc. of the twenty-fourth AAAI Conference on Artificial Intelligence (AAAI-10). Atlanta, Georgia, USA, July 11–15, 2010. P. 83–88.

КОМПЬЮТЕРНАЯ ГРАФИКА И ОБРАБОТКА ИЗОБРАЖЕНИЙ

УДК 004.93

В. А. Бобков, д-р. техн. наук, зав. лаб.,

e-mail: bobkov@iacp.dvo.ru,

Ю. И. Роншин, вед. инж.,

e-mail: ronshin@iacp.dvo.ru,

В. Ю. Машентцев, аспирант,

e-mail: v.mashentsev@gmail.com,

А. П. Кудряшов, канд. техн. наук,

e-mail: Kudryashova@iacp.dvo.ru,

Институт автоматизации и процессов управления
ДВО РАН, г. Владивосток

Навигация автономного подводного аппарата по видеопотоку¹

Предложен метод определения траектории подводного аппарата по видеoinформации, получаемой двумя бортовыми камерами. Метод основывается на сопоставлении точечных особенностей на последовательности изображений и вычислении матриц геометрических преобразований между локальными системами координат в разные моменты времени движения аппарата. Вычисление преобразований осуществляется методом нелинейной оптимизации по данным сопоставленных 3D-облаков точек-особенностей наблюдаемой обстановки.

Ключевые слова: подводный аппарат, навигация, SLAM, 3D-облако точек, сопоставление особенностей, фильтрация, калибровка, CUDA

¹ Работа выполнена при поддержке РФФИ (проект № 11-07-00088-а), Программ Президиума РАН (проекты 12-1-П15-05, 12-1-П18-03).

Введение

Идея использования видеоизображений для навигации подводного аппарата (ПА) так же, как и задача SLAM (Simultaneous Localization and Mapping), привлекает внимание исследователей на протяжении последних двух десятилетий. Одно из преимуществ визуального подхода состоит в том, что сегодня можно оснащать ПА распространенными, недорогими и компактными видеокамерами с высоким разрешением. Известно, что режим навигации по видеoinформации целесообразен при локальном маневрировании автономного ПА (при малых перемещениях и скоростях), когда традиционно используемые гидроакустические сонары не всегда обеспечивают требуемую точность позиционирования. С помощью методов компьютерного зрения можно не только вычислять реальную траекторию ПА и расстояния до окружающих объектов, но и решать задачу 3D-реконструкции подводной обстановки, что важно для ряда практических задач.

Первые решения задачи SLAM оперировали с акустическими данными. Одним из наиболее известных подходов является подход, основанный на применении фильтра Калмана. В последующих SLAM-разработках с использованием визуальной информации в основном применялись подходы с анализом 2D-свойств захватываемого роботом видеопотока. В последнее время появились работы, где решения строятся на основе совместного анализа свойств 2D-видеоданных и свойств генерируемых 3D-структур. Например, такой подход использовался в работе [1] по реконструкции подводной об-

становки. Подход основывался на комбинации фильтра Калмана со сглаживающим фильтром Rauch-Tung-Striebel. В работе [2] была предложена схема интегрированной обработки визуальных данных и данных измерений инерциального сенсора. В работе [3] сравнивались акустический и визуальный методы определения траектории робота. Здесь для идентификации особенностей на видеоснимках использовалась упрощенная схема с применением детектора Харриса. В работе [4] решалась задача SLAM на основе анализа визуальных 2D- и 3D-данных с оценкой энтропии в качестве основного критерия для объединения локальных данных реконструкции в мировой системе координат. В указанных работах можно найти и некоторый обзор по затрагиваемой тематике.

В настоящей статье предлагается метод навигации автономной ПА, оснащенной стереопарой, основанный на прослеживании точечных особенностей на видеоизображениях и сопоставлении соответствующих 3D-облаков точек при движении ПА по траектории. Работа является продолжением исследования, выполненного в [5], где решалась задача видеоSLAM с использованием расширенного фильтра Калмана применительно к ПА. Отличительной характеристикой предлагаемого метода является оригинальная вычислительная методика калибровки камер с многоступенчатой схемой фильтрации особенностей.

1. Постановка задачи

Предполагается, что ПА оснащен двумя видеокамерами и бортовым вычислителем, обеспечивающим программно-алгоритмическую обработку поступающих видеоданных. При движении ПА по пространственной траектории осуществляется синхронный захват кадров окружающей обстановки (морского дна и других объектов) с некоторой частотой. Относительная калибровка установленных камер фиксирована и задана. Каждая из пар снимков последовательности — стереопара, — получаемая одновременно с двух камер, привязана к определенному моменту времени t_i движения по траектории. Пространственное положение аппарата в мировой системе координат определяется шестью параметрами: его тремя координатами и тремя углами ориентации.

В качестве мировой системы координат может быть выбрана, например, система координат (СК), привязанная к одной из камер в момент начала движения t_0 . Задача заключается в точном определении параметров траектории в моменты t_0, t_1, \dots, t_N . Так как камеры жестко привязаны к аппарату, определение параметров траектории в каждый момент времени t_i эквивалентно вычислению матрицы преобразования из мировой СК в СК одной из камер, например левой (задача калибровки). Вычисления преобразований основываются на сопоставлении

особенностей на изображениях (снимках), в качестве которых рассматриваются точечные особенности. Поскольку точность вычислений зависит от числа и достоверности анализируемых особенностей, возникает и сопутствующая задача определения стратегии выбора моментов времени t_0, t_1, \dots, t_N на траектории, обеспечивающей оптимальное решение по критериям точности, времени расчетов и плотности точек на траектории.

Одновременно с задачей определения траектории должна решаться и другая практически важная для маневрирования ПА задача точного определения расстояний до видимых объектов окружающей обстановки. Ее решение также основывается на сопоставлении особенностей стереопары или множественной последовательности изображений.

2. Определение траектории по последовательности изображений

2.1. Описание метода

Применяется адаптивная стратегия выбора вычисляемых точек на траектории. Будем использовать следующие обозначения: пусть $\mathbf{p}(t_i)$ — i -я точка на траектории, $i = 1, \dots, N$; t_i — время движения ПА по траектории, относящееся к положению ПА в i -й точке; Δ — шаг по времени (шаг между двумя соседними парами снимков в последовательности), который используется для выбора очередной точки на траектории: $t_i = t_{i-1} + \Delta$.

Предварительно пользователем задается максимальный шаг во времени Δ_{\max} , с которого начинается итеративная процедура выбора точки, и минимальный Δ_{\min} , который ограничивает число итераций. Для каждого шага t_i на траектории анализируется пара снимков из данной точки $\mathbf{p}(t_i)$ траектории и пара снимков из предыдущей точки $\mathbf{p}(t_{i-1})$. Если число полученных особенностей окажется меньше порогового значения NF , установленного пользователем, то шаг Δ уменьшается и начинается новая итерация.

Предлагаемый метод состоит из последовательности следующих шагов:

1. На каждом снимке определяется множество особенностей и устанавливается взаимно однозначное соответствие между полученными четырьмя множествами (см. п. 2.2).

2. К каждой паре множеств особенностей, отвечающей стереопаре, применяется фильтр, основанный на учете эпиполярного ограничения (см. п. 2.3). Это возможно, поскольку для жестко установленной на ПА пары камер известна калибровка.

3. Для каждой стереопары определяется (триангуляцией) облако 3D-точек, заданных в СК этой пары.

4. Построенные облака пространственных точек обрабатываются 3D-фильтром, который основан на оценке сопоставления пространственного соседства точек в облаках (см. п. 2.4). Если число точек в об-

лаках меньше порогового значения (задаваемого пользователем), то шаг Δ уменьшается в два раза и действия 1—4 повторяются до тех пор, пока не будет необходимого числа точек или шаг не станет минимальным.

5. По полученным сопоставленным облакам методом нелинейной оптимизации вычисляется локальное преобразование (матрица \mathbf{H}_i) из СК камеры, связанной с $(i - 1)$ -й точкой на траектории (момент времени t_{i-1}), в СК камеры, связанную с i -й точкой (момент времени t_i) (см. п. 2.5).

2.2. Сопоставление особенностей на изображениях

Сопоставление точечных особенностей выполняется для четырех снимков — двух стереопар, соответствующих моментам времени t_{i-1} и t_i . Для этой цели использовалась библиотека OpenCV. Поскольку точность вычисления преобразований полностью зависит от качества (достоверности) определяемых соответствий особенностей на изображениях, была реализована схема сопоставления с многоступенчатой фильтрацией, существенно повышающей достоверность получаемых соответствий. Суть ее заключается в последовательном выполнении кросс-проверки как для отдельных пар

снимков, так и для всей цепочки изображений. Схема обработки состоит из нескольких шагов. Для каждого снимка определяется множество точечных особенностей (исходное). Для этой цели в данной реализации использовался SURF-детектор. Рассматриваются три пары снимков:

- а) левый для 1-й стереопары (изображение 1) — правый для 1-й стереопары (изображение 2);
- б) правый для 1-й стереопары — левый для 2-й стереопары (изображение 3);
- в) левый для 2-й стереопары — правый для 2-й стереопары (изображение 4).

Для каждой такой пары выполняется сопоставление слева направо и справа налево (cross-checking). Мы использовали класс SurfDescriptorExtractor для определения дескрипторов и класс BruteForceMatcher для сопоставления. Особенности, для которых расстояние между дескрипторами больше заданного (пользователем), удаляются — для этого применяется фильтр по близости дескрипторов. Результаты согласовываются таким образом, чтобы получить множество особенностей, сопоставленных по всей цепочке изображений 1—2—3—4 (рис. 1). Схема обработки (фильтрации) всей цепочки изображений работает следующим образом. Если \mathbf{M}_1 и \mathbf{M}_2 — исходные множества особенностей на снимках 1 и 2, то после сопоставления получаем взаимно однозначно сопоставленные множества $\mathbf{M}_{12} \leftrightarrow \mathbf{M}_{21}$, где $\mathbf{M}_{12} \in \mathbf{M}_1$, а $\mathbf{M}_{21} \in \mathbf{M}_2$. Здесь и далее первый индекс относится к номеру снимка.

Для пары б) в качестве исходного множества первого снимка для сопоставления используется множество \mathbf{M}_{21} . В результате получаем сопоставление $\mathbf{M}_{23} \leftrightarrow \mathbf{M}_{32}$, где $\mathbf{M}_{23} \in \mathbf{M}_{21}$, а $\mathbf{M}_{32} \in \mathbf{M}_3$, где \mathbf{M}_3 — исходное множество особенностей для снимка 3.

Аналогичным образом получаем для пары в): $\mathbf{M}_{34} \leftrightarrow \mathbf{M}_{43}$, где $\mathbf{M}_{34} \in \mathbf{M}_{32}$, а $\mathbf{M}_{43} \in \mathbf{M}_4$, где \mathbf{M}_4 — исходное множество для снимка 4. Таким образом, после прохождения всей цепочки получаем минимальное множество особенностей $\mathbf{M}_{43} \leftrightarrow \mathbf{M}_{34}$.

Последующий обратный проход цепочки 4—3—2—1, начиная от исходного множества \mathbf{M}_{43} , дает в итоге последовательность взаимно однозначно сопоставленных множеств $\mathbf{M}_{43} \leftrightarrow \mathbf{M}_{34} \leftrightarrow \mathbf{M}_{23} \leftrightarrow \mathbf{M}_{12}$, где $\mathbf{M}_{23} \in \mathbf{M}_{23}$, а $\mathbf{M}_{12} \in \mathbf{M}_{12}$. То есть из пар а) и б) удаляются особенности, для которых нет соответствий во множестве \mathbf{M}_{43} . На рис. 2 показан результат работы описанной схемы для одной из позиций на траектории в вычислительном эксперименте с модельной сценой.

Исходное число особенностей для каждого из изображений было соответственно 469, 492, 502, 508. Для сопоставлений пар 1—2, 2—3 и 3—4 (см. рис. 1) число точек уменьшается соответственно до 296, 149 и 115. Фильтрация при обратном проходе дает в итоге 115 сопоставленных на всех четырех изображениях особенностей.

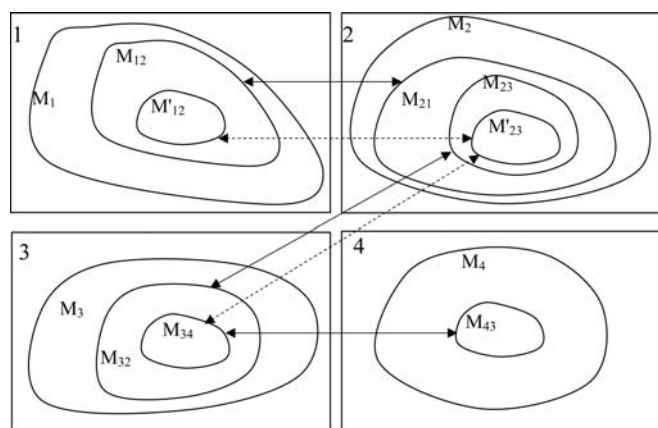


Рис. 1. Схема сопоставления и фильтрации точечных особенностей на последовательности из четырех изображений (2 стереопары, относящиеся к двум соседним позициям на траектории)

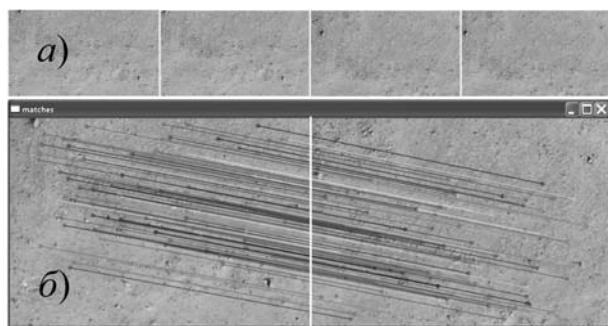


Рис. 2. Пример сопоставления особенностей: а — четыре снимка двух стереопар; б — сопоставленные особенности на 1-м и 4-м снимке

2.3. Эпиполярная фильтрация особенностей

В результате сопоставления получаются четыре множества сопоставленных точечных особенностей — по два множества для каждой стереопары. Так как калибровка для стереопар известна, то для каждой стереопары можно построить фундаментальную матрицу \mathbf{F} следующим образом. Для калиброванных камер известны следующие параметры: \mathbf{t} — вектор переноса камеры; \mathbf{R} — матрица поворота камеры; f — фокусное расстояние. Существенная матрица \mathbf{E} вычисляется по формуле

$$\mathbf{E} = [\mathbf{t}]_x \mathbf{R},$$

где $[\mathbf{t}]_x$ — матрица векторного произведения. Фундаментальная матрица \mathbf{F} определяется по формуле

$$\mathbf{F} = \mathbf{C}^{-1T} \mathbf{E} \mathbf{C}^{-1},$$

где \mathbf{C} — калибровочная матрица:

$$\mathbf{C} = \begin{Bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{Bmatrix}.$$

Далее для каждой стереопары выполняется проверка на эпиполярное соответствие каждой из пар точек — особенностей: оценивается расстояние между особенностью \mathbf{m}_2 на втором изображении и эпиполярной линией $\mathbf{m}_1 \cdot \mathbf{F}$, полученной из соответствующей особенности \mathbf{m}_1 на первом изображении. Если это расстояние больше заданного порога, такие точки удаляются. Результаты фильтрации для первой и второй стереопар согласовываются. Данный фильтр дополнительно удаляет три точки из 115.

2.4. Построение и фильтрация сопоставленных 3D-облаков точек

Построение облака 3D-точек, соответствующих сопоставленным особенностям на двух снимках стереопары, выполняется обычной триангуляцией (пересечение двух лучей в системе координат стереопары). Наличие двух сопоставленных облаков 3D-точек, наблюдаемых с двух соседних позиций ПА на траектории, позволяет выполнить дополнительную фильтрацию, основанную на учете сохраняющегося взаиморасположения точек в облаке. Таким образом, на входе этого фильтра имеются два облака взаимно однозначно сопоставленных 3D-точек $C_1(x, y, z)$ и $C_2(x, y, z)$, относящихся, соответственно, к моментам времени t_{i-1} и t_i , где i — номер шага на траектории.

Работа фильтра строится на предположении, что расстояние d_1 между двумя точками в облаке C_1 должно быть равно расстоянию d_2 между соответствующими точками в облаке C_2 . Оценка рассогласования для пары точек определяется как $D = d_2/d_1 - 1$, где $d_2 > d_1$. Оценка рассогласования для точки определяется как $\Sigma(D_j)/(n - 1)$, где D_j —

оценка рассогласования между данной точкой и любой другой в облаке; n — число точек. Фильтр работает итерационно. На первом шаге для всех точек вычисляются оценки рассогласования и выбирается точка с максимальной оценкой. Если эта оценка больше заданного (пользователем) порога, соответствующие ей точки из облака C_1 и облака C_2 удаляются. На следующем шаге вычисления повторяются для новых облаков. Процесс кончается, когда в облаках не остается точек с оценкой рассогласования, большей порога. Следует отметить, что такого рода фильтрация удаляет в том числе и особенности, связанные с динамическими объектами, характерными для подводной среды (рыбы и другие перемещающиеся объекты).

2.5. Вычисление локального преобразования

Перемещение ПА из точки $\mathbf{p}(t_{i-1})$ траектории в точку $\mathbf{p}(t_i)$ определяется некоторым неизвестным преобразованием \mathbf{H}_i , состоящим из вектора переноса и кватерниона вращения, i — номер шага на траектории. Будем искать это преобразование, основываясь на имеющемся взаимно однозначном сопоставлении двух облаков 3D-точек, наблюдаемых парой камер соответственно в моменты времени t_{i-1} и t_i . Первое облако C_1 задано в СК 1-й стереопары, 2-е облако C_2 — в СК 2-й стереопары. Задача нахождения матрицы \mathbf{H}_i решается методом нелинейной оптимизации с ограничением (используется библиотека MATLAB). В качестве параметров оптимизации используются три координаты вектора переноса и четыре координаты кватерниона, определяющего вращение \mathbf{H}_i . Ограничение задается условием: норма кватерниона = 1. Целевая функция F_i определяется следующим образом: пусть $\{\mathbf{c}_{1j}\}$ — множество точек в первом облаке и $\{\mathbf{c}_{2j}\}$ — множество точек во втором облаке, тогда

$$F_i = \Sigma \|\mathbf{c}_{2j} - \mathbf{c}_{1j} \cdot \mathbf{H}_i\|, j — номер точки в облаке.$$

С учетом вычисленной матрицы \mathbf{H}_i новое положение ПА определяется через предыдущее как $\mathbf{p}(t_i) = \mathbf{p}(t_{i-1}) \cdot \mathbf{H}_i$. В качестве точки, описывающей траекторию ПА, можно рассматривать точку начала СК камеры (0, 0, 0). Поскольку нам нужны координаты точек траектории в мировой СК, необходимо вычислить преобразование из мировой СК в СК i -й точки. Если в качестве мировой СК выбрана СК первой пары в последовательности, то \mathbf{H}_0 — единичная матрица (для произвольной мировой СК необходимо знать \mathbf{H}_0). С учетом знания локальных преобразований для всех предшествующих точек траектории преобразование \mathbf{Hw}_i из мировой СК в СК i -й точки вычисляется как последовательное произведение локальных преобразований: $\mathbf{Hw}_i = \mathbf{H}_0 \cdot \mathbf{H}_1 \cdot \dots \cdot \mathbf{H}_i$. Соответственно, мировые координаты i -й точки траектории ПА вычисляются как $\mathbf{p}(t_i) = \mathbf{p}(t_0) \cdot \mathbf{Hw}_i$.

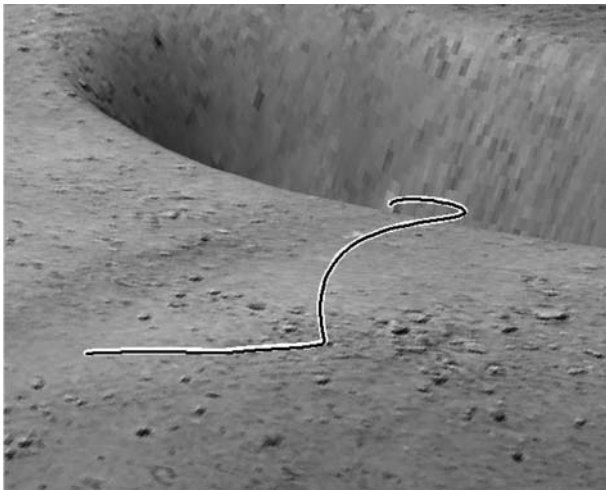


Рис. 3. Модельная сцена: показана реальная (темная линия) и вычисленная (светлая линия) траектория движения ПА

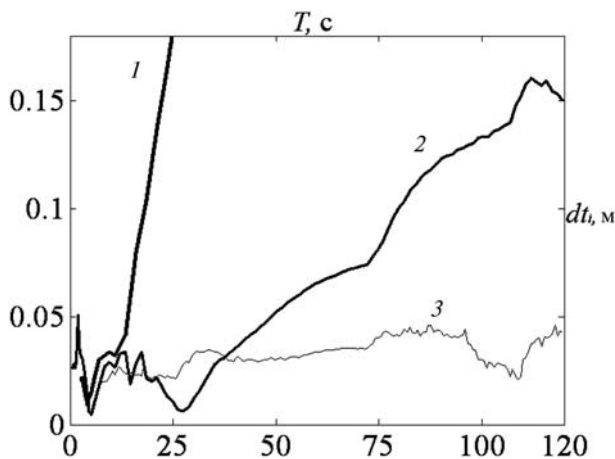


Рис. 4. Ошибка в вычислении смещения траектории в зависимости от времени:
 1 — график зависимости для $\Delta_{\max} = 2,56$ с (64 кадра); 2 — график для $\Delta_{\max} = 1,28$ с (32 кадра); 3 — график для $\Delta_{\max} = 0,64$ с (16 кадров)

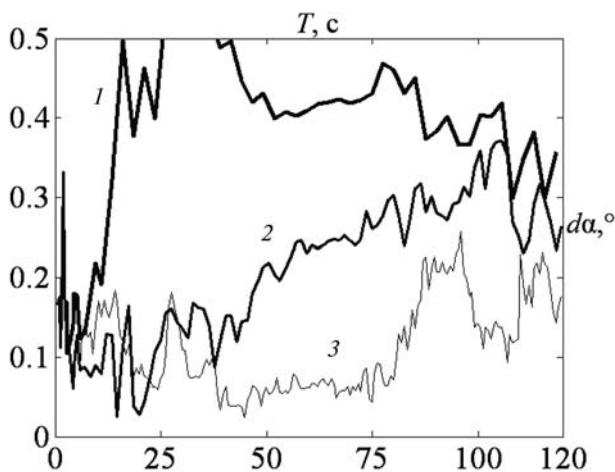


Рис. 5. Ошибка в вычислении ориентации ПА (усредненное по трем углам отклонение) в зависимости от времени. Графики соответствуют трем режимам, обозначенным в рис. 4

3. Вычисление расстояний до объектов окружающей обстановки

Эту задачу можно решать двумя методами. Первый метод базируется на использовании средств библиотеки OpenCV для определения и сопоставления особенностей на паре изображений. Вычисление расстояний до каждой из пространственных точек, соответствующих сопоставленным особенностям, эквивалентно построению облака 3D-точек в системе координат стереопары (см. выше), т. е. $r_j = \|c_j\|$.

