СЕКЦИЯ 1

ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ

Среда, 25 апреля 2012 г., читальный зал преподавателей (ауд.229), гл. корпус МГТУ им.Н.Э.Баумана.
Начало в 10.00.

Председатель: профессор, д.т.н. Норенков И.П.
Руководитель экспертной комиссии: к.т.н., доцент Федорук В.Г.
Ученый секретарь: к.т.н., доцент Власов А.И.

Экспертная комиссия:
БОЖКО А.Н., ВОЛОСАТОВА Т. М., ЖУК Д. М., МАНИЧЕВ В. Б., МАРТЫНЮК В. А., ТРУДОНОШИН В. А., ГРОШЕВ С. В., КНЯЗЕВА С., РОДИОНОВ С. В., ФЕДОРУК Е.

В рамках работы секции будет осуществлен первый этап конкурсного отбора проектов для участия в конкурсе инновационных проектов МГТУ им.Н.Э.Баумана. Критериями конкурсного отбора участников молодежного научно-инновационного конкурса являются:

1. Уровень инновационности идеи (предложения, метода, способа …).

1.1. Идея должна быть новой, впервые сформулированной именно самим номинантом. В этом смысле все номинанты равны.

1.2. Уровень наукоемкости тем выше, чем более:
- основательны научные исследования, в результате которых она появилась;
- основательны дальнейшие научные исследования, необходимые для ее реализации.

1.3. Техническая значимость тем выше, чем большее влияние ее реализация окажет на уровень техники. «Пионерные» идеи (изобретения) – пенициллин, лазер, синтез алмазов - открывают новые отрасли науки и техники. Идеальная по инновационности идея неожиданна для рынка. Поэтому она им не может быть сейчас востребована, она сама формирует новую потребность и нишу рынка.

Высокий технический уровень имеют решения, например, многоотраслевого использования.

Оригинальные технические решения дают новые принципы решения известной задачи.

Есть решения, позволяющие решить проблему еще одним, дополнительно к известным, способом. Но и в этом случае, если уже известно 10 способов решения задачи, то новый способ может и не давать существенных преимуществ в решении задачи или давать их только в очень ограниченном по масштабам применения числе случаев.

1.4. Масштабность использования предложения тоже может сильно различаться – от решения локальной задачи одного местного потребителя до …

1.5. Срок превращения идеи в конечный продукт с выходом его на рынок: новизна, рискованность идеи, объем необходимых научных исследований не позволяют уложиться в 2-3 года, но и не требуют 10-15 лет

1.6. Идея тем актуальней, чем меньше вероятность того, что за 5-7 лет в результате научных исследований появятся и «раскрутятся» до продукта другие более эффективные пути решения задачи.

Победитель сам организует работу по привлечению необходимого финансирования. Инновационная программа МГТУ им.Н.Э.Баумана оказывает ему поддержку на начальном этапе работы над его идеей.

ПРОБЛЕМЫ МОДЕЛИРОВАНИЯ КИНЕМАТИКИ И ДИНАМИКИ МЕХАНИЗМОВ БОЛЬШОЙ СЛОЖНОСТИ

Ивченков А. О.
Научный руководитель: к.т.н., доцент, Мартынюк В. А.
МГТУ им. Н.Э.Баумана, кафедра РК6, Москва, Россия

PROBLEMS OF MODELING KINEMATICS AND DYNAMICS OF MECHANISMS OF GREATCHALLENGES

Ivchenkov A. O.
Supervisor: Ph.Dr., Associate Prof., Martynyuk V. A.
MSTU, Moscow, Russia

Аннотация
В статье рассматриваются этапы, результаты и проблемы моделирования кинематики и динамики механизмов большой сложности в двух системах проектирования: Catia и NX UniGraphics. Подробно исследовано приложение DMU Kinematics тяжелой графической системы CATIA. Рассмотрены особенности сборки. На основе проведенного моделирования сделаны выводы об эксплуатационных условиях. В заключении даны рекомендации для будущих анализов сложных кинематических схем.