Второй метод состоит в применении "оптического дальномера", основанного на построении карт глубин для калиброванных изображений [6]. В отличие от первого метода здесь возможно задание точек в узлах регулярной решетки на левом снимке. Для них "оптический дальномер" получает 3D-точки, что, как уже было выше отмечено, эквивалентно вычислению "дальностей". Другое преимущество применения "оптического дальномера" заключается в том, что при вычислении карт глубин используется технология CUDA на графических процессорах, что значительно ускоряет вычисления.

4. Результаты вычислительных экспериментов

Тестирование разработанных программно-алгоритмических средств проводилось на модельных сценах, одна из которых показана на рис. 3. При движении ПА по заданной траектории и с частотой 25 кадр/с получали и сохраняли синхронные снимки подводной обстановки (морского дна) с двух камер (стереопара). Высота траектории изменялась в пределах 3,5 м... 7 м, длина соответствовала 120 с движения ПА. Матрицы Hw , вычисленные описанным выше методом, сравнивались с идеальными матрицами Hid преобразования из мировой СК в СК левой камеры. Для оценки точности метода на каждом шаге движения ПА по траектории вычислялось расхождение по вектору переноса и по углам вращения.

Расхождение по вектору переноса на шаге i определялось по формуле

$$dt_i = \|p_0 \cdot Hw_i^{-1} - p_0 \cdot Hid_i^{-1}\|,$$

где p_0 — точка с координатами (0, 0, 0) в СК левой камеры. Усредненное расхождение по углам ориентации камеры (ПА) на шаге i вычислялось по формуле

$$d\alpha_i = 1/3 \sum D\alpha_n,$$

где $D\alpha_n$ — угол между вычисленным ортом и идеальным ортом для ортов i, j, k .

Результаты вычислительных экспериментов по оценке точности расчета траектории представлены на рис. 4. и 5. На рис. 4 показан график поведения ошибки dt_i (отклонение вычисляемой траектории от идеальной) в зависимости от времени движения

по траектории для трех вариантов выбора шага Δ_{\max} (интервала времени между точками траектории, где вычисляются локальные матрицы преобразования). Минимальное число особенностей NF (настраиваемый параметр) в проведенных экспериментах равнялось 30. Для эпиполярного фильтра был установлен порог в 0,5 пикселя, а для 3D-фильтра — 3 см. Для варианта, когда последовательное вычисление точек траектории делалось с шагом $\Delta_{\max} = 2,56$ с, средняя ошибка отклонения вычисленной траектории от реальной равна 34,8 см. Для варианта с шагом $\Delta_{\max} = 1,28$ с эта ошибка равна 7 см. И, наконец, лучший результат получен при $\Delta_{\max} = 0,64$ с — ошибка равна 3,2 см.

Из анализа представленных графиков следует, что в первом и во втором случаях происходит достаточно быстрое накопление ошибки в зависимости от времени движения по траектории. В варианте с шагом $\Delta_{\max} = 0,64$ с на протяжении всей траектории сохраняется сравнительно высокая и устойчивая точность вычисления траектории. Такие результаты объясняются тем, что при больших временных интервалах между позициями ПА одновременно наблюдаемое с соседних позиций число особенностей становится недостаточным для точного вычисления калибровки камер, что приводит к быстрому накоплению ошибки. Этот вывод подтверждается полученными в экспериментах данными по числу используемых алгоритмом особенностей на каждом шаге движения по траектории.

На рис. 5 показан график поведения ошибки вычисления ориентации ПА $d\alpha$ для каждого из указанных выше трех вариантов. Усредненная по всей траектории ошибка для каждого из вариантов равна, соответственно, $0,37^\circ$, $0,2^\circ$ и $0,11^\circ$.

Экспериментальные оценки затрачиваемого на вычисления времени показали, что уменьшение

шага Δ_{\max} в два раза приводит к увеличению счетного времени примерно в два раза. Для варианта 3 время обхода одной точки траектории составляет порядка 1 с (процессор Intel (R) Core™ i5).

Оценка точности вычисления расстояний до объектов (каждым из двух описанных выше методов) строилась на сравнении восстановленных точек сцены с "идеальными" 3D-точками. В качестве "идеальной" точки рассматривалась точка пересечения луча, выходящего из центра левой камеры и проходящего через точку на снимке, с триангуляционной сеткой, используемой при построении объектов модельной сцены. На рис. 6 показан график поведения ошибки вычисления расстояний L до объектов в зависимости от времени движения по траектории для первого метода и для второго метода (см. п. 3). Из анализа полученных графиков можно сделать два вывода. Во-первых, ошибка, как и можно было предположить, практически не растет при движении по траектории, поскольку расстояния вычисляются в системе координат камеры. Во-вторых, "дальномер" показал более высокую точность в сравнении с первым методом, базирующемся на библиотеке OpenCV. Средняя ошибка для первого метода в проведенном эксперименте — 2,15 см, для второго метода — 1,27 см.

Заключение

Вычислительные эксперименты показали, что предложенный метод калибровки камер по генерируемым 3D-облакам точек является хорошей альтернативой подходу, основанному на использовании фильтра Калмана. Однако время вычислений не в полной мере отвечает режиму реального времени, поэтому в дальнейшем планируется увеличить быстродействие за счет использования технологии CUDA на этапе сопоставления особенностей.

Список литературы

1. Salvi J., Petillot Y., Battle E. Visual SLAM for 3D Large — Scale Seabed Acquisition Employing Underwater Vehicles // IEEE International Conference on Intelligent Robots and Systems. 2008. P. 1011—1016.
2. Hogue A., German A., Jenkin M. Underwater environment reconstruction using stereo and inertial data // Systems Man and Cybernetics ISIC IEEE International Conference, 7—10 Oct. 2007. P. 2372—2377.
3. Corke P., Detweiler C., Dunbabin M., Hamilton M., Rus D., Vasilescu I. Experiments with Underwater Robot Localization and Tracking // IEEE International Conference on Robotics and Automation. 2007. P. 4556—4561.
4. Saez J. M., Hogue A., Escolano F., Jenkin M. Underwater 3D SLAM through Entropy Minimization // Proceedings 2006 IEEE International Conference on Robotics and Automation, ICRA. 2006. P. 3562—3567.
5. Бобков В. А., Борисов Ю. С. Восстановление траектории подводного аппарата и структуры среды по изображениям // Мехатроника, автоматизация, управление. 2006. № 9. С. 25—31.
6. Бобков В. А., Роньшин Ю. И., Машенцев В. Ю. Построение карт глубин на множественной последовательности видов // Информационные технологии. 2012. № 6. С. 39—44.

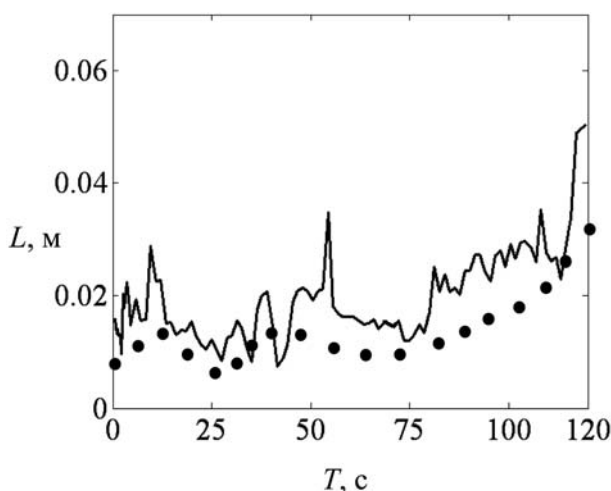


Рис. 6. Ошибка вычисления расстояний до объектов в зависимости от времени движения по траектории для первого метода (сплошная линия) и для второго метода (точки)

В. А. Бимаков, директор,
ООО "Воксельная техника", г. Ижевск,
e-mail: robint@mail.ru,
Т. Р. Касимов, инженер,
НТУ "ИТЦ" ОАО "ИМЗ "Аксион-холдинг",
г. Ижевск

Использование воксельных вычислителей для скоростного преобразования 2D- и 3D-изображений высокого разрешения

Излагаются принципы скоростного геометрического преобразования 2D- и 3D-изображений высокого разрешения с помощью воксельных вычислителей (ВВ) — цифровых устройств, в основе функционирования которых лежит принцип глубокого распараллеливания вычислительных процедур, наиболее часто используемых при обработке сцен и изображений. Метод обработки, реализуемый в ВВ, позволяет предложить новые сверхскоростные алгоритмы, время выполнения которых не зависит от геометрической формы обрабатываемых объектов.

Ключевые слова: скоростная обработка сцен и изображений, геометрические модели

1. Понятие воксельного вычислителя (ВВ), отличительные признаки ВВ

Пиксельный (иначе точечный или дискретный) метод геометрического моделирования обладает по сравнению с другими методами моделирования исключительными преимуществами по глубокому распараллеливанию основных процедур обработки сцен (ОПОС) [1]:

- теоретико-множественных операций над геометрическими моделями;
- анализа геометрических моделей на пересечаемость;
- геометрических преобразований (поворот, перенос, сжатие);
- операций вычисления объема (площади) воксельной (пиксельной) модели;
- определения положения модели в пространстве.

В полной мере реализовать эти преимущества позволяют специализированные вычислительные устройства — так называемые устройства параллельной обработки n -мерных пиксельных моделей (УОПМ) [1—5]. *Воксельный вычислитель (ВВ)* представляет собой частный случай УОПМ и предназначен для параллельной обработки двумерных (пиксельных) и трехмерных (воксельных) геометрических

моделей. Отличительными признаками ВВ являются следующие:

- в ВВ роль адресуемых информационных единиц выполняют не только числовые двоичные переменные (как в обычных числовых компьютерах), но также двумерные бинарные изображения (БИ) квадратного формата $A\sqrt{A} \times A\sqrt{A}$ и трехмерные БИ кубического формата $A \times A \times A$ (где $A = q^2$, $q \geq 2$ — целое число);
- в состав ВВ помимо обычного арифметико-логического узла, предназначенного для обработки чисел, входят аппаратные средства, позволяющие выполнять все вышеперечисленные ОПОС над пиксельными и воксельными геометрическими моделями (представленными в виде одноцветного содержимого адресуемых БИ) за время, сравнимое с элементарным тактом управления ВВ;
- система команд ВВ помимо арифметико-логических команд, используемых для обработки чисел, содержит набор команд, оперирующих адресуемыми 2D- и 3D-БИ, каждая из таких команд реализует одну ОПОС над одноцветным содержимым адресуемых БИ, этот набор является функционально полным, позволяющим выполнять весь цикл скоростной обработки сцены или изображения с использованием только воксельных и пиксельных моделей.

2. Состав воксельного вычислителя

ВВ — это многопроцессорная ОКМД-система, в которой функцию устройства управления выполняет блок программного управления (БПУ). Назначением БПУ (выполняемого по одной из известных схем) является хранение команд и числовых данных, организация их последовательного извлечения из памяти программ и данных (ППД), осуществление арифметико-логической обработки чисел, хранящихся в ППД. Однако главным компонентом ВВ, обеспечивающим рекордно высокую производительность этих устройств при обработке сцен и изображений, является блок обработки бинарных изображений (БОБИ). В состав БОБИ входят:

- *сверхоперативная RAM-память бинарных изображений (СПБИ)* для хранения адресуемых БИ (иначе АБИСП) формата $A\sqrt{A} \times A\sqrt{A}$ или формата $A \times A \times A$;
- A^3 -разрядные двоичные регистры (РгБИ) для хранения бинарных изображений-операндов, имеющих формат АБИСП и извлекаемых из СПБИ на время выполнения ОПОС;
- *теоретико-множественные преобразователи* одноцветного содержимого БИ-операндов (ТМПБИ);
- *геометрические преобразователи* одноцветного содержимого БИ-операнда (ГПБИ);
- *вычислитель значения меры (ВМБИ)* одноцветного содержимого БИ-операнда (понятие меры

- $\mu^{(n)}$ является n -мерным обобщением понятий длины ($\mu^{(1)}$), площади ($\mu^{(2)}$) и объема ($\mu^{(3)}$) [6];
- *формирователь* (ФКЗБИ) числового кода (КЗБИ), интегрально описывающего характер заполнения БИ-операнда одноцветным содержимым;
- *буферная память* (БПБИ) для временного (сверхоперативного) хранения участка обрабатываемого БИ-операнда;
- *оперативная RAM-память бинарных изображений* (ОПБИ) для хранения адресуемых БИ, формат которых представляет некоторую часть АБИСП-формата (строку, столбец, слой, группу столбцов, группу слоев и т. п.) и имеет общее название "АБИОП-формат". Емкость ОПБИ является наращиваемой и значительно превышает общую емкость СПБИ.

Первые три компонента БОБИ вместе могут быть представлены в виде единой процессорной матрицы, состоящей из A^3 однотипных процессоров (ОПр), связанных сетью межпроцессорных связей и обладающих собственной локальной памятью [1]. В процессорной матрице одному пикселю-квадрату 1×1 (или вокселю-кубику $1 \times 1 \times 1$) в АБИСП соответствует один ОПр. Локальные памяти всех ОПр матрицы ПрМ образуют вместе СПБИ.

3. Бинарное составное изображение и его представление в ВВ

Максимальное распараллеливание ОПОС достигается в ВВ только при обработке БИ, имеющих АБИСП-формат (в работах [1, 2] этот формат называется "формат F^m "). Число вокселей $1 \times 1 \times 1$ (или пикселей 1×1), входящих в АБИСП-фрагмент, ограничено возможностями элементной базы ПрМ (ее стоимостью, числом и размером применяемых кристаллов, плотностью размещения электронных компонентов на кристалле и т. п.) и ограничено в настоящее время десятками тысяч вокселей (пикселей). В то же время разрешение реальных 2D- и 3D-изображений, встречающихся в большинстве прикладных задач, значительно превышает это число (1 млн и более). В ВВ такие изображения высокого разрешения представляются несколькими АБИСП-фрагментами.

Двумерное БИ, формат которого образован объединением нескольких взаимно непересекающихся АБИСП-фрагментов $A\sqrt{A} \times A\sqrt{A}$, будем называть *двумерным бинарным составным изображением* (2D-БСИ) [2]. Трехмерное бинарное изображение, формат которого образован объединением нескольких взаимно непересекающихся адресуемых АБИСП-фрагментов $A \times A \times A$, будем называть *трехмерным бинарным составным изображением* (3D-БСИ) [2]. Оба вида БСИ обрабатываются в ВВ последовательно (АБИСП-фрагмент за АБИСП-фрагментом).

Полутонные и цветные изображения высокого разрешения (так называемые полутонные и цвет-

ные составные изображения) представляются в ВВ в виде нескольких БСИ того же разрешения и формата [2]. Каждый i -й пиксель j -го БСИ должен иметь значение (0 или 1), равное значению j -го разряда двоичного кода яркости, принадлежащего i -му пикселю полутонного составного изображения.

Очевидно, чем меньше число АБИСП-фрагментов входит в БСИ, тем выше скорость обработки данного БСИ. Поскольку это число зависит от $A = q^2$, то для увеличения скорости обработки БСИ необходимо использовать ВВ с большим значением q , при этом стоимость, габаритные размеры, энергопотребление ВВ возрастают, а тактовая частота ВВ уменьшается.

- БСИ, обрабатываемые в ВВ, могут различаться: числом АБИСП-фрагментов, входящих в БСИ (т. е. разрешением БСИ);
- пространственным положением АБИСП-фрагментов внутри БСИ.

Кроме того, одно и то же БСИ на разных этапах обработки может по-разному отображаться внутри ВВ, т. е. иметь различные формы внутреннего машинного представления. Выбор формы представления БСИ определяется способом хранения информации внутри ВВ. Поскольку адресуемые информационные единицы накопителей ВВ (СПБИ, ППД, ОПБИ, внешней памяти) различаются, то и форма машинного представления БСИ на разных этапах обработки БСИ в ВВ также будет различной.

Так как емкость СПБИ не всегда позволяет разместить все БСИ высокого разрешения в СПБИ, выполнение ОПОС над содержимым таких БСИ средствами ВВ предполагает активное использование ОПБИ для хранения исходных, промежуточных и окончательных результатов выполнения ОПОС.

4. Геометрическое преобразование содержимого АБИСП-фрагмента в ВВ

Особенностью ВВ является то, что в этих вычислителях геометрическое преобразование (ГП) одноцветного содержимого БИ моделируется с помощью дискретизирующих отображений (ДО) [1, 2]. Каждое ДО моделирует не одно ГП, а бесконечное

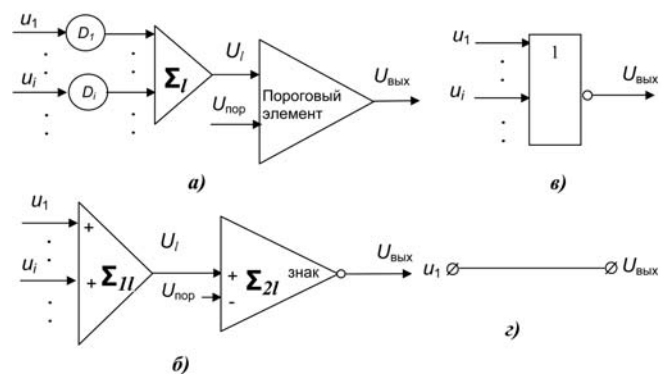


Рис. 1. Способы аппаратной реализации дискретизирующих отображений в ВВ

множество ГП, при выполнении которых любые тела, синтезируемые при этих ГП, имеют одинаковые воксельные (пиксельные) модели-образы. Функционально полный набор ДО, аппаратно реализованный в БОБИ, предназначен для преобразования содержимого только отдельных АБИСП-фрагментов. Принцип аппаратной реализации этих ДО (с максимальным распараллеливанием) описан в [1, 3, 4] и основывается на использовании:

- пар "многовходовый аналоговый сумматор — пороговый элемент" (рис. 1, а);
- пар "многовходовый цифровой параллельный сумматор — двухвходовой цифровой параллельный сумматор" (рис. 1, б);
- многовходовых схем ИЛИ (рис. 1, в);
- проводных электрических связей (рис. 1, г).

5. Геометрическое преобразование содержимого БСИ в ВВ

Высокая скорость моделирования ГП над одноцветным содержимым БСИ достигается в ВВ за счет глубокого распараллеливания:

- ДО над содержимым АБИСП-фрагментов;
- теоретико-множественных операций над содержимым АБИСП-фрагментов;
- процедур записи (чтения) АБИСП-фрагментов в (из) СПБИ и ОПБИ.

Платой за высокую скорость моделирования ГП является снижение (в общем случае) точности выполнения ГП. Этот недостаток ДО не позволяет моделировать все множество преобразований, относящихся к одному виду ГП, путем многократного выполнения одного ДО (например, моделировать поворот на угол m° путем m -кратного моделирования поворота на 1°). Конечно, при любом способе

моделирования ГП в цифровом устройстве выполнение цепочки (композиции, суперпозиции) ГП приведет (в общем случае) к потере точности моделирования ГП. Но в случае ДО допустимая длина такой цепочки крайне мала и, как правило, не превышает 2...3.

Однако существуют ГП, которые моделируются ДО без погрешности дискретизации (*дискретно-инвариантные геометрические преобразования — ДИГП*) или даже вообще без какой-либо погрешности (*форматные дискретно-инвариантные геометрические преобразования — ФДИГП*) [1].

5.1. Моделирование дискретно-инвариантных ГП над типовыми БСИ

Принцип аппаратно-программной реализации поворота 2D-БСИ и 3D-БСИ в ВВ представлен на рис. 2 и 3.

В качестве типовых БСИ [2] удобно использовать кубические БСИ формата $kA \times kA \times kA$, а также квадратные БСИ формата $kA\sqrt{A} \times kA\sqrt{A}$ (где k — целое число), так как эти форматы имеют наибольшее число ФДИГП, как-то:

- повороты на углы, кратные 90° , вокруг взаимно ортогональных осей Ox , Oy , Oz , проходящих через центр куба $kA \times kA \times kA$ параллельно его граням (рис. 3, а);
- ν -шаговые переносы (где ν — число, кратное A) в направлении осей Ox , Oy , Oz ;
- преобразования симметрии относительно плоскостей, проходящих через грани кубов-фрагментов $A \times A \times A$;
- повороты на углы, кратные 90° , вокруг центра формата $kA\sqrt{A} \times kA\sqrt{A}$;
- w -шаговые переносы (где w — число кратное $A\sqrt{A}$) в направлении осей $O_{\Pi}X_{\Pi}$, $O_{\Pi}Y_{\Pi}$ (см. рис. 2).

ФДИГП над указанными типовыми БСИ реализуется наиболее просто и состоит из следующих шагов:

- выполнение одноименного ФДИГП-преобразования над одноцветным содержимым каждого АБИСП-фрагмента данного БСИ средствами БОБИ;
- переадресация преобразованных АБИСП-фрагментов внутри СПБИ (или переадресации АБИОП-фрагментов внутри ОПБИ).

Чуть сложнее выполняются в ВВ S -шаговые ДИГП (где S — целое число, не кратное A) над содержимым БСИ форматов $kA \times kA \times kA$ и $kA\sqrt{A} \times kA\sqrt{A}$. Наряду с переадресацией адресуемых фрагментов внутри СПБИ (или ОПБИ) S -шаговые переносы предполагают обмен частью содержимого между отдельными АБИСП-фрагментами исходного БСИ.

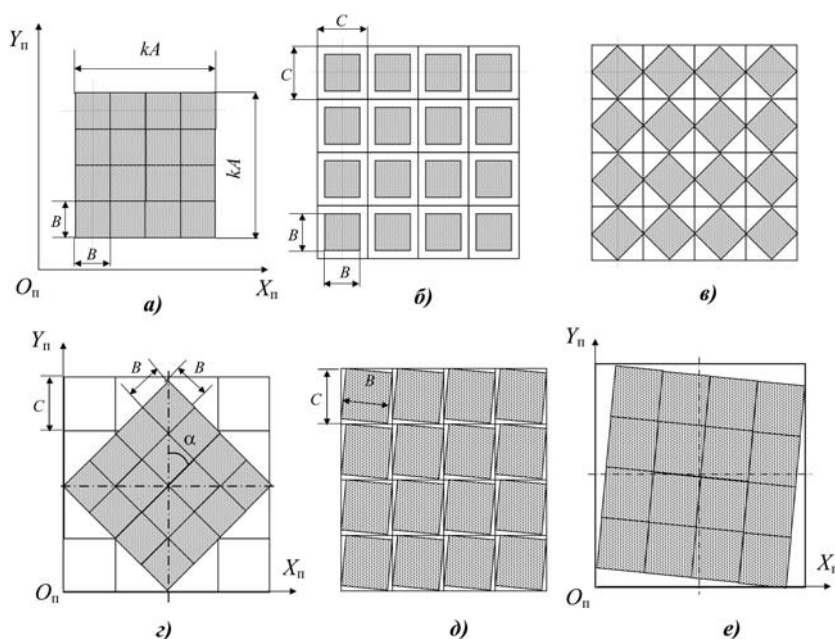


Рис. 2. Принцип аппаратно-программной реализации поворота 2D-БСИ в ВВ

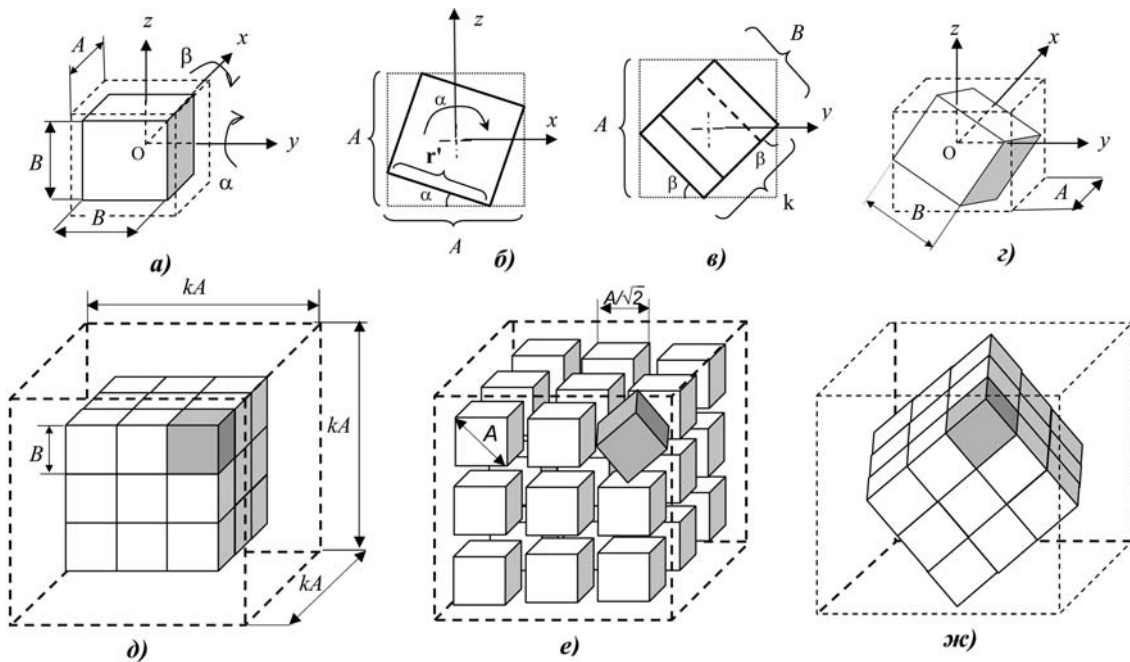


Рис. 3. Принцип аппаратно-программной реализации поворота 3D-БСИ в ВВ

5.2. Моделирование дискретно-неинвариантных ГП над типовыми БСИ

Наиболее сложно в ВВ моделируются дискретно-неинвариантные ГП [1] (ДНГП) над содержимым типовых БСИ. Принцип реализации ДНГП рассмотрим на примере моделирования ГП-поворота.

В общем случае погрешность моделирования ГП дискретизирующим отображением вызвана двумя причинами: погрешностью дискретизации и выходом содержимого преобразованного БИ за пределы АБИС-фрагмента [1, 2]. Чтобы подробно рассмотреть вопросы, связанные с погрешностью, вызванной второй причиной, реальное БИ удобно заменить его идеальной абстракцией — бинарным изображением бесконечного разрешения (БИБР).

Моделирование ДНГП-поворота содержимого 2D-БСИ. Повернем на угол α квадратное БИБР формата $C \times C$ (где C — действительное число) вокруг центра этого квадратного изображения. Часть точек повернутого БИБР $C \times C$ выйдет за границы исходного (неповернутого) квадрата $C \times C$. Если выделить из БИБР $C \times C$ его центральный квадратный участок $B \times B$ (см. рис. 2, а), где $B = C/(\cos\alpha + \sin\alpha)$, и повернуть его вокруг центра БИБР $C \times C$ на угол α , точки повернутого фрагмента $B \times B$ не выйдут за пределы $C \times C$.

Для предотвращения искажений, вызванных выходом содержимого БСИ за границы фрагментов, процедура ДО, моделирующая ДНГП над одноцветным содержимым БСИ, должна включать в себя три последовательных этапа:

- 1) расшивку исходного БСИ;
- 2) геометрическое преобразование содержимого фрагментов расшитого БСИ;

3) сшивание преобразованных фрагментов расшитого БСИ в единое результирующее БСИ.

Пусть имеется БСИ бесконечного разрешения (БСИБР) квадратного формата $kC \times kC$, состоящее из k^2 взаимно непересекающихся адресуемых БИБР-фрагментов (АБИБР) $C \times C$. Выделим из БСИБР $kC \times kC$ его центральную часть — БСИБР $kB \times kB$, состоящее из k^2 взаимно непересекающихся БИБР-фрагментов $B \times B$ со сторонами, параллельными сторонам квадрата $kA \times kA$ (см. рис. 2, б).

Под процедурой *расшивки* БСИБР $kB \times kB$ понимается процесс переноса БИБР-фрагментов $B \times B$ из БСИБР $kB \times kB$ в БСИБР $kC \times kC$, в котором центр i -го (где $i = 1, \dots, k^2$) фрагмента $B \times B$ совпадает с центром i -го фрагмента $C \times C$, а ребра квадратов-фрагментов $C \times C$ и $B \times B$ параллельны или взаимно ортогональны (см. рис. 2, в). Так как $B = C/(\cos\alpha + \sin\alpha)$, то при $\alpha \rightarrow 0^\circ$ следует $B \rightarrow C$, а площадь покрытия квадрата $kC \times kC$ квадратом $kB \times kB$ стремится к 100%. В ВВ аппаратно реализуются повороты на углы только из диапазона $0^\circ < \alpha \leq 45^\circ$, поэтому B не может быть меньше $A/\sqrt{2}$ (см. рис. 2, г).

Повернем на угол α содержимое каждого БИБР-фрагмента $C \times C$, полученного в результате расшивки БИБР $kB \times kB$, вокруг центра этого БИБР-фрагмента. При этом повернутся и БИБР-фрагменты $B \times B$, образующие центральную часть фрагментов $C \times C$ (см. рис. 2, в, д). После поворота k^2 фрагментов $B \times B$ выполним процедуру *сшивания* (сборки) этих k^2 БИБР-фрагментов $B \times B$ в одно общее повернутое $kB \times kB$. Сшивание должно обеспечивать целостность сшиваемого БИБР $kB \times kB$ и правильность его углового положения, отличающегося от исходного на угол α .

При шивании повернутого БИБР $kB \times kB$ в качестве реперных точек удобно использовать центры шиваемых фрагментов $B \times B$, совпадающие с центрами поворота этих квадратов-фрагментов. Пусть O — центр декартовой системы координат Oxy , одновременно являющийся центром БИБР $kB \times kB$, а x_i, y_i, z_i — координаты центра i -го БИБР-фрагмента $B \times B$.

Разница между описанными процедурами и их одноименными аналогами, реализуемыми в ВВ над содержимым 2D-БСИ, заключается в том, что:

- в ВВ значения C и B — целые числа;
- разрешающая способность БСИ (в отличие от БИБР) конечна;
- выполнение поворотов содержимого фрагментов $C \times C$ вокруг центров этих фрагментов заменено моделированием поворотов с помощью ДО, выполняемых над плоскими АБИСП-фрагментами формата $A\sqrt{A} \times A\sqrt{A}$.

Моделирование ДНГП поворота содержимого 3D-БСИ. Повернем БИБР $C \times C \times C$ вокруг его центральных осей Oy и Ox (рис. 3, а, б, в, г) на два угла Эйлера (α, β — углы нутации и прецессии). Часть точек повернутого БИБР $C \times C \times C$ выйдет за границы исходного БИБР $C \times C \times C$. Если выделить из БИБР $C \times C \times C$ его центральную кубическую часть $B \times B \times B$ (см. рис. 2, а), где $B = C/(\cos\beta + (\cos\alpha + \sin\alpha)\sin\beta)$, и повернуть его вокруг центра БИБР $C \times C \times C$ сначала на α , а затем на β , то точки повернутого БИБР $B \times B \times B$ не выйдут за пределы $C \times C \times C$.

Пусть имеется составное БСИБР $kC \times kC \times kC$, состоящее из k^3 взаимно непересекающихся кубических фрагментов $C \times C \times C$. Выделим из этого БСИБР его центральную часть — БСИБР размером $kB \times kB \times kB$, состоящее из k^3 взаимно непересекающихся БИБР-фрагментов $B \times B \times B$ с ребрами, параллельными ребрам куба $kA \times kA \times kA$ (рис. 3, д).

Под процедурой *расшивки* БСИБР $kB \times kB \times kB$ будем понимать процесс переноса фрагментов $B \times B \times B$ из БСИБР $kB \times kB \times kB$ в БСИБР $kC \times kC \times kC$, в котором центр i -го (где $i = 1, \dots, k^3$) фрагмента $B \times B \times B$ совпадает с центром i -го фрагмента $C \times C \times C$, а ребра кубов-фрагментов $C \times C \times C$ и $B \times B \times B$ или взаимно параллельны или взаимно ортогональны между собой (рис. 3, е). Так как $B = C/(\cos\beta + (\cos\alpha + \sin\alpha)\sin\beta)$, то при $\alpha \rightarrow 0^\circ$ и $\beta \rightarrow 0^\circ$ следует $B \rightarrow C$, а объем покрытия куба $kA \times kA \times kA$ кубом $kC \times kC \times kC$ стремится к 100%. В ВВ аппаратно реализуются повороты на углы только из диапазонов $0^\circ < \alpha \leq \arcsin(\sqrt{2}/\sqrt{3}) \approx 54,74^\circ$ и $0^\circ < \beta \leq 45^\circ$, поэтому B не может быть меньше $A/\sqrt{3}$.

Повернем на углы α и β содержимое каждого БИБР-фрагмента $C \times C \times C$, полученного в результате расшивки БСИБР $kB \times kB \times kB$, вокруг центра этого БИБР-фрагмента (рис. 3, е, з). При этом повернутся и БИБР-фрагменты $B \times B \times B$, образующие центральную часть фрагментов $C \times C \times C$. После поворота k^3 фрагментов $B \times B \times B$ выполним процедуру *шивания* (сборку) этих k^3 БИБР-фрагментов $B \times B \times B$ в одно общее повернутое $kB \times kB \times kB$. Шивание должно обеспечивать целостность шиваемого БИБР $kB \times kB \times kB$ и правильность его углового положения, отличающегося от исходного на углы α и β .

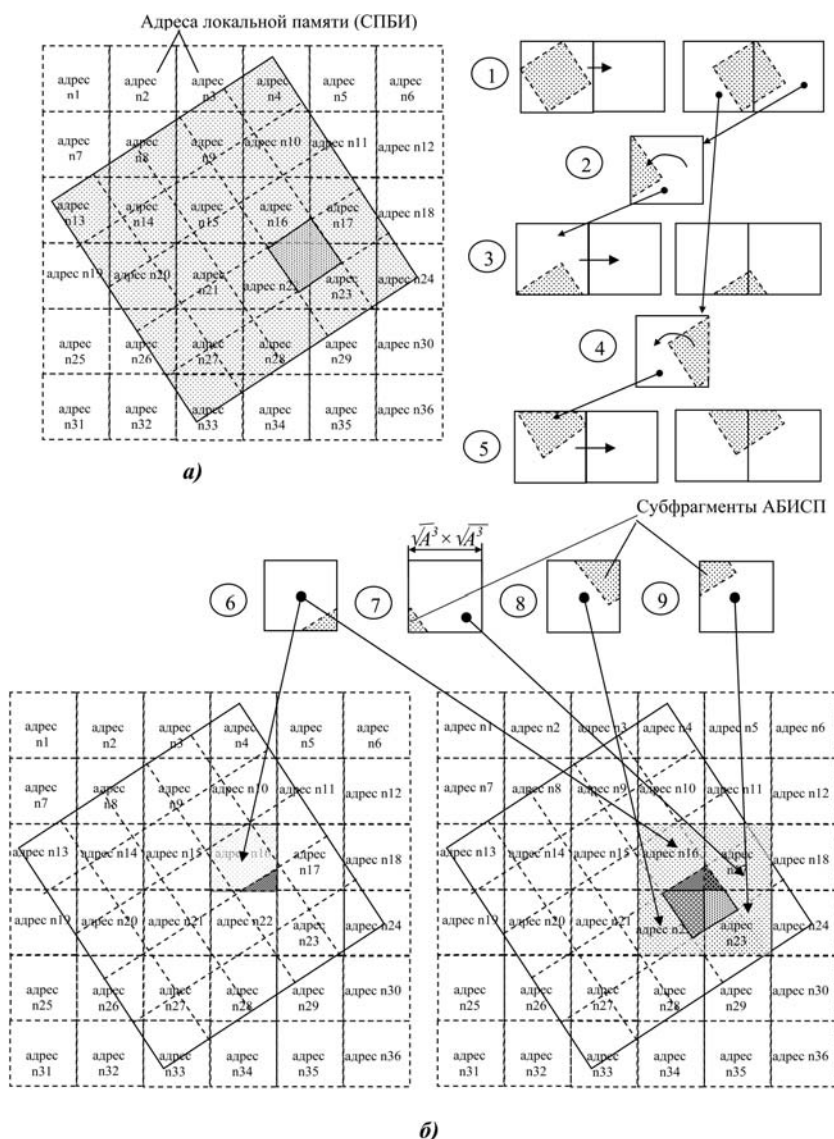


Рис. 4. Модификация содержимого адресуемых ячеек СПБИ при повороте 2D-БСИ

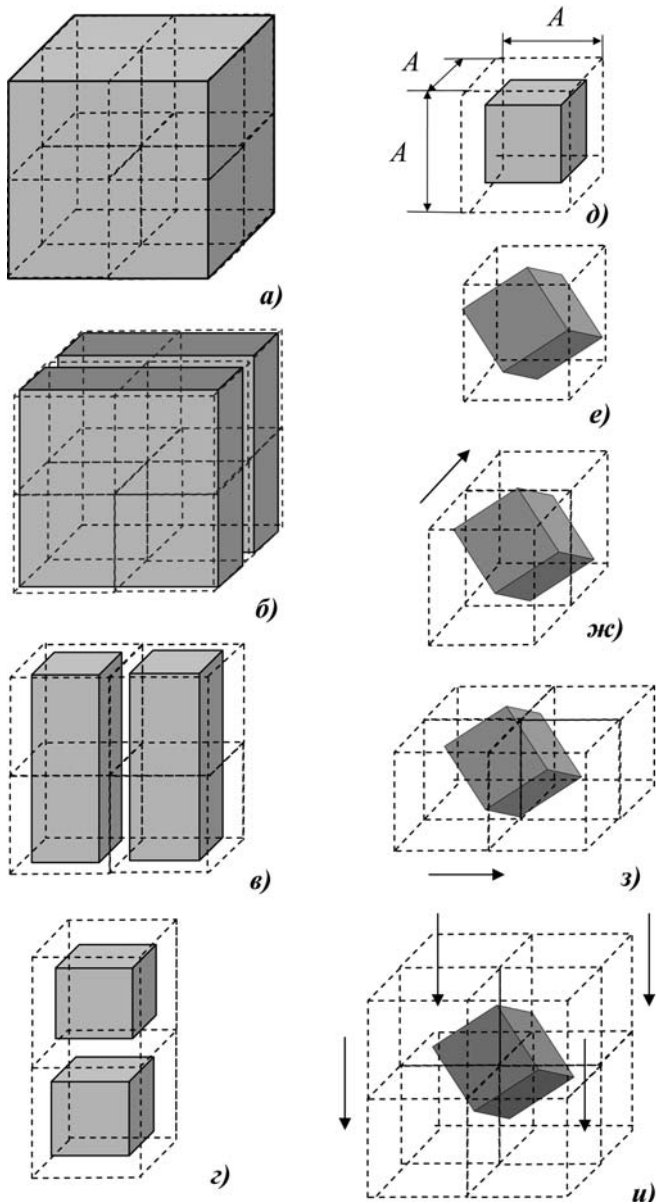


Рис. 5. Модификация содержимого адресуемых ячеек СПБИ при повороте 3D-БСИ

Разница между описанными процедурами и их одноименными аналогами, реализуемыми в ВВ над содержимым 3D-БСИ, заключается в том, что:

- в ВВ значения C и B — целые числа;
- разрешающая способность БСИ (в отличие от БИБР) конечна;
- выполнение поворотов содержимого фрагментов $C \times C \times C$ вокруг центров этих фрагментов заменено моделированием поворотов с помощью ДО, выполняемых над трехмерными АБИСП-фрагментами формата $A \times A \times A$.

Использование СПБИ при скоростных преобразованиях БСИ. Процесс скоростного преобразования БСИ непосредственно связан с процессом

преобразования содержимого СПБИ. Каждому БСИ в СПБИ выделяется свое адресное пространство, в котором адреса АБИСП-фрагментов жестко увязываются с координатным положением АБИСП-фрагментов внутри БСИ (рис. 4, а). Принцип модификации содержимого БСИ в процессе сверхоперативного преобразования БСИ продемонстрируем на примере ГП-поворота.

Перед поворотом БСИ расширяется на отдельные АБИСП-фрагменты (см. рис. 2, б, рис. 5, а—д). Затем содержимое каждого фрагмента расшитого БСИ поворачивается (см. рис. 2, в, рис. 5, е) и сдвигается (рис. 4, б, рис. 5, ж—и) с помощью аппаратно-реализованных ДО.

В СПБИ для сшивания повернутых фрагментов выделяются адресуемые ячейки $A \times A \times A$ ($A\sqrt{A} \times A\sqrt{A}$), в которых будет размещено результирующее (повернутое) БСИ. Перед выполнением процедуры сшивания эти ячейки обнуляются, после чего повернутые фрагменты расшитого составного изображения сшиваются в единое результирующее БСИ (см. рис. 2, г, е; рис. 3, ж).

Для сшивания каждый повернутый фрагмент $A\sqrt{A} \times A\sqrt{A}$ (см. рис. 4, б) расчленяется в общем случае на четыре субфрагмента. Каждый повернутый трехмерный фрагмент $A \times A \times A$ (рис. 5, е) расчленяется в общем случае на восемь субфрагментов (рис. 5, и). Каждому субфрагменту для его размещения в СПБИ присваивается свой адрес (из выделенных ячеек СПБИ), соответствующий пространственному положению этого субфрагмента в результирующем БСИ (см. рис. 4, б). По каждому такому адресу может размещаться несколько субфрагментов результирующего БСИ, сформированных ранее. Перед записью каждого субфрагмента (в соответствующую ячейку СПБИ) предыдущее содержимое этой ячейки извлекается из СПБИ и "склеивается" (теоретико-множественной операцией "объединение") с вновь устанавливаемым субфрагментом. После склейки синтезированный фрагмент $A \times A \times A$ записывается вновь в ту же самую ячейку.

Список литературы

1. Бимаков В. А. Высокая скорость обработки n -мерных пиксельных геометрических моделей // Информационные технологии. 2009. № 3. С. 29—36.
2. Бимаков В. А., Бимаков Е. В. Принципы построения и функционирования устройств для высокоскоростной обработки n -мерных пиксельных геометрических моделей // Информационные технологии. 2012. № 6.
3. Бимаков В. А., Бимаков Е. В. Устройство обработки двумерных и трехмерных изображений. Пат. РФ № 2289161. Бюл. № 34 от 10.12.2006.
4. Бимаков В. А., Бимаков Е. В. Устройство обработки двумерных и трехмерных изображений. Пат. РФ № 2376637. Бюл. № 35 от 20.12.2009.
5. Бимаков В. А., Бимаков Е. В. Устройство управления автономным роботом. Пат. РФ № 2424105. Бюл. № 20 от 20.07.2011.
6. Колмогоров А. Н., Фомин С. В. Элементы теории функций и функционального анализа. М.: Наука, 1981.

УДК 004.3

А. Б. Барский, д-р техн. наук, проф.,
e-mail: arkbarsk@mail.ru,
Нгуен Ван Лой, аспирант,
e-mail: loimiit@gmail.com,
МГУПС (МИИТ)

Оценка среднего времени выполнения запроса к многосерверной базе данных с циркулирующими сегментами

С помощью математической модели для фиксированных значений параметров определяется зависимость среднего времени выполнения запроса к многосерверной базе данных с циркулирующими сегментами от интенсивности потока запросов. Демонстрируется практическая возможность обслуживания в допустимое время потоков запросов значительно большей интенсивности.

Ключевые слова: многосерверная база данных, циркуляция сегментов, интенсивность потока запросов, интенсивность потока обслуживания, среднее время выполнения запроса

Введение

Современные сетевые базы данных (БД) представляют собой одноканальные системы массового обслуживания (с отказом или без), где запросы анализируются единственной СУБД (системой управления базой данных).

Из теории массового обслуживания известно, что в одноканальных системах с бесконечной очередью среднее время выполнения одного запроса находится как

$$T_{\text{вып}} = \frac{t_{\text{вып}}}{1 - \rho}, \quad (1)$$

где $t_{\text{вып}}$ — "чистое" время выполнения запроса; ρ — известное отношение λ/μ , λ — интенсивность потока запросов, $\mu = 1/t_{\text{вып}}$ — максимально возможная интенсивность обслуживания.

При возрастании λ , в частности, при подключении к серверу все новых пользователей наблюдается эффект $\rho \rightarrow 1$, что приводит к экспоненциальному росту значения $T_{\text{вып}}$.

Выход из данного положения кроется в превращении одноканальной системы массового обслуживания в многоканальную [1–5], с возможностью одновременного параллельного выполнения многих запросов.

В работе с базами данных интерес представляет не просто сервис, а оптимизация информационного

обслуживания. Тогда в качестве критерия оптимизации логично выбрать *минимум среднего времени выполнения одного запроса*.

На данном этапе исследований ограничимся рассмотрением баз данных с простыми запросами, где при выполнении одного запроса не формируются новые запросы. Такие БД характерны, например, для систем продажи билетов. БД со сложными структурированными запросами рассмотрены в работе [6].