Die Inhaltsangabe
The article discusses the steps, results and problems of modeling the kinematics anddynamics of mechanisms of great complexity in the design of two systems: Catia and NX Unigraphics. Studied in detail the application DMU Kinematics of heavy graphics CATIA. The features of the assembly. Based on the modeling conclusions about the operational conditions. Finally, recommendations for future analyzes of complex kinematic patterns.

Введение

В настоящее время при проектировании механизмов большой сложности чаще всего используются одна из двух «тяжелых» САПР - Catia или NX Unigraphics. При этом, в большинстве случаев необходимо смоделировать кинематику того или иного механизма. Насколько бы хорошо не были разработаны эти системы своими производителями, но на разных этапах моделирования проектировщик сталкивается с рядом проблем, возникающим по тем или иным причинам.

В докладе представлена трехмерная визуализация триподных многосекционных механизмов и рассмотрены проблемы проектирования и конструирования мехатронных технологических машин на базе многосекционных манипуляторов типа "хобот", построенных из механизмов с параллельной структурой.

Проблемы при проектировании больших сборок

В настоящее время модель одной секции манипулятора представляет собой так называемый трипод_3 с тремя подвижностями верхней платформы (рис.1).Эти подвижности можно реализовать с помощью трех активных телескопических штанг, длины которых мы можем менять по заданным законам. Телескопические штанги крепятся к нижнему и верхнему основанию с помощью шарниров Гука (2 вращательные степени свободы) (рис.2).

Кроме трех телескопических тяг верхняя платформа закреплена относительно нижней платформы ещё и центральной штангой. Это также телескопическая штанга, длина которой может меняться, но только под воздействием внешних сил. В нашей терминологии данная штанга считается пассивной. Нижний и верхний концы центральной штанги соединены соответственно с нижней и верхней платформой также при помощи шарнира Гука.

Таким образом, три подвижности верхней платформы – это её перемещение по вертикали (вдоль оси Z), и повороты вокруг осей X и Y. Целиком пятисекционный манипулятор выглядит как на рис. 3.

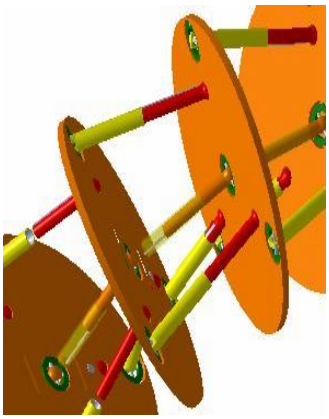


Рисунок 1 - Трипод_3

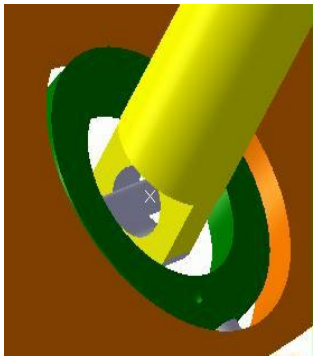


Рисунок 2 - шарнир Гука

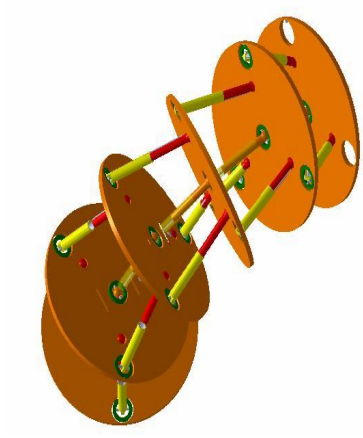


Рисунок 3 - Пятисекционный манипулятор

В данной работе манипулятор, представленный на рис. 3, является сборкой отдельных деталей в системе Catia. Механизм “наращивался” постепенно – деталь за деталью. Нетрудно заметить, что все пять секций имеют схожий вид. Отсюда возникает вопрос, а возможно ли собрать манипулятор в качестве подсборок, где одна секция является одной подсборкой? Если мы сможем это выполнить, то проектирование сократиться по времени в несколько раз и упростить задачу для разработчика.