1. Математическая модель

Разделим БД на m независимых сегментов, которые в режиме ротации размещены на разных связанных между собой серверах, снабженных копиями СУБД (рис. 1, см. третью сторону обложки). Эти серверы образуют отдельные каналы обслуживания. Пусть n — число таких серверов. Целесообразно выполнение соотношения $m \geq n$. Распределим пользовательские рабочие станции между серверами так, чтобы сократить интенсивность потока запросов к одному серверу до величины λ/n . Заставим серверы регулярно, циклически (τ — длительность такта) обмениваться сегментами БД в выбранном направлении такой ротации так, чтобы в целом обеспечить доступность всех данных каждому пользователю. Попутно будет решена и проблема синхронизации взаимного исключения обращения пользователей к одному сегменту.

Циркуляция сегментов может быть синхронной и асинхронной.

При синхронной циркуляции сегмент может находиться на сервере в течение всей длительности такта τ , даже если он этим сервером не востребован.

При асинхронной циркуляции, если сегмент поступает на сервер, не сформировавший к нему запрос, сегмент немедленно передается следующему серверу.

В то же время как при синхронной, так и асинхронной циркуляции сегмент может задерживаться на сервере более чем на один такт — до окончания выполнения запроса к нему, в соответствии с образовавшейся на сервере очередью.

В данной статье рассматривается синхронная обработка запросов.

Если на некоторый сервер поступил запрос к БД, а требуемого сегмента в данный момент в памяти сервера нет, то среднее "расстояние" — среднее время следования сегмента к данному серверу составляет $\tau n/2$. Однако на пути следования сегмента к данному серверу этот сегмент может потребоваться "попутному" серверу, ведь интенсивность обращения к одному сегменту равна λ/m . Считая, что в среднем половина всех возможных обращений к сегменту должны быть обслужены перед выпол-

нением исследуемого запроса, найдем оценку числа $r_{\text{поп}}$ "попутных" обращений к данному сегменту:

$$r_{\text{поп}} = \frac{\lambda}{2m} \cdot \tau \frac{n}{2} = \frac{\lambda \tau n}{4m}. \quad (2)$$

Тогда при условии, что $t_{\text{вып}}$ требует нескольких тактов, общее время ожидания сегмента

$$t_{\text{ожид}} = \tau n / 2 + t_{\text{вып}} \cdot r_{\text{поп}}. \quad (3)$$

Если производительность сервера такова, что запрос к сегменту обслуживается в такте поступления этого сегмента, то вторым слагаемым в выражении (3) следует пренебречь.

Общее время выполнения запроса с учетом ожидания сегмента в результате его циркуляции

$$T_{\text{вып}}^* = T_{\text{вып}} + t_{\text{ожид}}. \quad (4)$$

Здесь $T_{\text{вып}}$ определяется для одного канала обслуживания — для одного сервера, при интенсивности потока λ/n . Значение μ остается прежним: $\mu = 1/t_{\text{вып}}$.

Следовательно, параллельный многоканальный доступ к БД с помощью СУБД, расположенных на каждом сервере и в совокупности реализующих *SPMD*-технологии, приводит к следующей оценке среднего времени выполнения запроса в потоке обслуживания по одному каналу:

$$T_{\text{вып}}^* = \frac{t_{\text{вып}}}{1 - \frac{\lambda}{n\mu}} + \tau \frac{n}{2} + t_{\text{вып}} \frac{\lambda n}{4m}. \quad (5)$$

Эта оценка является окончательной, характеризующей среднее время выполнения запроса пользователя к многосерверной БД с циркулирующими сегментами.

Представленные соотношения определяют математическую, вероятностную модель. В ней не отражены "накладные расходы" на управление ротацией, не учтено время обмена данными, когда сервер недоступен и т. д.

При предоставлении ресурсов Центром *Grid*-технологий необходимо учесть широкий диапазон требований пользователя к проектируемой им базе данных, упрощающих реализацию.

В перспективе, прием объединения *Web*-серверов в единую систему с обеспечением циркуляции информации между ними [1] позволил бы обеспечить высокую доступность информации и сократить среднее время поиска в сети Интернет. Однако это характерно лишь для таких запросов к БД, которые не выполняют новые записи в сегменты, т. е. не изменяют их содержание. В сложных структурированных запросах, сопровождающихся внесением изменений в БД, новый запрос может формироваться в рамках выполнения предыдущего. Именно это на глобальном уровне характерно при работе в сети Интернет. Однако редко вносимые в этом случае изменения практически не требуют сложной синхронизации, так как возникающие противоречия разрешаются в рамках специфики приложений (например, социальных баз данных), т. е. не рассматриваются как конфликты.

Выполнение сложных запросов нельзя исключать и внутри локальных сетей: возможности сетевых БД должны быть универсальными. Начало исследований в этой области положила работа [6].

2. Пример расчета среднего времени выполнения запроса к БД на одном сервере

Рассмотрим БД, имеющую следующие варианты λ -характеристик обслуживания: $\lambda = 10, 20, 30, 40, 45, 50$ запросов в секунду.

Пусть $\mu = 50$ ($t_{\text{вып}} = 0,02$ с).

Для данного ряда значений λ найдем ряд значений $\rho = 0,2, 0,4, 0,6, 0,8, 0,9, 1$. По формуле (1) находим шесть значений $T_{\text{вып}}$ ($T_{\text{вып}} = 0,025, 0,033, 0,05, 0,1, 0,2, \infty$) и для данных значений параметров построим график (рис. 2, кривая I).

3. Время выполнения запроса к многосерверной БД не превышает длительности такта

Ту же БД реализуем как многосерверную с циркулирующими сегментами. Пусть число серверов $n = 8$, число независимых сегментов $m = 20$, длительность такта работы системы $\tau = 0,1$ с.

Учитывая появившиеся возможности, продолжим шкалу исследуемых значений λ до значения $\lambda = 160$.

Так как время выполнения запроса $t_{\text{вып}}$ существенно меньше длительности такта, то (без учета возникновения большой очереди к одному серверу) запрос выполняется за время обязательного нахождения сегмента на сервере, с которого он инициирован пользователем. То есть запрос выполняется сразу же по прибытии сегмента на этот сервер. Тогда поправка $r_{\text{поп}}$ в выражении (3) на время попутного обращения к этому сегменту не учитывается. Необходимо учитывать лишь среднее время движения запрашиваемого сегмента к серверу, инициировавшему запрос. Тогда среднее время выполнения запроса

$$T_{\text{вып}}^* = \frac{t_{\text{вып}}}{1 - \frac{\lambda}{n\mu}} + \tau \frac{n}{2}. \quad (6)$$

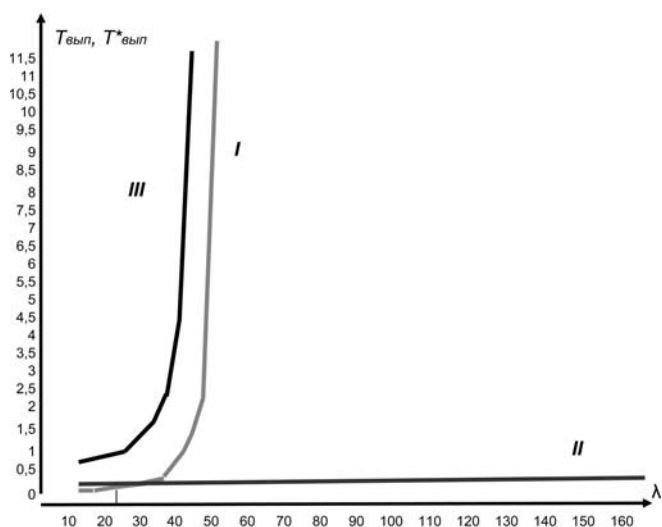


Рис. 2. Зависимость среднего времени выполнения запросов к БД от их интенсивности

Найдем $T_{\text{вып}}^*(10) = 0,42$; $T_{\text{вып}}^*(20) = 0,421$; $T_{\text{вып}}^*(30) = 0,4217$; $T_{\text{вып}}^*(40) = 0,4222$; $T_{\text{вып}}^*(50) = 0,423$; $T_{\text{вып}}^*(80) = 0,425$; $T_{\text{вып}}^*(160) = 0,433$ с. График зависимости $T_{\text{вып}}^*$ от λ для тех же значений других характеристик представлен на том же рис. 2, кривая II.

Можно судить о почти линейной зависимости среднего времени $T_{\text{вып}}^*$ от λ в рассматриваемой области при заданных значениях параметров БД с циркулирующими сегментами. Но самый главный вывод следует сделать такой: область применения многосерверной базы данных успешно развивается в область высоких значений интенсивности запросов, характерных для всех сетевых БД, включая сеть Интернет. Вид же нового графика соответствует начальной, пологой части графика, составленной для односерверной, традиционной БД.

4. Время выполнения запроса превышает длительность такта

Рассмотрим пример тех же значений характеристик многосерверной базы данных при условии $t_{\text{вып}} > \tau$. Тогда имеет смысл поправка $r_{\text{поп}}$ в выражении (3).

Для $t_{\text{вып}} = 0,2$ с ($\mu = 5$) и для каждого значения λ по (5) найдем среднее время выполнения запросов $T_{\text{вып}}^*(\lambda)$. Получим ряд значений: $T_{\text{вып}}^*(10) = 0,86$, $T_{\text{вып}}^*(20) = 1,2$; $T_{\text{вып}}^*(30) = 1,5$; $T_{\text{вып}}^*(40) = \infty$.

Кривая III полученной для рассмотренного примера зависимости также представлена на рис. 2. По-видимому, слишком сильно занижено значение μ . Система "испортилась". Это говорит о необходимости такой сбалансированности системы серверов, при которой производительность одного сервера (одного канала обслуживания запросов к БД) допускает высокое значение его характеристики μ .

Следует отметить очевидные преимущества такой производительности серверов, при которой в одном такте обрабатывается очередь многих запросов, и вероятность задержки их выполнения мала.

Последнее замечание характеризует основной недостаток исследований многосерверной БД с помощью математической модели. Математическая модель требует значительных допущений. Выше мы пренебрегли обработкой очереди запросов на каждом сервере. Это противоречит большому числу m сегментов.

Заключение

Исследуется математическая модель процесса обслуживания запросов к многосерверной базе данных с циркулирующими независимыми сегментами. Такие базы данных предполагают взаимно исключаящее обращение к сегментам, так как каждое обращение изменяет их содержание. Это характерно, например, для систем продажи билетов различного назначения — от театральных касс до авиарейсов.

В этих системах БД легко разбивается на независимые сегменты: по категориям, направлениям, отдельным маршрутам, обслуживаемым объектам.

Следует отметить, что данная схема БД с циркулирующими сегментами между серверами, с которыми жестко связаны рабочие станции пользователей, не соответствует какой-либо из "классических" схем, которые, например, изучаются в теории массового обслуживания. Данная схема требует своей "математики". Построение математической модели практически всегда основано на серьезных допущениях, которые присущи оперированию со случайными величинами. Абстрактный расчет вероятностных оценок зачастую разнится даже у одного автора. Формулы постоянно уточняются при необходимости рассмотрения новых факторов и возможных событий.

Разработчик системы все более убеждается в том, что если нет возможности поставить эксперимент, то необходимо провести *детерминированное имитационное моделирование*, воссоздав реальную систему на компьютере и изучив ее на основе *стохастического* задания исходных данных.

Таким образом, цель данной работы — приближенный анализ интуитивно чувствующихся преимуществ предлагаемой схемы на основе традиционных вероятностных оценок.

Самый главный вывод настоящей статьи заключается в том, что возможен эффективный прорыв на значительно большую, во много крат, интенсивность запросов к сетевой многосерверной базе данных, при которой ответ может быть получен в приемлемые сроки.

Поскольку в данном случае математические расчеты оказываются более оперативными (менее трудоемкими), а результаты имитационного моделирования более достоверны, с помощью калибровки и юстировки следует провести уточнение математических выражений, "подогнав" их под результаты экспериментов. Это послужит оперативному нахождению оптимальных характеристик проектируемых многосерверных баз данных с многоканальным параллельным доступом, использующих принцип циркуляции сегментов между серверами.

Однако более перспективным представляется подход, когда на основе имитационной модели насчитывается достаточное число вариантов (ситуаций) для различных наборов параметров. Затем множество полученных отношений {ситуация → среднее время выполнения запроса} записывается в виде обученной по ним логической нейронной сети [7]. Образуется информационно-справочная система. По ней обеспечивается точная интерполяция для получения среднего времени выполнения запроса по любой испытываемой ситуации. Работа с такой системой крайне нетрудоемка. Она легко добавляется новыми знаниями для повышения точности и расширения области применения и модифицируется.

Список литературы

1. Барский А. *Grid*-вычисления. Организация, методы, планирование. LAP LAMBERT Academic Publishing. 2012.
2. Барский А. Б. Параллельные информационные технологии: учеб. пособие. М.: ИНТУИТ; БИНОМ. Лаборатория знаний. 2007. 503 с.
3. Барский А. Б. Параллельные информационные технологии в основе *Grid*-системы // Информационные технологии. 2006. № 12. С. 54–60.

4. Загорский Г. С., Барский А. Б., Желенков Б. В. GRID-технологии — ресурсы супервозможностей // Мир транспорта. 2008. № 3.
5. Барский А. Б. Оптимизационные задачи в основе пакета параллельных прикладных программ и системы информационного обслуживания Центра GRID-технологий // Информационные технологии. 2010. № 10. Приложение. 32 с.
6. Желенков Б. В. Оптимизация доступа к ресурсам баз данных // Компьютеры в учебном процессе. 2005. № 6.
7. Барский А. Нейронные сети логического вывода. Курс лекций. LAP LAMBERT Academic Publishing. 2011.

УДК 004.652.6

С. В. Власов, аспирант,
e-mail: vlasovsv@gmail.com,
ОАО "Газпром промгаз"

Сущностно-ориентированный подход к разработке информационного хранилища (в задачах транспорта газа)

Рассмотрены классический подход к формированию структуры базы данных (БД) на основе нормализации данных и сущностно-ориентированный подход на основе описания метаинформации объектов предметной области. Описаны плюсы и минусы применения классического подхода, даны основные предпосылки перехода от классической структуры БД к структуре на основе метаинформации данных, рассмотрены условия применения каждого из методов.

Ключевые слова: реляционная база данных, структура БД, сущности, нормализация, версияльность

Введение

В работе над крупными проектами, такими как разработка Генеральных схем газоснабжения и газификации регионов РФ, задействовано большое число специалистов различного профиля. Информация, необходимая специалистам для формирования качественных и обоснованных технических решений, зачастую не структурирована, а ее источники децентрализованы.

ОАО "Газпром промгаз" — головной научно-исследовательский институт компании ОАО "Газпром" — выполняет работы по разработке и актуализации генеральных схем газоснабжения и газификации регионов РФ. Генеральная схема представляет собой систему взаимосвязанных технических решений по развитию региональных систем газоснабжения в увязке с развитием топливно-энергетического комплекса региона.

Для обеспечения более эффективной работы над проектами ОАО "Газпром промгаз" необходимо

единое информационное хранилище, объединяющее всю накопленную и полученную в ходе работ над проектами информацию в единое информационное пространство.

Основные термины и определения

Сущность — абстракция множества объектов реального мира, удовлетворяющих следующим требованиям:

- объекты множества имеют одни и те же характеристики (далее свойства);
- все объекты подчинены и согласованы с одним и тем же набором правил и линий поведения.

Экземпляр сущности — это конкретный объект (реализация сущности), удовлетворяющий требованиям этой сущности.

ГРС — газораспределительная станция.

МГ — магистральный газопровод.

ПХГ — подземное хранилище газа.

Для хранения, актуализации и обработки информации необходима база данных (БД), удовлетворяющая следующим требованиям:

- хранение большого числа сущностей и их свойств;
- легкое добавление новых сущностей и их свойств без изменения структуры БД;
- версияльность данных по экземплярам сущности и их свойствам (возможность просмотра среза состояний объекта за любую дату в прошлом);
- хранение хронологических данных за большой период времени;
- хранение связей между различными сущностями (включая связи "один ко многим" и "многие ко многим");
- интеграция данных из сторонних хранилищ.

Рассматривая задачу выбора реализации хранения произвольной неструктурированной информации в реляционной БД, можно увидеть два различных подхода:

- *классический подход* (список сущностей и их свойства заранее известны);
- *сущностно-ориентированный подход* (сущности и их свойства неизвестны на этапе проектирования структуры БД и могут быть добавлены во время работы БД без изменения ее структуры).

Классический подход

Классический подход эффективен в том случае, если сущности и их свойства заранее известны на этапе проектирования структуры БД и в дальнейшем меняться не будут. Этот метод определяет для каждой сущности отдельные таблицы, в которых будут храниться экземпляры сущности. Пример такого подхода приведен на рис. 1.

Как видно из рис. 1, классический подход формирует для каждой сущности свою таблицу в реляционной БД, каждая сущность имеет свой уникальный набор свойств, представленных столбцами таблицы сущности и таблицы данных, связанной с таблицей сущности. При этом данные, которые не зависят от времени (паспортные данные), хранятся в таблице сущности, остальные свойства сущности, зависящие от времени, хранятся в отдельной

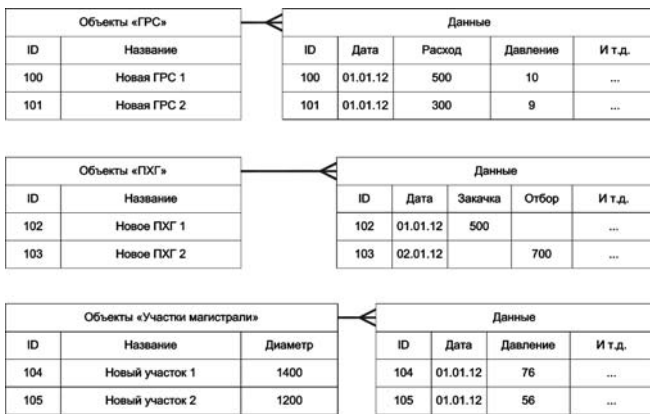


Рис. 1. Классический подход к формированию структуры базы данных

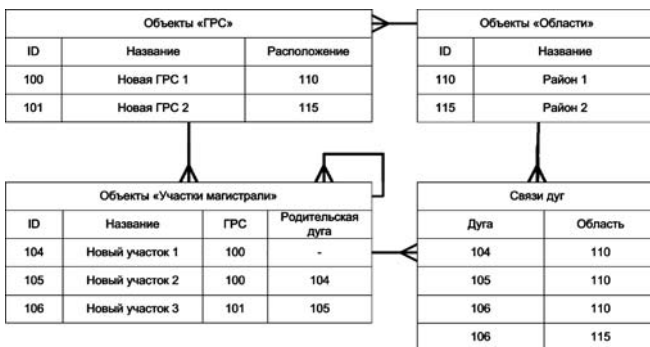


Рис. 2. Связи между объектами в классическом подходе



Рис. 3. Структура хранения версионных данных

таблице с временными данными. Таким образом, чтобы хранить в такой структуре N сущностей, требуется создать в реляционной базе данных $2N$ таблиц. Такой подход прост и хорош только в том случае, когда состав свойств сущности и число сущностей заранее известны и не изменяются во времени. Структура БД, построенная по классическому подходу, не является по сути масштабируемой или расширяемой, так как какое-либо добавление новых сущностей или свойств сущности влечет за собой изменение во внутренней структуре БД, включая изменение программных единиц, ответственных за чтение, запись и обработку данных.

Связи между объектами (рис. 2) хранятся в виде столбцов в таблицах сущностей как внешние ключи на экземпляры других сущностей. Также существует возможность задания связей сущности самих на себя (иерархические связи). Таким образом, достигается целостность данных.

Возьмем для примера связь "один ко многим" между ГРС и участком магистрали. В таблице сущности "Участки магистрали" добавляется новый столбец "ГРС", который является внешним ключом на объект таблицы "ГРС". Причем к одной ГРС может быть привязано несколько участков и для любой ГРС можно получить список дуг, технологически связанных с ней. Точно так же можно описать свойство ГРС "Расположение", указывающее на субъект РФ, в котором она находится.

Для добавления связи между экземплярами одной сущности (например, ссылка на "Родительскую дугу" участка магистрали) необходимо в таблицу сущности вставить новый столбец "Родительская дуга" в качестве внешнего ключа на ту же таблицу. Таким образом, можно хранить иерархию объектов и для каждого из них получить как родительскую дугу, так и список дочерних дуг для текущего участка магистрали.

Для связей "многие ко многим" в структуру базы данных необходимо добавить дополнительную таблицу-развязку, которая будет хранить внешние ключи на объекты связанных сущностей. Уникальность связи будет определяться комбинацией связанных экземпляров сущностей. Таким образом, для формирования связей "многие ко многим" между экземплярами сущностей на каждый тип связи необходимо добавить еще по одной таблице в структуру базы данных.

Добавление версионности в классическом подходе подразумевает создание отдельной таблицы с версионными данными для каждой сущности. Как уже было описано выше, паспортные данные хранятся в таблице сущности в виде отдельных столбцов. Если паспортные данные будут меняться, то они должны быть вынесены в таблицу версионных данных. Подобно неизменяемым данным, версионные данные хранятся в виде соответствующих столбцов с той лишь разницей, что в таблицу добавляются два дополнительных столбца: дата начала и дата окончания, определяющие время актуальности версионных данных (рис. 3).

При добавлении версионных данных в качестве даты начала указывается дата фактического изменения одного из атрибутов, входящих в версионную

таблицу. В качестве даты окончания указывается произвольная максимальная дата (в примере это 1 января 2010 г.). Таким образом, формируется интервал актуальности набора значений версионных свойств. В случае последующих изменений находится интервал, в который входит дата начала актуальности свойств, и разбивается на два непересекающихся интервала: предыдущая дата начала до текущей даты начала и от текущей даты начала до даты конца исходного интервала. В случае совпадения дат начала обновляется запись, связанная с найденным интервалом.

Как видно из вышесказанного, добавление версионности хоть и позволило хранить историю изменений свойств экземпляра сущности, но при этом добавило дублирование данных, которые не подверглись изменению, и привело к удвоению числа таблиц сущностей. Нормализация данных в классическом подходе позволяет четко разделить сущности на атомарные таблицы, обеспечить целостность и непротиворечивость данных. Нормализации данных посвящена работа [1]. Но дополнительные условия, накладываемые на структуру БД, приводят к формированию огромного числа таблиц и последующему дублированию информации, что впоследствии затрудняет администрирование, поддержку и разработку программных единиц в БД, построенной по классическому методу.

Сущностно-ориентированный подход

Сущностно-ориентированный подход предлагает иную реализацию структуры БД. Идеи сущностно-ориентированного подхода рассматриваются в работе [2]. Структура БД, построенной по этому принципу, состоит из нескольких частей (моделей):

- модель метайнформации;
- модель данных;
- модель версий.

Модель метайнформации. Каждая сущность (таблица "Сущность") (рис. 4), хранится в общей таблице сущностей, уникальность записи определяется кодом сущности. Каждая сущность описывается набором свойств, характеризующих эту сущность.

Свойства сущности (таблица "Свойства сущности") формируются из глобального списка свойств

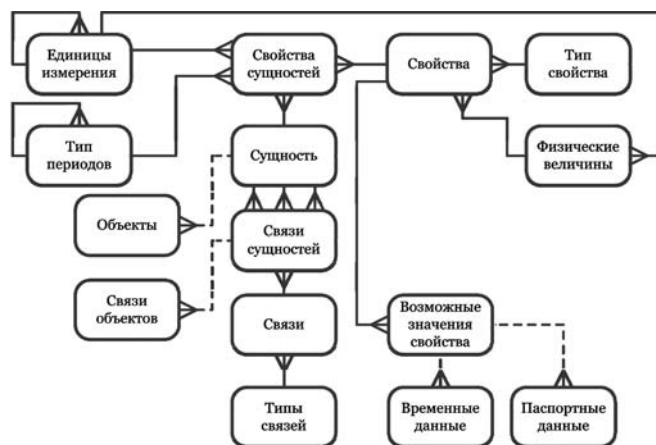


Рис. 4. Структура модели метайнформации

(таблица "Свойства"). Глобальные свойства описываются кодом свойства, описанием свойства, ссылкой на тип свойства и ссылкой на физическую величину, в которой хранится это свойство. В свою очередь тип свойства (таблица "Тип свойства") описывает тип данных (числовой, строковый, массив, перечисление и др.), в котором хранится свойство и методы (программные единицы БД) получения этого свойства. Такой подход позволяет реализовать интеграцию с данными из внешних источников. Например, можно описать тип свойства "Внешнее свойство" и указать SQL-запрос, с помощью которого можно получить данные по этому свойству из внешнего источника информации. Под внешним источником понимается реляционная БД, доступная из текущей БД.

Физическая величина (таблица "Физические величины") представляет собой реальную физическую величину, такую как:

- давление;
- расход;
- температура;
- скорость;
- длина и т. д.

Физическая величина описывается уникальным идентификатором, описанием физической величины и ссылкой на базовую единицу измерения (таблица "Единицы измерения"), в которой она хранится. Связка свойства с физической величиной добавлена для сравнения различных свойств, представляющих собой одну и ту же физическую величину, между собой.

Как уже было сказано выше, свойства сущности формируются на основе глобального списка свойств. Каждое свойство сущности описывается уникальным идентификатором и наименованием свойства. Свойства сущности содержат внешний ключ на глобальное свойство, размерность, в которой оно будет храниться и период данных (таблица "Тип периодов"), характеризующий время актуальности свойства (например, годовые, суточные, часовые данные).

Таким образом, можно описать одно свойство "Расход" и получить несколько реализаций этого свойства в экземплярах сущностей, например, суточный, годовой, часовой расход по ГРС. При этом каждая реализация этого свойства будет храниться в своей размерности и иметь свой период актуальности, но описываться одной и той же физической величиной — "Расход".

Таблица "Единицы измерения" представляет собой иерархический справочник единиц измерения, в котором указываются взаимосвязи между различными единицами измерения и коэффициенты перевода из одной единицы измерения в другую.

Таблица "Тип периодов" описывает возможные периоды времени, характеризующие актуальность значения свойства сущности.

Модель данных. В отличие от классического подхода в сущностно-ориентированном подходе для хранения значений свойств экземпляров сущностей используется списочное представление (рис. 5).

Экземпляры сущности (реальные объекты) представлены уникальным идентификатором, перечень их хранится в общей таблице "Объекты". Каждый объект имеет наименование, внешний ключ на сущность и номер версии объекта, что позволяет легко группировать объекты по сущностям или находить нужный объект по имени.

Значения свойств объектов сущностей хранятся в вертикальных таблицах с разделением на следующие типы: паспортные версионизируемые данные, хронологические данные и данные массивов.

Таблица паспортных данных "Паспортные данные" хранит значения свойств экземпляров сущностей, которые не привязаны к временным рамкам. Для всех таких данных предусмотрена возможность версионирования. Таблица "Паспортные данные" хранит внешний ключ на свойства сущности, внешний ключ на экземпляр сущности и номер версии данных. Значение свойства хранится в специальном объектном типе, позволяющем сохранять и манипулировать данными в виде строки, числа, даты и т. д.

Таблица "Хронологические данные" хранит список хронологических данных с указанием фактической даты изменения свойства. Подобно паспортным данным в таблице имеются ссылки на свойства сущности, внешний ключ на экземпляр сущности и столбец с объектным типом для хранения значения свойства.

Таблица "Данные массивов" хранит линейный список данных, привязанных к одному свойству экземпляра сущности. Организация хранения данных в этой таблице похожа на организацию хранения данных в таблице "Паспортные данные" с тем

лишь отличием, что добавляется новое поле "Номер в массиве", указывающее на текущий индекс значения свойства в общем массиве значений.

Модель версий. Модель версий в сущностно-ориентированном подходе в отличие от классического подхода не требует добавление новых таблиц для версионных данных (рис. 6).

Структура модели состоит из нескольких таблиц, обеспечивающих полную поддержку версионности паспортных данных. Таблица "Версии данных" содержит уникальный идентификатор версии, ее наименование, внешний ключ на источник данных (таблица "Источник данных"), из которого были получены данные, и отчетную дату создания версии.

Таблица "Источник данных" является справочной таблицей и хранит перечень источников, из которых информация попадает в БД. Каждый источник данных представлен уникальным идентификатором и описанием источника.

Также модель версий содержит таблицу "Лог изменений", в которой фиксируются все изменения данных, происходящие в БД. Таблица "Лог изменений" содержит внешний ключ на таблицу "Тип изменения", в которой описаны типы изменений данных: вставка, удаление и обновление. С помощью лога изменений можно восстановить всю последовательность действий, выполняемую над массивом данных, хранящимся в структуре БД.

Заключение

В работе были рассмотрены два различных подхода к организации хранения разнотипной информации. Были проанализированы основные требования, накладываемые на структуру БД единого информационного хранилища ОАО "Газпром промгаз".

В результате рассмотрения двух подходов стало понятно, что сущностно-ориентированный подход лучше применим для формирования масштабируемых информационных систем, в которых заранее неизвестен состав сущностей и их свойств, которые будут храниться в БД, в то время как классический подход хорош для разработки небольших систем, в которых состав сущностей и их свойств заранее определен и не изменяется со временем.

Поэтому при разработке единой информационной системы ОАО "Газпром промгаз" был использован сущностно-ориентированный подход. Данная система находится в промышленной эксплуатации и активно применяется при разработке Генеральных схем газоснабжения и газификации регионов РФ для оптимизации региональных систем газоснабжения и расчетов распределительных газопроводов.

Список литературы

1. Энсор Д., Стивенсон Й. Oracle. Проектирование баз данных // Пер. с англ. К.: Издат. группа BHV, 1999. С. 86—96.
2. Константинова И. М., Дубинский А. В., Дубровский В. В., Сиперштейн Б. И., Фридман В. Е. Математическое моделирование технологических объектов магистрального транспорта газа. М.: Недра, 1988. С. 46—48.



Рис. 5. Модель данных



Рис. 6. Модель версий

ЖУРНАЛ В ЖУРНАЛЕ



**НЕЙРОСЕТЕВЫЕ
ТЕХНОЛОГИИ**

№ 3
МАРТ
2013

Главный редактор:

ГАЛУШКИН А.И.

Редакционная коллегия:

АВЕДЬЯН Э.Д.
БАЗИАН Б.Х.
БЕНЕВОЛЕНСКИЙ С.Б.
БОРИСОВ В.В.
ГОРБАЧЕНКО В.И.
ЖДАНОВ А.А.
ЗЕФИРОВ Н.С.
ЗОЗУЛЯ Ю.И.
КРИЖИЖАНОВСКИЙ Б.В.
КУДРЯВЦЕВ В.Б.
КУЛИК С.Д.
КУРАВСКИЙ Л.С.
РЕДЬКО В.Г.
РУДИНСКИЙ А.В.
СИМОРОВ С.Н.
ФЕДУЛОВ А.С.
ЧЕРВЯКОВ Н.И.

**Иностранные
члены редколлегии:**

БОЯНОВ К.
ВЕЛИЧКОВСКИЙ Б.М.
ГРАБАРЧУК В.
РУТКОВСКИЙ Л.

Редакция:

БЕЗМЕНОВА М.Ю.
ГРИГОРИН-РЯБОВА Е.В.
ЛЫСЕНКО А.В.
ЧУГУНОВА А.В.

Артеменков С. Л.

Иерархия процессов опознания в "перцептроне" Миракяна . . . 56

Пекунов В. В.

О классификации лиц методом голосования с нейросетевым арбитром. Распараллеливание вычислений на многоядерных видеокартах 61

Дорогов А. Ю., Абатуров В. С., Раков И. В.

PMML-модели быстрых нейронных сетей 65

С. Л. Артеменков,
канд. техн. наук, проф., зав. лаб.,
Московский городской
психолого-педагогический университет,
e-mail: start@inbox.ru

Иерархия процессов опознания в "перцептроне" Миракяна

Представлена реконструкция общего строения модели "перцептрона" Миракяна, которая основана на реализации принципов трансцендентальной психологии и осуществляет "кодопорождение идеи (имени) и формы" объектов. Дано описание некоторых важных особенностей и функциональных свойств этой модели.

Ключевые слова: Миракян, модель, перцептрон, форма, имя, кодирование, восприятие, трансцендентальная психология

Введение

Модель, которую условно можно назвать "перцептроном" Миракяна, представляет собой специальную систему, построенную на основе структурно-процессуального образования отношений и разработанную проф. А. И. Миракяном для демонстрации гипотетических качественных характеристик и принципиальных особенностей процесса восприятия. Прототип одной из частей этой модели был реализован более 20 лет тому назад в виде электронно-технического устройства и отчасти описан в работах [1, 2]. Однако полные данные о строении, вариантах и особенностях модели не были опубликованы, и в этой связи данная модель в целом остается неизвестной для широкой научной общественности. Вместе с тем модель представляет не только исторический, но и общий научно-практический интерес, поскольку ее строение основано на фундаментальных научных принципах и методологии теории трансцендентальной психологии [3–5]. Развитие этой теории, на наш взгляд, может открыть пути к построению новой технологии моделирования порождающих процессов и создания на этой основе полифункциональных искусственных перцептивных устройств. В настоящей статье на основе опыта личного сотрудничества автора с А. И. Миракяном принята попытка реконструировать общую структуру модели "перцептрона" и описать некоторые ее особенности и важные функциональные свойства.

Принципы трансцендентальной психологии

Научно-исследовательская деятельность проф. Миракяна была посвящена проблеме познания закономерностей и механизмов процессов непосредственно-чувственного отражения (восприятия).

Проведя исторический анализ подходов к исследованию этой проблемы, Миракян обнаружил, что непосредственные свойства образов, полагаемые естественными продуктами этих процессов, исследователи широко используют как в объяснении процессов восприятия, так и при поиске закономерностей этих процессов. Этому способствовал естественно сложившийся продуктивный образ мышления [6], при котором исследователь имеет дело "с готовыми образами восприятия и представления, понятиями и т. п., но отнюдь не с порождающими и не с формирующими их психическими процессами" [7].

Продуктивный подход такого рода (когда о скрытом от наблюдения процессе судят по свойствам входных и выходных продуктов этого процесса) имеет место в любых науках и часто является вполне адекватным. Миракян, однако, поставил революционный вопрос об адекватности продуктного подхода в психологии восприятия [8, 9]. Мотивировал он это тем, что процессы психического отражения являются порождающими процессами. Восприятие по сути определяется, как процесс порождения чувственных образов, возникающих при условии воздействия внешнего мира на органы чувств. Вместе с тем логично предположить, что закономерности порождающих процессов не должны быть основаны на свойствах тех компонентов или продуктов, которые взаимодействуют или возникают в результате действия этих процессов.

Требование преодоления продуктного подхода в психологии обусловило ту задачу, которую поставил проф. Миракян при исследовании непосредственно-чувственного процесса отражения. Она состояла в том, чтобы найти принципы и механизмы, которые не используют заранее постулированные, уже отраженные эмпирические свойства объектов и обеспечивают возможность порождения образа (формы) любой фигуры в различных условиях восприятия. Эти поиски в результате потребовали коренного пересмотра традиционной методологии и серьезной смены научных оснований.

В разработанном Миракяном новом подходе трансцендентальной психологии в качестве исходной аксиоматики для объяснения и моделирования процессов восприятия используются фундаментальные системно-процессуальные принципы, обуславливающие возможность понимания как сущности процессов порождения форм объектов (формопорождения), так и природы существования самих перцептивных процессов [8, 9]. Эти принципы были выявлены Миракяном на основе общего анализа особенностей строения перцептивных систем живых существ и в общем виде опосредованно характеризуют качественные свойства процесса порождения психического отражения, т. е. не на основе эмпирических свойств его продуктов [4, 6, 10, 11].

По Миракяну процессуальные условия существования и действия отражательной системы в живых организмах следующие [6]: дискретизация форм пространственности и непрерывности времени; образование и фиксация отношений между дискретными элементами пространственной формы в определенном моменте времени и отношений между зафиксированными данными различных моментов времени. Общая структура перцептивных процессов определяется принципом анизотропной (дискретной) структурно-процессуальной организации отражательной системы, необходимой для формопорождения. Элементарный механизм формопорождения определяется принципом образования пространственно-временных, и, в частности, так называемых симметрично-двуединых отношений [4, 6, 10]. Симметрия, в частности, известна как ключевое биологическое свойство, которое может играть существенную роль в определении формы объектов [12].

Схематически симметрично-двуединое отношение показано на рис. 1. Сигналы с двух однородных дискретных элементов $A1$ и $A2$, симметричных относительно некоторой оси S , проходят через элементы $B1$ и $B2$, где возможна временная задержка и фиксация этих сигналов. Далее они в элементе C образуют отношение сравнения, результат которого используется для осуществления в системе обратной связи.

Для случая использования бинарной сигнальной логики (0 и 1 — отсутствие и наличие сигнала) образование симметрично-двуединых отношений можно рассматривать как фиксацию несимметричных входных воздействий в элементах $B1$, $B2$ и результата логической операции "И" в элементе C . Элементы $B1$, $B2$ повторяют значения сигналов с выхода элементов A , но имеют свою память. При этом образованное отношение в C по обратной связи приводит к "стиранию" зафиксированных ранее в $B1$, $B2$ сигналов. Общее поведение элементов определяется таблицей, где показаны состояния элементов $B1$, $B2$ и C (после срабатывания обратной связи) при соответствующих значениях сигналов $A1$ и $A2$.

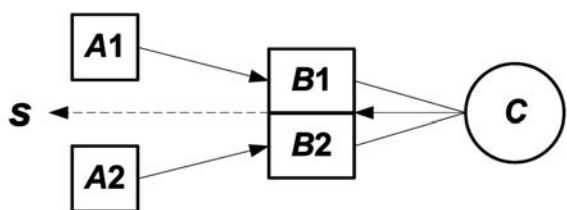


Рис. 1. Схема симметрично-двуединого отношения

Таблица истинности симметрично-двуединого элемента

№	1	2	1	2	1	2	1	2
$A1, A2$	0	0	0	1	1	0	1	1
$B1, B2$	0	0	0	1	1	0	0	0
C		0		0		0		1

В целях исследования возможностей гипотетических принципов и закономерностей психического отражения Миракян предложил устройства "некибернетических "перцептронов", моделирующих процессы порождения психических новообразований" [7].

Функциональные возможности "перцептрона" Миракяна

Предложенный Ф. Розенблаттом в 1957 г. и реализованный в виде электронной машины перцептрон стал одной из первых моделей нейросетей [13]. При этом исследования автора перцептрона были, главным образом, связаны не с изобретением устройств, обладающих "искусственным интеллектом", а с изучением нейродинамических принципов. В этом отношении предложенную Миракяном модель (хотя она существенным образом отличается от традиционного перцептрона Розенблатта) условно тоже можно назвать перцептроном, поскольку он призван продемонстрировать работу предложенных принципов трансцендентальной психологии восприятия.

На основе развитого нового подхода Миракяном были инициированы работы по моделированию процессов восприятия в искусственных системах и созданию соответствующих технических устройств перцептронного типа [1, 6]. Общая модель восприятия, которую предложил Миракян, имеет иерархическое строение и реализует процессы, включающие как восходящее кодирование сначала формы, а затем идеи или имени объекта, так и обратный нисходящий процесс обратного воссоздания кода формы объектов от кода их имени. Процесс восприятия в модели является как минимум двухступенчатым и двунаправленным. По Миракяну искусственная система отражения в принципе должна быть способна осуществлять ответ как на вопрос (А) об опознании объекта (как объекта имеющего данное имя), так на вопрос (Б) об идентификации формы объекта (например квадрата).

А. Вопрос: Что это такое? Например, объект этой формы (с этим кодом) условились называть квадрат. При этом важно, что соответствующий первичный процесс кодирования определяет и выделяет единичный объект среди множества объектов, первично как некоторый паттерн общего кода формы и затем как некоторую свертку — "идею" (нумен или имя) объекта. Сравнение этого имени с содержанием памяти обеспечивает ответ на поставленный вопрос.

Б. Вопрос: Этот ли это объект (например, квадрат ли это)? Ответ на этот вопрос предполагает встречный процесс, идущий от фиксированного в памяти имени квадрата к объекту, находящемуся вовне, от представления к объекту. Этот процесс является "идеальным" в том смысле, что здесь происходит обратное порождение от "точки" фиксации памяти и встреча с тем процессом, который идет извне, что обеспечивает подтверждение того, что приходящее

извне на орган чувств воздействие по форме действительно является таковым. По Миракяну у животных ответ на представленный выше вопрос Б всегда "да", т. е. для них вопрос "Этот ли это объект?" не имеет смысла, поскольку возникновение этого вопроса связано с необходимостью идеального выбора объекта по уже оформленному имени или идеальному представлению, которое выступает в качестве заместителя актуального объекта восприятия.

Таким образом, отражение в "перцептроне" Миракяна есть встреча двух процессов: восходящего порождающего, идущего от внешнего воздействия, и обратного процесса, который является процессом порождения изнутри. В отличие от традиционных представлений важно, что эта встреча может происходить на разных уровнях и заключается в образовании отношений, которые обуславливают дальнейшие возможности взаимодействия.

Строение "перцептрона" Миракяна

Общая структурная схема "перцептрона" Миракяна может быть представлена в виде двунаправленной системы структурно-процессуального кодирования (рис. 2). Система состоит из следующих взаимосвязанных подсистем: 1 — рецептивное поле (РП); 2 — поле (блок) калибровки; 3 — структура кодирования; 4 — оперативное поле (образов); 5 — система управления перемещением и фиксацией изображения объекта; 6 — (безадресная) память; 7 — именованное поле (блок "имяпорождения"). Система в качественном смысле (реализации ее работы на основе гипотетических принципов) имитирует работу зрительного анализатора и обеспечивает как порождение представления о форме объекта, так и обратное порождение изображения объекта от его представления. Встречными стрелками на рис. 2 показаны поля, где встречаются процессы для ответов на рассмотренные выше вопросы А и Б.

Различные блоки, представленные на рис. 2, были в разной степени детализированы Миракяном и эти блоки в целом могут быть по-разному воссозданы и интерпретированы. В частности, механизмы блока 2 были представлены очень абстрактно [4]. В наибольшей степени были разработаны механизмы блока 3 [1]. В варианте представленной ниже упрощенной технической реализации "перцептрона" (см. рис. 4) рецептивное поле 1 имело вид квадратной матрицы однородных дискретных элементов и блоки 2 и 7 в нем вообще отсутствуют [1]. Рецептивное поле 1 образует поле для представления изображений объектов и имеет вид круга, составленного из однородных дискретных рецептивных элементов, расположенных по диаметрам круга симметрично относительно центра.

Поле блока калибровки 2 повторяет форму рецептивного поля и осуществляет редукцию элементов и нормализацию размера изображения объекта (И), представленного на рецептивном поле — увели-

чение до некоторой стандартной величины, представленной на поле стандартизированного изображения (СИ). По оригинальной идее Миракяна этот процесс происходит на основе пространственно-временной дискретизации так, что пространственные размерности образуются с прямым участием фактора времени [10]. Отсюда следует, что время играет ключевую роль в формировании чувства пространственной протяженности, которое порождается благодаря ощущению процессуальной длительности [4].

Структура кодирования 3 обеспечивает порождение кода формы объектов и в простейшем виде может быть построена на основе послойного образования симметрично-двуединных отношений (см. рис. 1) между дискретными элементами каждого из диаметров поля СИ. На схеме 3 рис. 3 представлена диаметрально-симметричная линия этого поля (с семью дискретными элементами), центральный элемент и двуединные элементы трех слоев симметрично-двуединных отношений. Элементы оперативного поля образов 4 и ячейки памяти 6 повторяют струк-

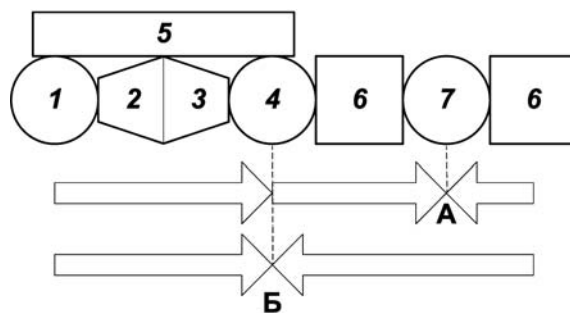


Рис. 2. Блок-схема "перцептрона" Миракяна

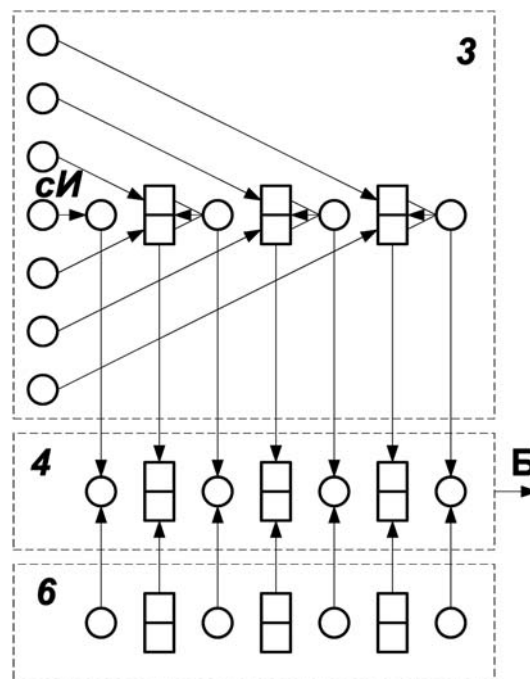


Рис. 3. Схема структуры формопорождения 3, оперативного поля образов 4 и элемента памяти 6 для одного из диаметров поля СИ

туру двуединых элементов в схеме 3. Поле 4 представляет собой поле встреч (Б) кода образа линии с соответствующим кодом образа линии, представленным в текущей памяти 6.