Проектируя весь механизм в качестве подсборок, я столкнулся с проблемой задания законов кинематики для всего манипулятора в целом. А именно, система Catia создает для каждой подсборки свой механизм для симуляции в течение определенного промежутка времени. И проанализировать весь манипулятор в целом не получается возможным. Данная проблема остается открытой для дальнейшего её изучения.

Ориентация осей координат в сборке

Поскольку необходимо как можно точно получить траекторию движения определенных точек манипулятора, то очень важна одинаковая ориентация манипулятора относительно глобальной системы координат. То есть в системе CATIA нам важно правильно ориентировать сборочную модель манипулятора относительно начала координат сборки.

В CATIA эта задача представляет определенные сложности. Точнее – для подобной ориентации требуется вспомнить некоторые правила построения сборки.

А именно, первую секцию нашего манипулятора нужно ориентировать так, чтобы начало глобальной системы координат совпадало с точкой, представленной на рис. 4 маленьким крестиком.

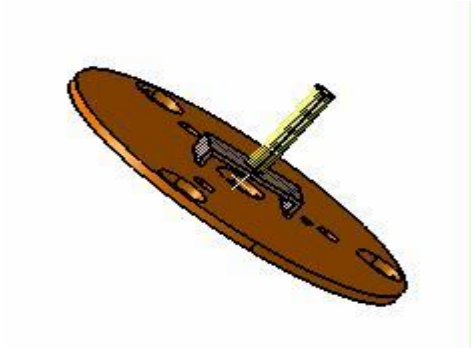


Рисунок 4 - Основание Трипода_3

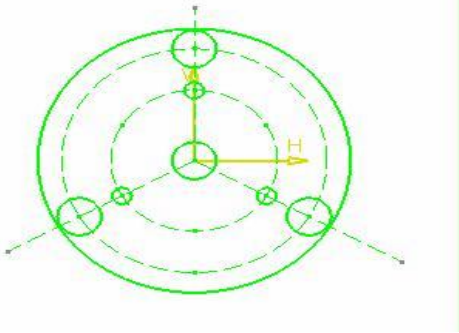


Рисунок 5 - Глобальная система координат

Кроме этого, ориентация осей глобальной системы координат относительно эскиза первой секции должно быть таким, каким оно показано на рис.5. Здесь вертикальная ось V (на рис.5 желтый цвет) соответствует оси Y глобальной системы координат. А горизонтальная ось H соответствует оси X глобальной системы координат.

Установка начальных длин выдвижных штанг

Сама выдвижная штанга представляет собой некоторую трубу, внутрь которой вставлен выдвигающийся шток (рис.6).

Рисунок 6 - Выдвижная штанга

Любая траектория начинается с некоторого начального состояния системы. Это начальное состояние определяется начальными длинами всех пятнадцати штанг манипулятора.

В системе CATIA механизм задания начального положения длины выдвижной штанги определяется так называемым «ручным» режимом, в котором пользователю предоставляется таблица (рис.7), в которой для каждой штанги можно менять длину штанги по своему усмотрению.

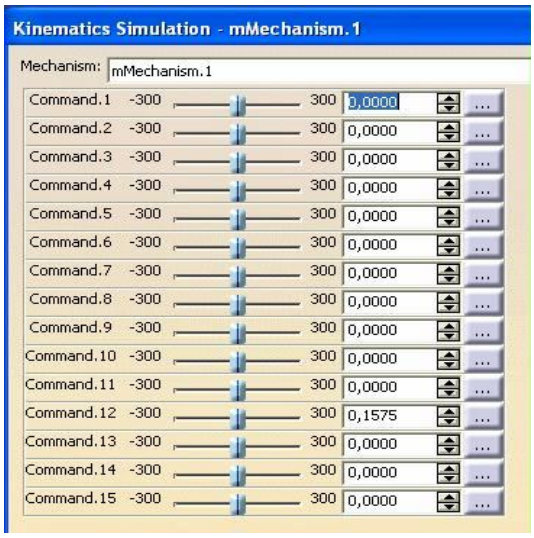


Рисунок 7 - «ручной» режим задания начального положения штанги

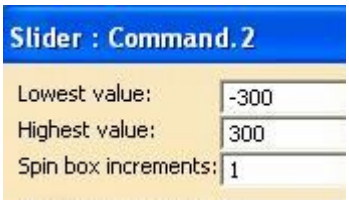


Рисунок 8 - установка точности начального положения штанги

Таким образом, изменение начальной длины проблем не вызывает. Вопрос в другом - с какой точностью мы можем установить начальную длину штанги?