В случае если объект движется или его форма меняется в процессе одной фиксации объекта, обеспечиваемой блоком 5, то процесс сравнения (Б) обеспечивает также дальнейшую возможность последовательного образования отношений между различными фиксациями кодов в различные дискретные моменты времени. Такая модель образования отношений во времени между различными кодовыми паттернами была использована для верификации принципов формопорождения в психологическом эксперименте [14].

Система безадресной памяти состоит из множества слоев ячеек 6 и сохраняет диаметрально-линейную организацию. Запись осуществляется в первую ячейку, свободную от записанного кода, а чтение обеспечивается путем выявления групп ячеек, имеющих совпадения соответствующих кодов. Именное поле 7 обеспечивает далее свертку кода формы объектов в их имя и сохранение этих имен в далее расположенной памяти 6.

Система управления перемещением и фиксацией изображения объекта 5 действует на основе организации обратной связи и проводит симметризацию изображения объекта на рецептивном поле, обеспечивая как фиксацию статических, так и проследивание движущихся объектов. Устройство этой системы для структуры кодирования на основе квадратного рецептивного поля описано в работе [2].

Работа перцептрона может быть представлена следующим образом. При появлении изображения во всех элементах блока кодирования 3 проявляется код формы и этот код встречается на оперативном поле образов 4 с кодом слоя памяти 6. При этом если код формы имеет хотя бы один элемент, совпадающий по состоянию с элементом слоя памяти, то код в 4 стирается и далее код формы сравнивается со следующим слоем памяти, и так до тех пор, пока не находится свободный вышележащий слой памяти. Далее весь код переходит в это свободное поле памяти и полностью стирается в блоке кодирования. Образование затем имени формы выполняется как спиральная свертка в стек номеров всех активных точек кода поля памяти, начиная с краевых элементов рецептивного поля. Последний элемент стека записывается как имя данной формы, которое может сравниваться с другими именами (см. вариант А встречи двух процессов, показанных на рис. 2 стрелками). Такое кодирование в принципе обеспечивает возможность от одной "точки" именного поля представления обратно восстановить кодовый паттерн формы. Этот обратный процесс, в частности, происходит в перцептроне в ситуации ответа на вопрос Б (см. рис. 2).

Устройство для кодирования и фиксации форм объектов

Структура кодирования 3 была реализована в виде технического устройства, которое было названо устройством для кодирования форм объектов и зарегистрировано в виде изобретения [1]. Устройство состояло из блока преобразования входного воздействия в дискретное двухградационное изображение, представленное на квадратном рецептивном поле (РП) 8×8 элементов. Основу структуры кодирования составляли идентичные двуединые элементы, выполненные на основе стандартных логических микросхем и триггеров. Выходное кодовое поле имело четыре уровня иерархии. На рис. 4 представлена упрощенная структурная схема иерархии симметрично-двуединых соединений между двуединными элементами для случая РП 4×4 элементов. Кодовое поле устройства состоит в этом случае из трех уровней иерархии 1—3.

Форма в устройстве предстает в виде ее кода, зафиксированного в однородной иерархической структуре дискретных элементов, фиксирующих симметрично-двуединые отношения.

Двуединые элементы линий первого уровня иерархии фиксируют совместное появление единичных сигналов с пар симметричных относительно центральной горизонтальной оси элементов каждой вертикальной линии РП. Двуединые элементы второго уровня иерархии фиксируют тетрарное симметрическое возбуждение соответствующих элементов РП и организованы симметрично относительно вертикальной оси аналогично двуединным элементам линий первого уровня. Двуединые элементы третьего уровня фиксируют симметрические отношения элементов второго уровня. При большем размере РП число уровней структурной иерархии соответственно увеличивается.

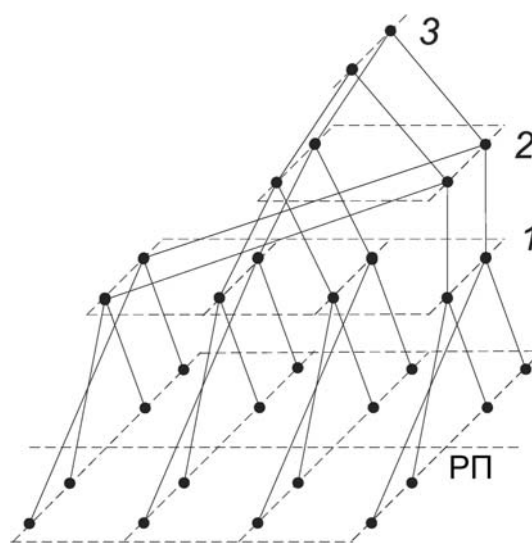


Рис. 4. Структурная схема иерархии симметрично-двуединых отношений в устройстве с квадратным РП 4×4

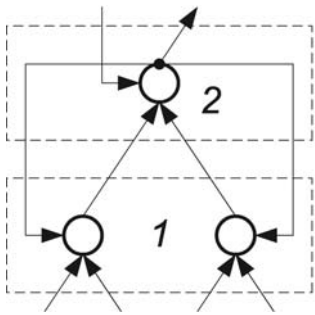


Рис. 5. Схема связей между двуединицами элементами разных уровней иерархии в структуре кодирования

Отношения между уровнями иерархии имеют не только прямые связи формирования отношений, но и обратные связи, обеспечивающие стирание зафиксированных отношений нижележащего уровня или слоя при условии образования и фиксации вышележащего отношения (см. пример для уровней 1 и 2 иерархии на рис. 5). Механизм обратных связей уменьшает общее число активных двуединиц элементов, т. е. обеспечивает редукцию единичных состояний в коде формы.

Результаты экспериментального исследования созданного технического устройства показали, что оно вместе с блоком управления перемещением РП может поддерживать фиксацию, прослеживание объектов, обеспечивает создание редуцированного кода формы объектов, а также запись и воспроизведение этого кода в структуре памяти. При этом в коде форм объектов представлен комплекс сопредставленных свойств, определенных симметричностью формы объекта. Было показано [1, 2], что с функциональной точки зрения, устройство после обучения и заполнения памяти обеспечивает: спонтанное выделение и быструю идентификацию отдельных паттернов кодов объектов; выделение совместно заданных объектов в сцене без предварительного описания свойств объектов или сцены; относительную (в пределах разрешающих возможностей устройства) устойчивость опознания объектов при определенных изменениях ориентации, положения и формы объектов в присутствии изменяющегося окружения.

Заключение

Представленный в работе "перцептрон" является реконструкцией модели проф. А. И. Миракяна, иллюстрирующей работу предложенных им трансцендентальных принципов построения формопорождающих процессов и на этой основе моделирования процессов восприятия в искусственных системах. Рассмотренная выше модель не претендует на полноту реализации свойств реальных процессов восприятия и представляет собой относительно простую аналитическую редукцию перцептивной системы до уровня гипотетических фундаментальных принципов, которые показывают процессуальные основы психического отражения и позволяют по-

иному взглянуть на организацию перцептивных процессов как процессов образования отношений. Рассмотренная модель, например, демонстрирует, как представления о репрезентации в восприятии могут быть непротиворечиво совмещены с идеей порождения образов. Механизм образования отношений в модели, с одной стороны, обеспечивает процессы новообразования — собственно порождения кода формы объектов, а с другой стороны, является основой для процессов двухуровневого репрезентативного сравнения кодов встречных процессов формопорождения (от внешнего воздействия и из памяти).

Таким образом, в первую очередь "перцептрон" Миракяна предназначен для исследования предложенных трансцендентальных психических принципов. Вместе с тем следует отметить, что Миракян возлагал на эти принципы и реализующую их модель большие надежды. Он считал, что продолжение исследований в этом направлении может привести к построению новой технологии моделирования порождающих процессов и созданию на этой основе гибко-адаптивных полифункциональных искусственных перцептивных устройств и систем опознания для разных сенсорных модальностей. К сожалению, реализовать эти планы на практике не удалось. Подробности строения модели и конструктивные схемы не были полностью опубликованы и по разным причинам работы в направлении моделирования формопорождающих процессов не были продолжены.

Следует отметить, что трудности, стоящие на пути дальнейшего развития как модельных исследований, так и всей области трансцендентальной психологии, связаны с двумя весьма существенными факторами. Во-первых, это собственно трансцендентальный характер онтологии исследуемых систем и процессов, т. е. исследуемая сфера, по сути, скрыта от непосредственного наблюдения и требует для своего изучения особой как теоретической, так и экспериментальной методологии. Во-вторых, это фактор, связанный с методологическими и психологическими проблемами, которые возникают в результате требуемого радикального изменения парадигмальных представлений. В частности, необходимость преодоления физикального (продуктного) образа мышления [15] вступает в противоречие с традиционными научными взглядами и подходами, что обуславливает то, что в целом трансцендентальный подход остается для исследователей во многом непонятным и потому мало применяемым на практике. Рассмотренная выше модель возможно послужит более ясному пониманию как механизмов действия перцептивных устройств, так и нового научного подхода, предложенного А. И. Миракяном. Важно подчеркнуть, что трансцендентальные принципы и их развитие предоставляют возможности для уточнения старых и создания новых моделей как процессов восприятия, так и других психических процессов [14—16].

Список литературы

1. **Миракян А. И.** Устройство для кодирования форм объектов // Принципы порождающего процесса восприятия / Под ред. А. И. Миракяна. М.: Бюрег, 1992. С. 47–52.
2. **Миракян А. И., Артеменков С. Л.** Устройство центрации и слежения динамического объекта // Принципы порождающего процесса восприятия / Под ред. А. И. Миракяна. — М.: Бюрег, 1992. С. 53–61.
3. **Артеменков С. Л.** Методология трансцендентальной психологии и проблемы моделирования и экспериментального исследования порождающих процессов // Труды ФИТ (2). М.: РУСАВИА, 2005. С. 37–57.
4. **Миракян А. И.** Контурсы трансцендентальной психологии. Кн. 2. М.: Изд. Института психологии РАН, 2004. 383 с.
5. **Artemenkov S. L., Harris M. G.** From Constancy & Polyfunctionality in Perception to the Transcendental Psychology approach // Journal of Integrative Neuroscience. 2005. N 4(4). P. 523–536.
6. **Миракян А. И.** Афизиальные принципы психического отражения и их моделирование. // Принципы порождающего процесса восприятия / Под ред. А. И. Миракяна. М., 1992. С. 9–46.
7. **Михайлов Ф. Т.** Креативность воображения // Перемены. 2000. № 3. С. 143–160.
8. **Миракян А. И.** Психология пространственного восприятия. Ереван: Айастан, 1990. 206 с.
9. **Миракян А. И.** Константность и полифункциональность восприятия. М., 1992. 216 с.
10. **Миракян А. И.** Начала трансцендентальной психологии восприятия // Философские исследования. 1995. № 2.
11. **Миракян А. И.** Контурсы трансцендентальной психологии. Кн. 1. М.: Изд. Института психологии РАН, 1999. 205 с.
12. **Blum H.** Biological shape and visual science // Journal of Theoretical Biology. 1973. N 38. P. 205–287.
13. **Розенблатт Ф.** Принципы нейродинамики: Перцептроны и теории механизмов мозга. М.: Мир, 1965. 480 с.
14. **Артеменков С. Л.** Метод экспериментальной проверки гипотетической модели образования симметрично-двуединных отношений в процессе зрительного восприятия. Экспериментальная психология в России. М.: Изд-во Института психологии РАН, 2010. С. 205–210.
15. **Артеменков С. Л.** Трансцендентальная психология как изменение образа мышления. А. И. Миракян и современная психология восприятия. М.: УРАО "Психологический институт"; Обнинск: ИТ-СОЦИН, 2010. С. 324–358.
16. **Панов В. И.** Экологическая психология: Опыт построения методологии. М.: Наука, 2004.

УДК 528.854::004.032.26::004.272.2

В. В. Пекунов, д-р техн. наук,
ООО "Спецлаборатория", г. Иваново

О классификации лиц методом голосования с нейросетевым арбитром. Распараллеливание вычислений на многоядерных видеокартах

Обсуждаются различные подходы к идентификации, базирующиеся на использовании метода главных компонент. Предложен новый метод идентификации, включающий применение методов наибольшей близости и k -ближайших соседей с последующим нейросетевым арбитражем. Описан подход к распараллеливанию нейросетевых расчетов на видеокартах в рамках метода обратного распространения ошибки с применением технологии OpenCL. Продемонстрировано повышение качества идентификации лиц.

Ключевые слова: идентификация лиц, метод главных компонент, нейросетевой арбитр, метод голосования, параллельный OpenCL-расчет на видеокарте

Введение

Задача классификации лиц, т. е. идентификация личности по ее изображению, полученному, в частности, с камеры видеонаблюдения, весьма актуальна для современных систем безопасности. Среди методов идентификации выделяют следующие: 1) методы, основанные на выделении и сравнении

характерных черт лица (метод эластичных графов и др. [1]); 2) методы, основанные на редукции изображения всего лица к вектору проекций данного изображения на новый специфический базис (метод главных компонент [2, 3] с последующим привлечением нейронных сетей, линейного дискриминантного анализа [2] и/или иных методов); 3) методы, предполагающие какой-либо иной подход (скрытые марковские модели [4], дискретные преобразования Фурье) к анализу отдельных фрагментов изображения лица и их комплексов. Первый класс методов весьма чувствителен к точности определения координат черт лица и иногда требует наличия цветовой компоненты изображения [5], что в общем случае, когда изображение снимается с простых камер видеонаблюдения, часто черно-белых и не дающих качественного разрешения, затруднительно. Поэтому в сфере нашего особого внимания оказался второй класс методов, в основе которого лежит надежный и хорошо формализованный метод главных компонент [3].

Задача данной работы состояла в том, чтобы построить надежный и быстродействующий классификатор лиц, представленных вектором координат в пространстве главных компонент. Было выдвинуто требование работы всех компонентов системы в реальном масштабе времени.

1. Предварительная обработка изображений лиц

Поступающие с камеры изображения лиц автоматически *центрировались* в результате работы детектора лиц (двухэтапной детекции с фильтром LBP (*Local Binary Patterns*, [6]) на первом этапе и отсеиванием результатов на втором этапе фильтром с

примитивами Хаара [7]). Далее *отсекался фон*, для этого на изображении подбирали наиболее адекватный эллипс, выделяющий исключительно лицо, опираясь на его граничные точки, определенные детектором Кенни [8]. Подбор эллипса осуществляли методом Хафа [8] с двумя параметрами: горизонтальным положением центра эллипса и его горизонтальным радиусом (вертикальный радиус равнялся половинной высоте изображения лица). "Голос" за ту или иную комбинацию этих параметров подавался, если соответствующий эллипс проходил через точку, в которой детектором Кенни [8] была отмечена граница. После отсекающего фона выполнялась *эквализация изображения* [8], позволяющая девальвировать эффект снижения качества распознавания, обусловленный различиями в условиях освещенности.

2. Основная обработка

После предварительной обработки к изображению лица применяли метод главных компонент, изображение проецировалось на базис из 45 собственных лиц, сгенерированных по базе BioID. Полученная метрика лица (вектор координат в пространстве главных компонент) подлежала идентификации по базе данных метрик лиц персон. Сразу же было принято решение отказаться от чисто нейросетевой [9] идентификации с одной сетью-классификатором по причине ее высокой сложности и, следовательно, чрезмерно высоких затрат на обучение и переобучение сети (при вводе в базу новых лиц), хотя и известно, что такой подход иногда дает весьма высокую вероятность распознавания. Однако справедливости ради следует отметить, что такой результат достигается при идентификации тех изображений лиц, которые непосредственно входят в обучающую выборку. При попытке идентификации лиц в несколько иных ракурсах (сдвинутых, наклоненных, повернутых) процент распознавания обычно уменьшается. Также мы отказались от одного из классических методов — линейного дискриминанта Фишера [2], поскольку сразу же оказалось, что ковариационная матрица часто не имеет полного ранга. Вполне возможно, что в таком случае помог бы отбор переменных (координат пространства главных компонент) с применением корреляционного анализа, но было принято более простое решение — использовать в качестве варианта линейной классификации линейный перцептрон.

В качестве возможных нейросетевых вариантов рассматривали лишь: *линейный перцептрон*¹; *классическая двуслойная нейронная сеть прямого распространения*² с первым нелинейным (экспоненциальные сигмоиды) слоем из 20...32 нейронов и од-

¹ Число входов n равно размеру вектора метрики лица. Сеть, в данном случае состоящая из одного нейрона со смещением, обучалась методом наименьших квадратов, сводящемуся к решению системы из $n + 1$ линейных алгебраических уравнений.

² Число входов равно размеру вектора метрики лица. Все нейроны имели смещение.

ним линейным нейроном во втором слое, обучение которой ведется градиентным методом (обратного распространения ошибки). Значительные надежды возлагались на *метод группового учета аргументов* (МГУА³, см. [2]) с простейшими квадратичными функциями-блоками вида

$$y(x_1, x_2) = a + b \cdot x_1 + c \cdot x_2 + d \cdot x_1 \cdot x_2 + f \cdot x_1^2 + g \cdot x_2^2,$$

который позволяет инкрементально быстро строить не менее сложные и перспективные интерполяторы, чем нейронные сети. Кроме того, рассматривались такие элементарные методы как метод *наибольшей близости* [3] и метод *k-ближайших соседей* [2], базирующиеся на расстоянии Махаланобиса.

Было принято решение провести сравнительное экспериментальное исследование всех указанных методов. В трех первых случаях для каждой персоны строился отдельный интерполятор (сеть полиномов МГУА или нейронная сеть), который обучался выделять метрики лиц данной персоны среди других. Значение выхода интерполятора принадлежало диапазону [0; 1], значение определяло степень уверенности идентификации данной персоны. В таком случае при появлении в базе данных нового лица иногда можно оперативно переопределить (переобучить) требуемое число интерполяторов за время, допустимое по требованиям решаемых нами задач. Более подробно этот вопрос будет рассмотрен далее.

3. Исследование различных методов идентификации лиц

Методы рассматривали на базе данных из 180 изображений лиц, представляющих 28 персон в различных ракурсах. Системе предъявляли изображения персон в тех же ракурсах, что и в базе, а также и в несколько иных. То есть задача была приближена к реальности: было необходимо добиться идентификации персон не только по изображениям, непосредственно входящим в базу данных, но и по *схожим с ними*, полученным с камеры и прошедшим предварительную обработку.

Первые результаты оказались несколько неожиданными (см. таблицу).

Примечательна *разница в качестве распознавания в первом и втором случаях*. Вероятно, именно лица в "обычных", наиболее часто встречающихся ракурсах, в предложенных условиях распознавались первыми двумя методами несколько хуже, чем лица в "существенно отклоняющихся" ракурсах. Для прочих методов ситуация обратная.

³ В первом каскаде элементарных интерполяторов МГУА обычно осуществляется выбор из n^2 элементарных классификаторов, где n — число входов. В нашем случае для повышения скорости работы проводился выбор из m^2 классификаторов, $m < n$, где m — число наиболее "перспективных" входов, отобранных в ходе корреляционного анализа.

Результаты верного распознавания персон различными методами, %

Метод				
Наибольшая близости	k -ближайших соседей	Линейные перцептроны	Двухслойные сети	МГУА
Изображения лиц в произвольных ракурсах				
77,5	72,5	67,5	—	60
Изображения лиц с принудительным варьированием ракурса				
83	76,6	55,3	57,8	55,3

Лидируют два простых метода — наибольшей близости и k -ближайших соседей. Из трех интерполяционных методов наихудшие результаты показал МГУА, видимо потому, что полиномы дали достаточно маловероятное приближение, которое в сочетании с завышенной степенью таких полиномов привело к возникновению множества очагов неверной идентификации. Нейросетевые методы дали более правдоподобное приближение, но, видимо, проявился хорошо известный эффект [9] скорее чрезмерного подстраивания под элементы обучающей выборки, чем выделения общей закономерности. Таким образом, для дальнейшей работы в качестве основных были выбраны оба упомянутых выше простых метода в сочетании с принципом голосования. Персона считается идентифицированной, если за нее подали голос оба метода. Особенно же интересен случай "полемики" методов, когда они рекомендуют разные персоны.

4. Арбитраж методов идентификации

При анализе экспериментальных данных по "полемике" было обнаружено, что помимо случаев, в которых оба простых метода ошибаются, есть ряд случаев, где хотя бы один из них дает верный ответ. Это позволило бы повысить точность предсказания, если бы удалось привлечь некоторый алгоритм-арбитр, который выберет верный ответ.

Достаточно сложной задачей оказался выбор арбитра. Было сделано предположение, что оказавшиеся более правдоподобными в сравнении с МГУА нейросетевые методы могут, если не идентифицировать одну персону среди множества, то, по меньшей мере, верно выбрать персону среди двух, предложенных методами наибольшей близости и k -ближайших соседей.

Были проведены дополнительные исследования, целью которых являлось определение наиболее перспективного метода-арбитра в случае "полемики" основных простых методов. Обнаружилось, что линейный перцептрон дает верный ответ в 50 % таких случаев, а двухслойная сеть — в 80 % случаев. Таким образом, предлагаемый в настоящей работе метод идентификации является двухэтапным. На первом этапе проводится идентификация двумя первич-

ными методами: 1) поиск ближайшей (по Махалобиссу) метрики в пространстве собственных лиц с рекомендацией соответствующей персоны; 2) поиск k -ближайших метрик в том же пространстве с рекомендацией персоны, которой соответствует наибольшее число этих метрик. Если рекомендации методов совпадают, то идентификация завершена. В противном случае начинается второй этап, на котором включается арбитр (либо линейные перцептроны, либо двухслойные сети), выбирающий персону среди двух рекомендованных. Выбирается персона, для которой арбитр дает большее значение.

Требование быстрой работы определило следующий алгоритм работы арбитра:

1. Если присутствуют обученные двухслойные нейронные сети для каждого из кандидатов, то выбирается персона, сеть которой дает лучший прогноз.

2. Если обученных сетей нет, то либо они оперативно обучаются (если ресурсы вычислительной системы позволяют выполнить обучение за доли секунды) и работают в соответствии с пунктом 1, либо задействуются линейные перцептроны (если обученные перцептроны отсутствуют, то они оперативно обучаются, это весьма быстрый процесс на современных ЭВМ).

5. Повышение скорости обучения нейронных сетей

В заключение рассмотрим подробно вопрос о максимально возможном повышении скорости обучения двухслойной нейронной сети. Методы, основанные на неполном переборе (например, случайного поиска) были отвергнуты по причине весьма значительных временных затрат на переобучение даже с применением параллельных вычислений. Метод обратного распространения ошибки (как и любой иной градиентный метод) подразумевает последовательную (даже в случае какого-либо распараллеливания счета) подстройку коэффициентов для каждого элемента обучающей выборки размера k . Поэтому время обучения включает существенный компонент с прямой пропорциональностью k . Очевидно, что данный факт приводит к существованию разумного ограничения k_{\max} на значение данной величины.

Предлагается включать в обучающую выборку лишь некоторые, наиболее характерные элементы. В нашем случае это можно сделать особенно просто, учитывая, что нейронная сеть является арбитром двух методов. В обучающую выборку двухслойной сети, соответствующей каждой персоне, должны входить: а) лица, соответствующие данной персоне; б) лица, соответствующие "конкурирующим" персонам, т. е. таким, с которыми у данной наиболее часто возникает "полемика" методов наибольшей близости и k -ближайших соседей; в) лица, соответствующие некоторому числу случайным образом выбранных персон, что должно несколько повысить потенциальную надежность арбитража.

6. Параллельный расчет на видеокартах

Для ускорения расчета было принято решение задействовать локальные вычислительные ресурсы многоядерных видеокарт. Этого должно было оказаться более чем достаточно для одновременного обучения двух и более нейронных сетей. Чтобы исключить привязку к конкретному производителю видеокарты, для распараллеливания был выбран интерфейс OpenCL.

Был выбран следующий принцип распараллеливания: каждая сеть отдавалась одному многопоточному мультипроцессору карты, каждый нейрон первого слоя сети обрабатывался отдельным потоком. Единственный нейрон второго, выходного слоя обрабатывался в каждом из этих потоков. Такая дублирующая схема оказалась наименее затратной, поскольку практически не требует координации потоков. При расчетах максимально возможно задействовалась внутренняя регистровая память потоков, там же, где это было невозможно, применялась разделяемая (*shared* — в терминологии производителя видеокарт, она же *local* — в терминологии OpenCL) память мультипроцессоров. При нехватке разделяемой памяти использовалась глобальная (преимущественно для передачи на видеокарту обучающей выборки).

7. Результаты параллельного счета

Проводилось сравнение результатов, полученных с применением видеокарты, с результатами, полученными на классическом процессоре. Прямое сравнение временных характеристик обычной однопроцессорной программы и OpenCL-программы дало результат, на первый взгляд производящий впечатление абсурдного: расчет на 64 потоковых процессорах nVidia GeForce GTX570 оказался в 217 раз быстрее, чем на одном ядре процессора Intel Core i5-2300 (2,8 ГГц). Однако, если учесть, что однопроцессорная программа не была оптимизирована (Debug-версия компилятора C++ среды Microsoft Visual Studio) и содержала значительное число операций индексного доступа к элементам массивов, от которых была свободна программа для видеокарты (за счет перехода от индекса массива к номеру потока), а также OpenCL-программа работала преимущественно с быстродействующей регистровой памятью, становится ясно, что вышеуказанный результат возможен.

Тем не менее, для контроля было принято решение о дополнительном сравнении с результатами, полученными той же OpenCL-программой, но запущенной уже не на видеокарте, а на вышеуказанном многоядерном центральном процессоре (ЦП) Intel с применением среды Intel OpenCL SDK 1.5. В данном случае расчет на видеокарте занял в 2,1 раза меньше времени, чем расчет на четырех ядрах ЦП,

т. е. примерно в $2,1 \times 4 = 8,4$ раза меньше, чем расчет на одном ядре ЦП. Данную величину (8,4) можно было бы обозначить как "физическое ускорение" P_{opt} , помимо которого нас интересует также "логическое ускорение" S_{opt} по отношению к одному ядру ЦП, определяемое масштабированием результата с учетом разницы тактовых частот. Тактовая частота f_{cpu} используемого ЦП (2,8 ГГц) в 2 раза выше тактовой частоты f_{gpu} потоковых (шейдерных) процессоров (1,4 ГГц) видеокарты. Следовательно, "логическое ускорение" составляет

$$S_{\text{opt}} = P_{\text{opt}} \frac{f_{\text{cpu}}}{f_{\text{gpu}}} = 8,4 \frac{2,8}{1,4} = 16,8.$$

Это значение все же приблизительно в 4 раза меньше ожидаемого (64, поскольку было задействовано именно такое число потоковых процессоров). Вероятно, такой дополнительный множитель объясняется меньшей эффективностью исполнения серии команд потоковыми процессорами, т. е. меньшим числом команд, исполняемым за такт, в сравнении с ЦП Intel, что связано, в частности, с применением векторных SSE-команд ЦП.

По той же формуле легко подсчитать "логическое ускорение" S_{nopt} по сравнению с расчетом неоптимизированной программой, работающей на одном ядре ЦП:

$$S_{\text{nopt}} = 217 \frac{2,8}{1,4} = 434.$$

Заключение

В данной работе предлагается новый метод идентификации лиц, включающий опрос методов наибольшей близости и k -ближайших соседей с последующим использованием нейронной сети (двуслойной или линейной однослойной) в качестве арбитра в случае "полемики" методов.

Показано, что применение арбитра позволяет повысить точность прогнозирования до 87–88 % при сохранении высокой скорости идентификации. Это достаточно хороший показатель, особенно с учетом того, что идентифицирующей системе предъявлялись изображения лиц не только из базы данных, но схожие с ними, в том числе повернутые и наклоненные.

При обучении двуслойных нейронных сетей применен перенос вычислений (с распараллеливанием) на видеокарту. На 64 потоковых процессорах nVidia GeForce GTX570 получено существенное логическое ускорение в 16,8 раз (физическое — в 8,4 раза) по сравнению с оптимизированным расчетом на одном ядре процессора Intel Core i5-2300 (2,8 ГГц) и в 434 раза (физическое — в 217 раз) по сравнению с неоптимизированным расчетом на том же процессоре.

Список литературы

1. Самаль Д. И. Алгоритмы идентификации человека по фотопортрету на основе геометрических преобразований: автореф. дис. ... канд. техн. наук. Минск, 2002. 20 с.
2. Дюк В., Самойленко А. Data mining: учебный курс. СПб.: Питер, 2001. 368 с.
3. Turk M., Pentland A. Eigenfaces for Recognition // Journal of Cognitive Neuroscience. 1991. Vol. 3, № 1. P. 71–86.
4. Гуляева Т. А., Попов А. А. Скрытые Марковские модели с одномерной топологией в задаче распознавания лиц // Сборник научных трудов НГТУ. 2006. № 1. С. 1–6.
5. Лопатина А. Д. Выделение области лица с помощью комбинации методов цветовой и яркостной сегментации // Вестник УГАТУ. Управление, ВТ и И. Т. 13, № 2 (35). С. 180–187.

6. Ahonen T., Hadid A., Pietikäinen M. Face recognition with local binary patterns // Proc. Eighth European Conf. Computer Vision. Springer, 2004. P. 469–481.
7. Viola P. A., Jones M. J. Rapid Object Detection using a Boosted Cascade of Simple Features // Computer Vision Pattern Recognition. 2001. N 1. P. 511–518.
8. Визильтер Ю. В., Желгов С. Ю., Князь В. А., Ходарев А. Н., Моржин А. В. Обработка и анализ цифровых изображений с примерами на LabVIEW и IMAQ Vision. М.: Книга по требованию. 2011. 464 с.
9. Рассел С., Норвиг П. Искусственный интеллект: современный подход. М.: Вильямс, 2007. 1408 с.

УДК 004.032.26

А. Ю. Дорогов¹, д-р техн. наук, проф.,
доц., e-mail: vaksa2006@yandex.ru,

В. С. Абатуров², инженер, аспирант,
e-mail: vasilianch@yandex.ru,

И. В. Раков², канд. техн. наук, начальник НИО,
e-mail: i_rakov@mail.ru,

¹ Санкт-Петербургский государственный
электротехнический университет "ЛЭТИ"

² ОАО "Интеллектуальные информационные
технологии", г. Санкт-Петербург

PMML-модели быстрых нейронных сетей

Рассмотрено использование PMML-стандарта для описания алгоритмов быстрых нейронных сетей и спектральных преобразований. Показано, что для представления быстрых алгоритмов может быть использована стандартная PMML-модель нейронной сети. Отмечаются недостатки данного решения. Представлено описание структуры и топологии быстрых алгоритмов на основе лингвистических моделей. Предложена новая PMML-модель, ориентированная на быстрые алгоритмы и устраняющая отмеченные недостатки. Приведены листинги PMML-моделей.

Ключевые слова: язык моделирования PMML, спектральные преобразования, быстрые алгоритмы, топологическая модель, структурная модель, быстрые нейронные сети

Введение

Стандарт PMML (Predictive Model Markup Language — PMML) предназначен для унификации представлений моделей алгоритмов и сопутствующих данных, относящихся к областям прогнозной аналитики и извлечению знаний [1]. Стандарт раз-

работан международной организацией Data Mining Group (DMG) на основе языка разметки XML.

Модель PMML не содержит алгоритма обработки данных, но полностью определяет его параметры, включая ссылку на используемый алгоритм, описание входных, выходных и целевых переменных, ссылки на вспомогательные преобразования для входных и выходных данных.

Глубокая унификация позволяет пользователям разрабатывать модели в приложении одного производителя и использовать в приложениях других производителей в целях визуализации, анализа, оценки и иного использования моделей. Ранее это было затруднено, но с PMML обмен моделями между совместимыми приложениями упростился. В настоящее время PMML стандартизует модели искусственных нейронных сетей, метода опорных векторов, ассоциативных правил, наивного байесовского классификатора, модели кластеризации, текстовые модели, деревья принятия решений, а также различные регрессионные модели. Перечень стандартизуемых моделей постоянно увеличивается. Стандарт поддерживает возможности расширения существующих моделей пользователем и не препятствует разработке собственных моделей для специфичных алгоритмов анализа и обработки данных. В данной статье рассматривается вариант построения PMML моделей для быстрых нейронных сетей и обобщенных быстрых спектральных преобразований.

Алгоритмы быстрых спектральных преобразований основаны на возможности факторизации матрицы преобразования в произведение слаботраченных матриц. Появление алгоритма быстрого преобразования Фурье (БПФ) стимулировало интерес и к другим видам спектральных преобразований. В задачах фильтрации, сжатия и выделения информативных признаков широкое применение нашли такие преобразования, как Адамара—Уолша, Хаара, Виленкина—Кристенсона, Хартли, Наклонное, Вейвлет и другие, также обладающие быстрыми алгоритмами. Несмотря на отличия по

видам функций оказалось, что большинство алгоритмов быстрых преобразований имеют подобную структуру и отличаются друг от друга не больше, чем значениями коэффициентов базовых операций. Осознание этого факта привело к идее построения обобщенных перестраиваемых спектральных преобразований, наделенных быстрым алгоритмом. Первые предложения по построению обобщенного ортогонального преобразования были высказаны Эндрюсом и Каспари [2] в 70-х годах прошлого века. А первые подходы к обучению подобных преобразований были развиты в работах А. И. Солодовникова и его научной группы [3]. В то время подобный класс преобразований называли приспособленными быстрыми преобразованиями. Быстрые нейронные сети сохраняют топологию быстрых преобразований, но имеют более широкие возможности по варьированию размерностями обрабатываемых данных. Если для спектральных преобразований размерности векторов входа и выхода совпадают, то для быстрых нейронных сетей размерности входных и выходных данных могут отличаться. Кроме того, нейронные сети содержат нелинейные функции активации и дополнительные входы смещений.

Представление спектрального преобразования PMML-моделью многослойной нейронной сети

Возможность перестройки значений весовых коэффициентов и многослойная структура алгоритма роднит быстрые перестраиваемые преобразования с многослойными нейронными сетями прямого распространения. Иногда используют термин — ортогональные нейронные сети. В рамках данной парадигмы быстрые линейные перестраиваемые преобразования являются частным случаем быстрых нейронных сетей [4] и отличаются от последних линейными функциями активации и нулевыми смещениями в нейронах.

Подобие с нейронными сетями позволяет использовать стандартные конструкции PMML для представления моделей быстрых преобразований. На рис. 1 показано нейросетевое представление БПФ размерности 8 в топологии Кули—Тьюки "с прореживанием по частоте". Граф состоит из трех слоев нейронов (первый слой — входные контакты). Каждая вершина графа (кроме первого слоя) представляет собой нейрон с двумя входами. Все нейроны имеют линейные функции активации и нулевое смещение. Входным дугам соответствуют значения весов базовых операций, в общем случае это комплексные числа.

Графу можно сопоставить произведение слабозаполненных матриц, как показано на рис. 2. Элементы матриц, помеченные точками, отличны от нуля, все остальные значения элементов матриц — нулевые. Число ненулевых значений определяет число операций умножения при выполнении быстрого алгоритма, в данном случае полное число

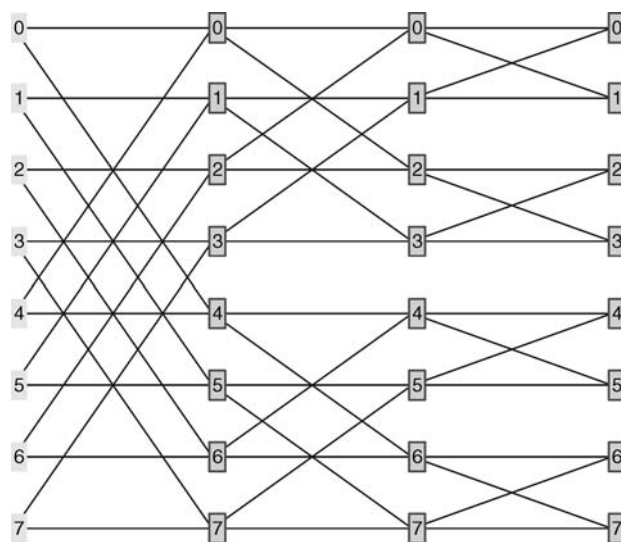


Рис. 1. Граф алгоритма быстрого преобразования Фурье в топологии Кули—Тьюки "с прореживанием по частоте"

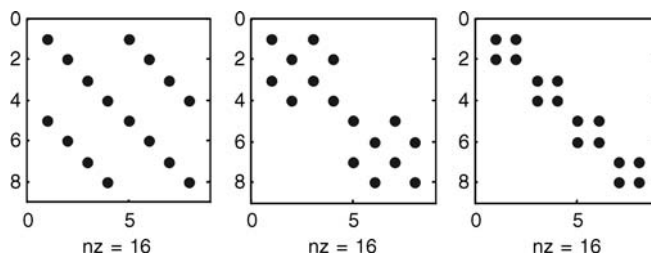


Рис. 2. Факторизованное представление алгоритма БПФ; nz — число ненулевых элементов матрицы

ненулевых элементов равно 48, в то время как для прямого выполнения дискретного преобразования Фурье размерности 8 это значение равно 64. Выигрыш по быстродействию быстро растет с увеличением размерности алгоритма.

Листинг 1 представляет фрагмент PMML модели нейронной сети [1] для данного преобразования. Фрагмент содержит заголовок модели и для экономии места включает описание только трех нейронов первого слоя. В листинге подробно не раскрыты также поля DataDictionary и MiningSchema, поскольку они специфичны для каждого конкретного использования модели.

Листинг 1. Фрагмент PMML модели для БПФ в стандартном представлении

```
<?xml version="1.0" encoding="utf-8"?>
<PMML xmlns="http://www.dmg.org/PMML-4_0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="4.0">
  <Header copyright="NIO-012"/>
  <MiningBuildTask>
    <Extension buildTask="NeuralNetwork Creator"/>
  </MiningBuildTask>
  <DataDictionary/>
```

```

<NeuralNetwork activatioFunction="" function-
Name="Digital Processing" modelName="Orthogonal
Fast Transformation Fourier" numberOfInputs="8"
numberOfLayers="3">
  <MiningSchema/>
  <NeuralInputs>
    <NeuralInput id="0"/>
    <NeuralInput id="1"/>
    <NeuralInput id="2"/>
    <NeuralInput id="3"/>
    <NeuralInput id="4"/>
    <NeuralInput id="5"/>
    <NeuralInput id="6"/>
    <NeuralInput id="7"/>
  </NeuralInputs>
  <NeuralLayer numberOfNeurons="8">
    <Neuron id="10">
      <Con from="0" weight="0.7071"/>
      <Con from="1" weight="0"/>
      <Con from="2" weight="0"/>
      <Con from="3" weight="0"/>
      <Con from="4" weight="0.7071"/>
      <Con from="5" weight="0"/>
      <Con from="6" weight="0"/>
      <Con from="7" weight="0"/>
    </Neuron>
    <Neuron id="11">
      <Con from="0" weight="0"/>
      <Con from="1" weight="0.7071"/>
      <Con from="2" weight="0"/>
      <Con from="3" weight="0"/>
      <Con from="4" weight="0"/>
      <Con from="5" weight="0.7071"/>
      <Con from="6" weight="0"/>
      <Con from="7" weight="0"/>
    </Neuron>
    <Neuron id="12">
      <Con from="0" weight="0"/>
      <Con from="1" weight="0"/>
      <Con from="2" weight="0.7071"/>
      <Con from="3" weight="0"/>
      <Con from="4" weight="0"/>
      <Con from="5" weight="0"/>
      <Con from="6" weight="0.7071"/>
      <Con from="7" weight="0"/>
    </Neuron>
  </NeuralLayer>
  <NeuralOutputs numberOfOutputs="8">
    <NeuralOutput outputNeuron="0"/>
    <NeuralOutput outputNeuron="1"/>
    <NeuralOutput outputNeuron="2"/>
    <NeuralOutput outputNeuron="3"/>
    <NeuralOutput outputNeuron="4"/>
    <NeuralOutput outputNeuron="5"/>
    <NeuralOutput outputNeuron="6"/>
    <NeuralOutput outputNeuron="7"/>
  </NeuralOutputs>
</NeuralNetwork>
</PMML>

```

Недостатком данного представления является избыточность описания: каждый нейрон в алгоритме БПФ (степени двойки) имеет только два входа, но в соответствии со стандартом необходимо при описании указывать связи, идущие от всех нейронов предшествующего слоя, показывая их веса как нулевые. Избыточность описания быстро растет с увеличением размерности преобразования. Вторым недостатком связан с отсутствием в стандартной форме информационных полей для представления топологии и структуры быстрого алгоритма. Для режима использования модели это не имеет принципиального значения, однако для создания и настройки модели, особенно в классе обобщенных преобразований, эта информация принципиально необходима.

Топологическая модель быстрого алгоритма

Обычно под топологией быстрого алгоритма понимают граф его реализации. Топологическая модель быстрого преобразования является производной от структурной модели, поэтому далее они будут рассмотрены вместе. Обе модели используют понятие базовой операцией, которое возникло в самом начале разработок быстрых алгоритмов. Для БПФ алгоритмов с размерностью, равной степени двойки, базовая операция представляет собой матрицу размерности 2×2 . Исторически базовую операцию данного типа называют "бабочкой". На рис. 3 показан граф быстрого преобразования с выделением базовых операций. Граф описывает тот же алгоритм, что и граф, показанный на рис. 1.

Граф топологической реализации имеет регулярную структуру и для его компактного описания можно воспользоваться лингвистическими конструкциями. Обозначим через u_m переменную, определяющую локальный номер контакта базовой операции в слое m . В теории быстрых алгоритмов

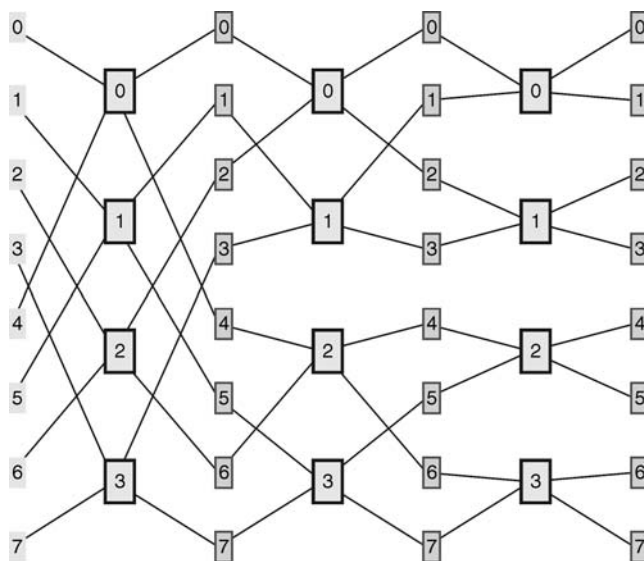


Рис. 3. Граф БПФ с выделением базовых операций

нумерация контактов начинается с нулевого индекса, поэтому переменная u_m для базовой операции "бабочка" будет принимать значения $\{0, 1\}$. Аналогично введем переменную v_m , определяющую локальный номер выходного контакта базовой операции в слое m . Тогда топологию алгоритма БПФ с прореживанием по частоте можно описать регулярной схемой [5]:

$$U^m = \langle v_0 v_1 \dots v_{m-1} u_m u_{m+1} \dots u_n - 2 u_n - 1 \rangle;$$

$$V^m = \langle v_0 v_1 \dots v_{m-1} v_{m+1} \dots u_n - 2 u_n - 1 \rangle,$$

где n — число слоев быстрого алгоритма. Контакты имеют локальную нумерацию, определенную разрядной переменной в пределах базовой операции и глобальную нумерацию в пределах слоя. Схема описывает правило порождения слов, состоящих из букв с родовыми именами u и v для слоя m . Совокупность слов, полученная из схемы для $m = 0, 1, \dots, n - 1$, образует пару лингвистических предложений, отвечающих системному инварианту быстрых алгоритмов [6]. Например, для $N = 8$, где граф алгоритма имеет три слоя, получим два лингвистических предложения, состоящих из трех слов каждое:

$$\begin{aligned} U &= [\langle u_0 u_1 u_2 \rangle \langle v_0 u_1 u_2 \rangle \langle v_0 v_1 u_2 \rangle]; \\ V &= [\langle v_0 u_1 u_2 \rangle \langle v_0 v_1 u_2 \rangle \langle v_0 v_1 v_2 \rangle]. \end{aligned} \quad (1)$$

Любая перестановка букв в пределах слова определяет новый вариант топологии. Слово предложения интерпретируется как поразрядное представление номера контакта в двоичной системе счисления в пределах слоя, например,

$$\langle u_0 u_1 u_2 \rangle = u_0 2^2 + u_1 2^1 + u_2 2^0.$$

Учитывая, что разряды u_m и v_m определяют номера входных и выходных контактов базовых операций в слое m , из формулы (1) получим, что номера вершин по слоям будут иметь разрядные представления:

$$i^0 = \langle u_1 u_2 \rangle, i^1 = \langle v_0 u_2 \rangle, i^2 = \langle v_0 v_1 \rangle.$$

Для построения графа топологической реализации алгоритма БПФ размерности $N = 8$ воспользуемся поразрядным представлением номеров координат векторов на входе и выходе слоя (см. табл. 1). Для входного вектора слоя используется обозначение x^m , для выходного — y^m . Каждый столбец таблицы определяет номер вершины топологического графа (см. рис. 1). При построении графа вершины смежных слоев объединяются между собой дугами, если в смежных столбцах таблицы одноименные разрядные переменные имеют совпадающие значения. Нетрудно проверить, что граф БПФ "с прореживанием по частоте", показанный на рис. 1, полностью отвечает предложенной схеме построения.