В CATIA эта точность ограничена. А именно – длина штанги задается только с точностью до 1 мм (рис.8). Это является главным источником погрешностей в исследуемых траекториях пятисекционного манипулятора.

Таким образом, в кинематическом анализе конструктивной модели сложного механизма, проведенным с помощью приложения DMU Kinematics тяжелой графической системы CATIA уже заложена некоторая погрешность. Это погрешность находится в пределах одно процента.

Заключение

Представьте, что секций будет не пять, а семь; базовой секцией станет не трипод, а гексапод; а выдвижная штанга будет выполнена не с одним штоком, а с помощью двух – трех телескопически выдвигаемых штоков. Сколько тогда будет подвижных элементов нашего манипулятора? Сколько законов изменения длины придется рассчитать? И сможет ли CATIA справится с такой задачей?

Ответ на этот вопрос можно получить выполнив соответствующие анализы. Но уже сейчас можно перечислить возможные препятствия на пути выполнения подобных экспериментов:

1) Будет очень трудно соединить кинематическими парами сложный механизм. Механизм все время старается «вывернуться» так, что вернуть его в исходное состояние довольно трудно. Приходится разрабатывать особую технику сборки механизма. В частности, придется применить инструмент внешних публикаций. И, тем не менее, придется очень постараться, чтобы получить, наконец, заветное сообщение рис.9.

Рисунок 9 – информационное сообщение о том, что механизм может быть смоделирован

2) Даже если механизм будет собран, и система будет готова анализировать его кинематику, возможны непонятные «клинчи», при вхождении в которые, механизм просто перестает двигаться. И природу этих «клинчей» пока распознать не удалось.

Литература

1. Моделирование в Catia – Уроки [http://www.catiav5.ru/ (дата обращения: 01.12.11)]
2. К. А. Басов. CATIA V5. Геометрическое моделирование// Компьютерная литература , 2008.

ЭКСПЕРТНАЯ СИСТЕМА ПО БАЗЕ ЗНАНИЙ ПРЕДМЕТНОЙ ОБЛАСТИ ТЕХНОЛОГИЙ МИКРО- И НАОСИСТЕМ

Фраерман А.И.
Научный руководитель: доцент, к.т.н., Лавров А.В.
МГТУ им. Н.Э. Баумана, кафедра ИУ4, Москва, РФ

EXPERT SYSTEM ACCORDING TO THE KNOWLEDGE BASE OF SUBJECT DOMAIN OF TECHNOLOGIES MICRO AND NAOSISTEM

Fraerman A.I.
Research advisor: docent, c.e.s.,Lavrov A.V.
Bauman University, Moscow, RF

Аннотация
В данной работе речь идет о разработке экспертных систем для синтеза технических решений в области технологий микро- и наносистем. Для визуализации дерева решений предложена технология CMAP.

Abstract
In the given work it is a question of working out of expert systems for synthesis of technical decisions in the field of technologies micro- and nanosystems. For visualization of a tree of decisions technology CMAP is offered.

Введение

В настоящее время для накопления, систематизации и хранения данных и для оперирования знаниями применяют *базы знаний.* Полноценные базы знаний содержат в себе не только фактическую информацию, но и правила вывода, допускающие автоматический синтез концепций о вновь вводимых фактах и, как следствие, осмысленную обработку информации.