Эта же таблица может быть использована для построения графа с выделением базовых операций (см. рис. 3). К базовой операции слоя относятся

Двоичная нумерация контактных полей

Глобальный номер контакта	x^0	$x^1 = y^0$	$x^2 = y^1$	y^2
	$\langle u_0 u_1 u_2 \rangle$	$\langle v_0 u_1 u_2 \rangle$	$\langle v_0 v_1 u_2 \rangle$	$\langle v_0 v_1 v_2 \rangle$
0	$\langle 0 0 0 \rangle$	$\langle 0 0 0 \rangle$	$\langle 0 0 0 \rangle$	$\langle 0 0 0 \rangle$
1	$\langle 0 0 1 \rangle$	$\langle 0 0 1 \rangle$	$\langle 0 0 1 \rangle$	$\langle 0 0 1 \rangle$
2	$\langle 0 1 0 \rangle$	$\langle 0 1 0 \rangle$	$\langle 0 1 0 \rangle$	$\langle 0 1 0 \rangle$
3	$\langle 0 1 1 \rangle$	$\langle 0 1 1 \rangle$	$\langle 0 1 1 \rangle$	$\langle 0 1 1 \rangle$
4	$\langle 1 0 0 \rangle$	$\langle 1 0 0 \rangle$	$\langle 1 0 0 \rangle$	$\langle 1 0 0 \rangle$
5	$\langle 1 0 1 \rangle$	$\langle 1 0 1 \rangle$	$\langle 1 0 1 \rangle$	$\langle 1 0 1 \rangle$
6	$\langle 1 1 0 \rangle$	$\langle 1 1 0 \rangle$	$\langle 1 1 0 \rangle$	$\langle 1 1 0 \rangle$
7	$\langle 1 1 1 \rangle$	$\langle 1 1 1 \rangle$	$\langle 1 1 1 \rangle$	$\langle 1 1 1 \rangle$

контакты, которые в смежных столбцах таблицы имеют два одноименных разряда с попарно совпадающими значениями. Например, для слоя 0 базовая операция $i^0 = \langle u_1 u_2 \rangle = \langle 00 \rangle$ имеет контакты по входу и выходу с глобальными номерами 0 и 4. Для слоя 1 базовая операция $i^1 = \langle v_0 u_2 \rangle = \langle 01 \rangle$ имеет контакты по входу и выходу с глобальными номерами 1 и 3. Топологический граф показывает, как располагаются базовые операции в контактном поле слоев.

Структурная модель

Структурная модель быстрого алгоритма представляет собой граф, вершинами которого являются базовые операции, а дуги устанавливают операторные связи между ними. На уровне структурной модели абстрагируются от координатной привязки базовых операций и значений их коэффициентов. Покажем построение графа структурной модели на примере БПФ преобразования для $N = 8$. В аналитической форме структурная модель представляется лингвистическим предложением

$$[\langle u_1 u_2 \rangle \langle v_0 u_2 \rangle \langle v_0 v_1 \rangle],$$

состоящим из трех слов, составленных из букв с двумя родовыми именами u, v . Предложение дополняется таблицей значений для оснований разрядных переменных. Каждое слово предложения интерпретируется как номер базовой операции. Для построения структурной модели воспользуемся поразрядным представлением номеров базовых операций, представленным в табл. 2.

Таблица 2

Поразрядная нумерация базовых операций

Номер вершины	Слой 0	Слой 1	Слой 2
	$\langle u_1 u_2 \rangle$	$\langle v_0 u_2 \rangle$	$\langle v_0 v_1 \rangle$
0	$\langle 0 0 \rangle$	$\langle 0 0 \rangle$	$\langle 0 0 \rangle$
1	$\langle 0 1 \rangle$	$\langle 0 1 \rangle$	$\langle 0 1 \rangle$
2	$\langle 1 0 \rangle$	$\langle 1 0 \rangle$	$\langle 1 0 \rangle$
3	$\langle 1 1 \rangle$	$\langle 1 1 \rangle$	$\langle 1 1 \rangle$

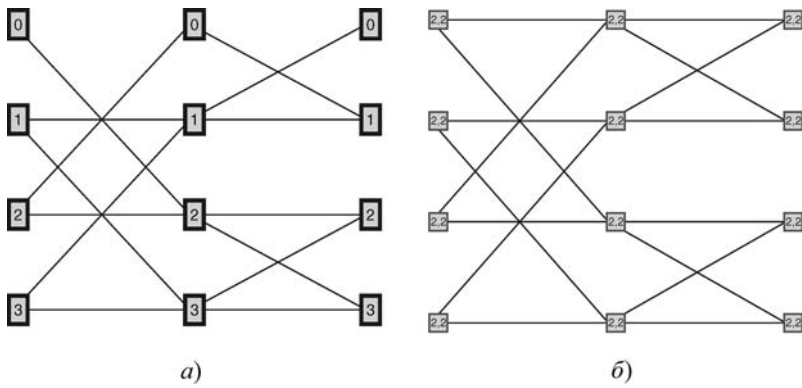


Рис. 4. Структурная модель алгоритма БПФ:
a — нумерация вершин; *б* — веса вершин

Объединяя между собой вершины смежных слоев, для которых одноименные разрядные переменные в таблице имеют совпадающие значения, получим граф, показанный на рис. 4, *a*. Вес вершин представлен парой чисел, определяющих размерность базовой операции по входу и выходу. На приведенном графе рис. 4, *б* веса записаны в поле вершины.

Базовые операции смежных слоев связаны между собой только по одной из координат, поэтому операторный ранг (r_{ij}) всех связей равен единице.

Для спектральных преобразований основания разрядных переменных u_m и v_m совпадают. Для быстрых нейронных сетей эти значения могут быть различны, в этом случае базовая операция (точнее — карта синапсов) представляет собой прямоугольную матрицу. Пример топологического графа быстрой нейронной сети для оснований, заданных соответ-

ствиями $\begin{pmatrix} u_0 & u_1 & u_2 \\ 2 & 3 & 2 \end{pmatrix}, \begin{pmatrix} v_0 & v_1 & v_2 \\ 2 & 2 & 2 \end{pmatrix}$, показан на рис. 5.

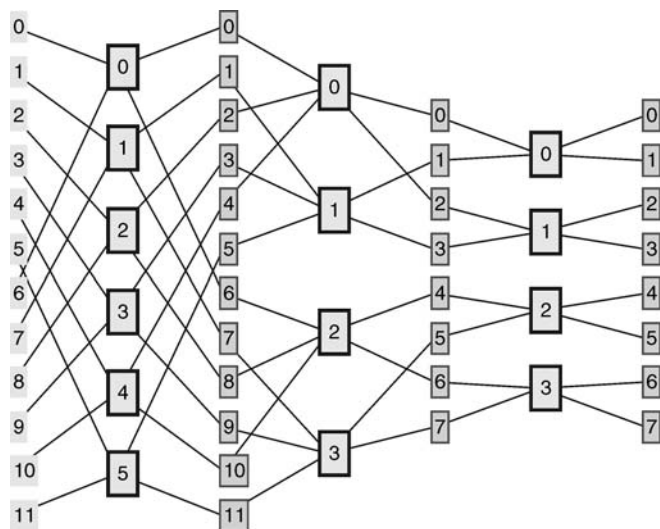


Рис. 5. Граф быстрой нейронной сети в топологии Кули—Тьюки "с прореживанием по частоте" и с выделением нейронных ядер

PMML-модель быстрого алгоритма

Представленная ниже PMML модель (см. листинг 2) учитывает особенности быстрых алгоритмов. Для представления топологии и структуры алгоритма в модель введен дополнительный раздел ProductionScheme (порождающая схема), который содержит элементы Radixes (основания разрядных переменных), LayerStructures (описание структурной модели) и LayerTopologies (описание топологической модели). Каждый слой алгоритма описывается лингвистическим словом, содержащим переменные с родовыми именами u и v . Для описания структуры слоя используется элемент LayerStructureScheme, а для описания топологии — элементы LayerTopologicalScheme, содержащие вложенные описания для входных и выходных полей базовых операций: ReceptorField и AxonField.

Данный раздел позволяет однозначно восстановить топологию и структуру быстрого алгоритма. Числовые параметры преобразования описаны на уровне базовых операций, для представления которых используется элемент Kernel (нейронное ядро). В пределах ядра связи между аксонами и рецепторами описаны, как и в классической модели нейронной сети, с помощью элементов Con.

Листинг 2. Фрагмент PMML модели алгоритма БПФ в ядерном представлении

```
<?xml version="1.0" encoding="utf-8"?>
<PMML xmlns="http://www.dmg.org/PMML-4_0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="4.0">
  <Header copyright="NIO-012"/>
  <MiningBuildTask>
    <Extension buildTask="NeuralNetwork Creator"/>
  </MiningBuildTask>
  <DataDictionary/>
  <FastNeuralNetwork activationFunction="" functionName="Digital Processing" modelName="Orthogonal Fast Transformation Fourier" numberOfInputs="8" numberOfLayers="3">
    <MiningSchema/>
    <ProductionScheme>
      <Radixes>
        <inputRadixes type="INT-ARRAY">2 2 2
</inputRadixes>
        <outputRadixes type="INT-ARRAY">2 2 2
</outputRadixes>
      </Radixes>
      <LayerStructures>
        <LayerStructureScheme id="0" type="STRING-ARRAY">u0 u1 </LayerStructureScheme>
        <LayerStructureScheme id="1" type="STRING-ARRAY">u0 v1 </LayerStructureScheme>
      </LayerStructures>
    </ProductionScheme>
  </FastNeuralNetwork>
</PMML>
```

```

    <LayerStructureScheme id="2" type=
"STRING-ARRAY">v0 v1 </LayerStructureScheme>
  </LayerStructures>
  <LayerTopologies>
    <LayerTopologicalScheme id="0">
      <ReceptorFieldType="STRING-
ARRAY">u0 u1 u2</ReceptorField>
      <AxonField type="STRING-
ARRAY">u0 u1 v2</AxonField>
    </LayerTopologicalScheme>
    <LayerTopologicalScheme id="1">
      <ReceptorField type="STRING-
ARRAY">u0 u1 v2</ReceptorField>
      <AxonField type="STRING-
ARRAY">u0 v1 v2</AxonField>
    </LayerTopologicalScheme>
    <LayerTopologicalScheme id="2">
      <ReceptorField type="STRING-
ARRAY">u0 v1 v2</ReceptorField>
      <AxonField type="STRING-
ARRAY">v0 v1 v2</AxonField>
    </LayerTopologicalScheme>
  </LayerTopologies>
</ProductionScheme>
<NeuralInputs>
  <NeuralInput id="0"/>
  <NeuralInput id="1"/>
  <NeuralInput id="2"/>
  <NeuralInput id="3"/>
  <NeuralInput id="4"/>
  <NeuralInput id="5"/>
  <NeuralInput id="6"/>
  <NeuralInput id="7"/>
</NeuralInputs>
<FastNeuralLayer id="0" numberOfKernels="4">
<Kernel id="0" numberOfNeurons="2">
<Neuron id="0">
  <Con from="0" weight="0.70711"/>
  <Con from="1" weight="0.70711"/>
</Neuron>
<Neuron id="1">
  <Con from="0" weight="0.70711"/>
  <Con from="1" weight="-0.70711-
8.6596e-017i"/>
</Neuron>
</Kernel>
<Kernel id="1" numberOfNeurons="2">
  <Neuron id="0">
    <Con from="0" weight="0.70711"/>
    <Con from="1" weight="0.70711"/>
  </Neuron>
  <Neuron id="1">
    <Con from="0" weight="0.70711"/>
    <Con from="1" weight="-0.70711 -
8.6596e-017i"/>
  </Neuron>
</Kernel>
..

```

```

..
</FastNeuralLayer>
  <NeuralOutputs numberOfOutputs="8">
    <NeuralOutput outputNeuron="0"/>
    <NeuralOutput outputNeuron="1"/>
    <NeuralOutput outputNeuron="2"/>
    <NeuralOutput outputNeuron="3"/>
    <NeuralOutput outputNeuron="4"/>
    <NeuralOutput outputNeuron="5"/>
    <NeuralOutput outputNeuron="6"/>
    <NeuralOutput outputNeuron="7"/>
  </NeuralOutputs>
</FastNeuralNetwork>
</PMML>

```

Модель соответствует алгоритму БПФ "с прореживанием по частоте". Для экономии места представлено описание только двух базовых операций первого слоя быстрого преобразования. Для быстрых нейронных сетей элементы Kernel будут содержать также элемент bias (смещения) и атрибут activation-Function (тип функции активации).

Заключение

Для представления быстрых алгоритмов может быть использована стандартная PMML модель нейронной сети. Однако такое решение приводит к неоправданной избыточности описания и не позволяет использовать модель на этапе инициализации и обучения. Специфика алгоритмов быстрых алгоритмов требует разработки специальных PMML моделей, включающих описание структуры и топологии алгоритма. Предложенная модель устраняет избыточность и содержит дополнительные элементы, позволяющие однозначно восстановить структуру и топологию быстрого алгоритма. Модель покрывает класс быстрых нейронных сетей [7], для которого быстрые спектральные преобразования являются частным случаем.

Список литературы

1. **The Data Mining Group (DMG)**. URL: <http://www.dmg.org/>
2. **Andrews H. C., Caspari K. L.** A General Techniques for Spectral Analysis // IEEE. Tr. Computer. 1970. Vol. C-19. Jan, N 1. P. 16–25.
3. **Солодовников А. И., Спиваковский А. М.** Основы теории и методы спектральной обработки информации. Л.: Изд. ЛГУ, 1986. 272 с.
4. **Дорогов А. Ю.** Быстрые нейронные сети: Проектирование, настройка, приложения // Лекции по нейроинформатике. Ч. 1. Тр. школы-семинара "Современные проблемы нейроинформатики", науч.-техн. конф. "Нейроинформатика-2004" 28–30 января 2004 г. Москва. Изд. М.: МИФИ. 2004. С. 69–135.
5. **Дорогов А. Ю.** Порождающие грамматики быстрых нейронных сетей // Нейрокомпьютеры: разработка и применение. 2002. № 9–10. С. 10–25.
6. **Дорогов А. Ю.** Системные инварианты быстрых перестраиваемых преобразований // Нейрокомпьютеры: разработка и применение. 2006. № 6. С. 43–50.
7. **Дорогов А. Ю.** Реализация спектральных преобразований в классе быстрых нейронных сетей // Программирование. 2003. № 4. С. 13–26.

CONTENTS

Stennikov V. A., Barakhtenko E. A., Sokolov D. V. <i>Ontologies Application for Implementation of the Software for Solving Optimal Design Problems of Heat Supply Systems</i>	2
--	----------

The complex method for the solving of optimal design problems of heat supply systems is suggested. The method is based on principles of decomposition and further coordination of the calculation results during integrated computing process. Applied earlier approaches for software development didn't provide the implementation of the complex method in the form of the integrated software. To solve the problem the authors suggest a methodological approach to create of the software used new technologies. The approach is based on application of model-driven development where the ontological system is used for model formalization. The developed approach has been used for implementation of the SOSNA-M software for the solution of optimal design problems of heat supply systems.

Keywords: ontology, ontological system, software, heat supply system, optimal design, script, metaprogramming

Cheremisina L. D., Kirienko N. A. <i>Low Power Driven Synthesis of Multi-Level Logical Circuits</i>	8
--	----------

An approach to synthesis of multi-level logical gate circuits that could serve as good base for logical design on the base of CMOS library elements is proposed and investigated. The method is targeted minimization of integrated microcircuit area implemented on CMOC VLSI chip and average value of dissipated power.

Keywords: logic design automation, custom CMOS VLSI, low-power synthesis

Mammadov R. G., Vladimirovsky E. I., Mustafayeva S. R. <i>Anomalous Diffusion and its Influence on Chaotic Systems</i>	15
---	-----------

In article the new structures of interaction systems of types Lorenz — Chen — anomalous diffusion and Rabinovich-Fabrikant — anomalous diffusion are submitted. The describing features of structures — recurrence diagrams are received.

Keywords: sequence, Lorenz—Chen system, Rabinovich—Fabrikant system, anomalous diffusion, nonlinear recurrence analysis

Chetyrbotsky A. N. <i>The Application of Direct Method for Solving Systems of Linear Equations for the Numerical Simulation of Spherical Couette Flow of an Incompressible Fluid in MATLAB</i>	20
---	-----------

Questions of algorithms velocity components of the spherical Couette flow, which corresponds to the model representation of diverse real-world processes of convective transport. An algorithm for calculation and evaluation of the effectiveness of its implementation in the environment MATLAB. Based on a series of numerical experiments shows performance of the proposed method of numerical simulation of spherical Couette flow of Newtonian viscous incompressible fluid.

Keywords: viscous liquid spherical Couette flow, methods of solving systems of linear equations, sparse matrix of large dimension, tools, graphical presentation of the results of computational experiments.

Elesina S. I., Loginov A. A., Nikiforov M. B. <i>Methods of Assessing the Quality of Image Matching Criterion Function</i>	24
---	-----------

This paper describes the method of assessing the quality of the criterion function obtained when combining images of correlation-extremal navigation system. We consider the correlation and difference criterial function. Formed set of quality indicators of the criterion function. A ranking of quality kriteriynoy features prioritized and implemented a multi-criteria selection of the appropriate functions.

Keywords: the criterion function, global extremum, multiobjective optimization, the objective function, the correlation-extremal navigation systems, indicators of quality

Rechistov G. S. <i>Full Platform Simulation of Supercomputer Systems Aimed at Performance Prediction of Scientific Applications</i>	29
--	-----------

Extensions for two existing analytical models of performance of multicore computers into area of multi-host computing clusters are proposed. For both extended models full platform simulation is used to obtain input data in order to predict speed and behavior of scientific applications on future hardware systems.

Keywords: simulation, supercomputers, Simics, CPI analysis, bottleneck analysis

Saak A. E. <i>Polynomial Algorithms for Hyperbolic-Type Task Queues Scheduling</i>	33
---	-----------

A hyperbolic-type task queue waiting for service in a Grid system or multiprocessor computer system is considered. Level polynomial algorithms for hyperbolic-type quadratic tasks assigning are proposed and considered. The algorithms can be used by a scheduler of MCS or Grid technology center.

Keywords: Grid system, multiprocessor computer system, scheduling, hyperbolic-type quadratic task queue, level polynomial algorithm

Bobkov V. A., Ronshin Yu. I., Mashentsev V. Yu., Kudrashov A. P. <i>Autonomous Underwater Vehicle Navigation from the Video Stream</i>	36
---	-----------

The method of determining the trajectory of the underwater vehicle on video stream from two onboard cameras is represented. The method is based on feature matching and calculation of transformation between coordinate systems on different times. A calculation of transformation is accomplished by means of optimization from matching of two 3D point clouds.

Keywords: underwater vehicle, navigation, SLAM, 3D point cloud, feature matching, filtering, calibration, CUDA

Bimakov V. A., Kasimov T. R. *Using of Voxel Computers for High-Speed Transformation of 2D and 3D High-Resolution Images* 42

The principles of a high-speed geometric transformation of 2D and 3D high resolution images by means of voxel computers (VC) are being described. VC — digital devices, based on the deep paralleling of computational procedures most used in the processing of scenes and images. The processing method realized in VC, allows to offer the new high-speed algorithms, running time of which does not depend on geometric form processed object.

Keywords: high-speed processing of scenes and images, geometrical models.

Barskiy A. B., Nguen Van Loy. *Evaluation of the Average Inquiry Execution Time for a Multi-Server Database with Circulating Segments* 48

With the help of a mathematical model for fixed values of the parameters is the mean time a query on a multi-server database with circulating segments on the intensity of the flow. Demonstrates the feasibility of service flows is much greater intensity.

Keywords: a multi-server database, the circulation of the segments, the intensity of the flow of requests, the intensity of the flow of the service, the average query execution time.

Vlasov S. V. *Entity-Oriented Approach to the Development of Information Storage (in Gaz Transportation Problems)* 51

The paper studies the classic approach to DB structure building, based on data normalization, and entity-oriented approach, based on meta-information description of data domain objects.

The article shows advantages and disadvantages of classic method, the basic premises of transition from the classic DB structure to the structure based on data meta-information, and conditions of use for each method.

Keywords: relational database, DB structure, entity, normalization, version control

Artemenkov S. L. *Recreation of the Mirakyan's "Perception" Model for "Idea's and Form's Code Creation" of Objects* 56

The recreation of a general construction of the Mirakyan's "perceptron" model is presented. Model is based on implementation of transcendental psychology principles and performs code creation of "idea (name) and form" of objects. Some important characteristics and functional features of the model are described.

Keywords: Mirakyan, model, perceptron, form, name, coding, perception, transcendental psychology

Pekunov V. V. *The Face Identification Using the Voting Method with a Neural Network Arbitration. The Parallel Solution on the Multicore Videocards* 61

A different approaches to the identification using principal components analysis are discussed. The new identification method is proposed. This method involves using of the nearest closure method and of the k nearest neighbours method with a sequential neural network arbitration. The approach to the neural network calculations parallelizing using an OpenCL technology on the videocards is developed. This approach concerns a back-propagation calculations. An enhancement of the face identification quality is shown.

Keywords: face identification, principal components analysis, neural network arbitration, voting method, parallel OpenCL solution on the videocard

Dorogov A. Yu., Abaturov V. S., Rakov I. V. *PMML Models of Fast Neural Networks* 65

In the article the application of PMML standard for description of algorithms for fast neural networks and spectral transformations is investigated. It is shown that standard neural networks PMML model may be used for representation of the fast algorithms. It is marked disadvantages of the decision. It is represented description of structure and topology of fast algorithms on base of linguistic models. It is suggested new PMML model, which oriented to fast algorithms without marked disadvantages. Examples of PMML models are shown.

Keywords: model language PMML, spectral transformations, fast algorithms, topological model, structure model, fast neural networks

Адрес редакции:

107076, Москва, Стромьинский пер., 4

Телефон редакции журнала (499) 269-5510

E-mail: it@novtex.ru

Дизайнер *Т.Н. Погорелова*. Технический редактор *Е.В. Конова*.

Корректор *Е.В. Комиссарова*.

Сдано в набор 27.12.2012. Подписано в печать 19.02.2013. Формат 60×88 1/8. Бумага офсетная.

Усл. печ. л. 8,86. Заказ ИТ313. Цена договорная.

Журнал зарегистрирован в Министерстве Российской Федерации по делам печати, телерадиовещания и средств массовых коммуникаций.

Свидетельство о регистрации ПИ № 77-15565 от 02 июня 2003 г.

Оригинал-макет ООО "Авансес солюшнз". Отпечатано в ООО "Авансес солюшнз".

105120, г. Москва, ул. Нижняя Сыромятническая, д. 5/7, стр. 2, офис 2.