Информация относительно микро и нанотехнологий весьма обширна и известны попытки ее систематизировать, аккумулировать в концентрированном виде и как-то обозреть. Кроме обычных баз данных (информационных, патентных) можно указать на пример построения «дерева нанотехнологий», а также на различные «дорожные карты нанотехнологий». Эти прототипы плохо структурированы и могут решать ограниченное число локальных задач. Ввиду недостатков описанных подходов для хранения и извлечения информации был применен подход с использованием баз знаний и экспертных систем.

1. Анализ предметной области

Проводя декомпозицию первого уровня с использованием концептуальных карт (CMAP) можно выделить следующие основные разделы предметной области (рисунок 1.1): маршруты проектирования микро- и наносистем, маршруты производства микро- и наносистем, компоненты микро- и наносистем, методы измерений параметров микро- и наносистем и системы компьютерного сопровождения жизненного цикла изделий.

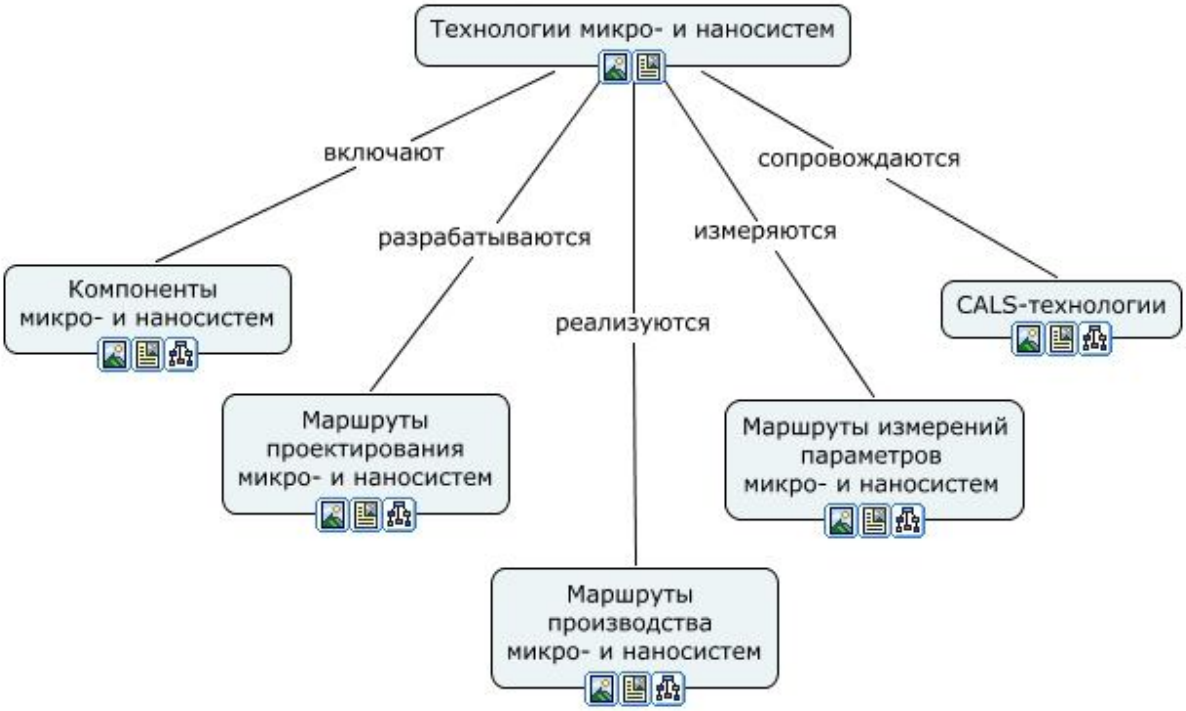


Рисунок 1.1 – СМАР-карта технологий микро- и наносистем

Каждый из блоков на рисунке 1.1 является корневым узлом древовидной семантической структуры для следующих уровней декомпозиции.

Трактовка понятия компоненты микро- и наносистем включает в себя: компоненты, структурные единицы микро и наносистемной техники, назначение, виды преобразования, условия эксплуатации. На рисунке 1.2 приведен пример причинно-следственной связи в виде диаграммы Исикавы.

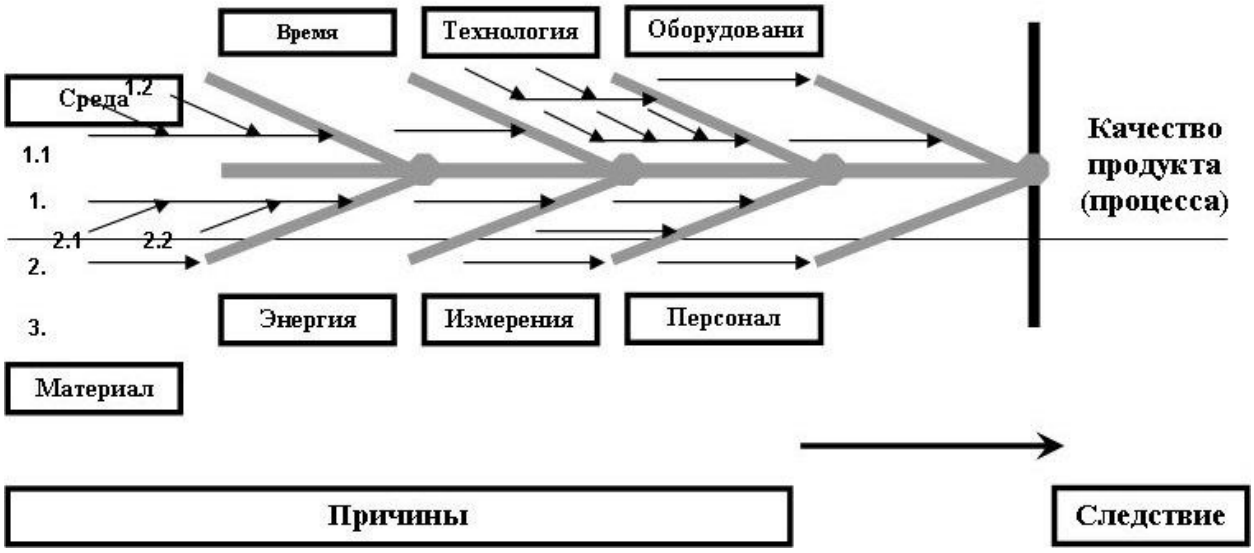


Рисунок 1.2 – Пример анализа причинно-следственных связей для предметной области «Технологии микро- и наносистем»

Маршруты проектирования микро- и наносистем – это последовательность этапов и процедур для представления процесса проектирования микро- и наносистем. Они включают в себя виды обеспечений, определяющих выполнение проектных процедур.

Маршруты производства микро- и наносистем – последовательность работ по непосредственному изготовлению микро- и нанообъектов. Маршруты производства микро- и наносистем включают в себя общую технология, технологические процессы, моделирование технологических процессов.

Методы измерений параметров микро-и наносистем – систематизированная совокупность шагов, действий, необходимых для исследования микро- и наносистем.

CALS технологии – совокупность программных компонентов направленных на информационное сопровождение жизненного цикла изделий.

При разработке СМАР карт предметной области изначально проводится анализ причинно-следственных связей - таких связей между явлениями (объектами), при которых одно явление, называемое причиной, при наличии определенных условий с необходимостью порождает или вызывает к жизни другое явление, называемое следствием.

2. Реализация базы знаний

В качестве эффективного инструмента, обеспечивающего единое понимание концептуальных положений области технологий микро- и наносистем предлагается использовать средства интерактивного диалога по фреймовому семантическому дереву онтологии предметной области и визуальную семантическую навигацию посредством понятийных СМАР карт по расширенному визуальному дереву онтологии.

Роль онтологии области технологий микро- и наносистем в обеспечении согласованного понимания предметной области достаточно наглядно демонстрирует модель передачи знаний между двумя специалистами, представленная на рисунке .

Так как технология - это кодифицированные знания, то есть представленные в виде текста, формализованных записей, электронных моделей, то передача знаний в предлагаемой модели осуществляется посредством документа.

Первый специалист является автором документа, второй - его потребителем. Документ научно-технического характера содержит определенные сведения в области технологий.

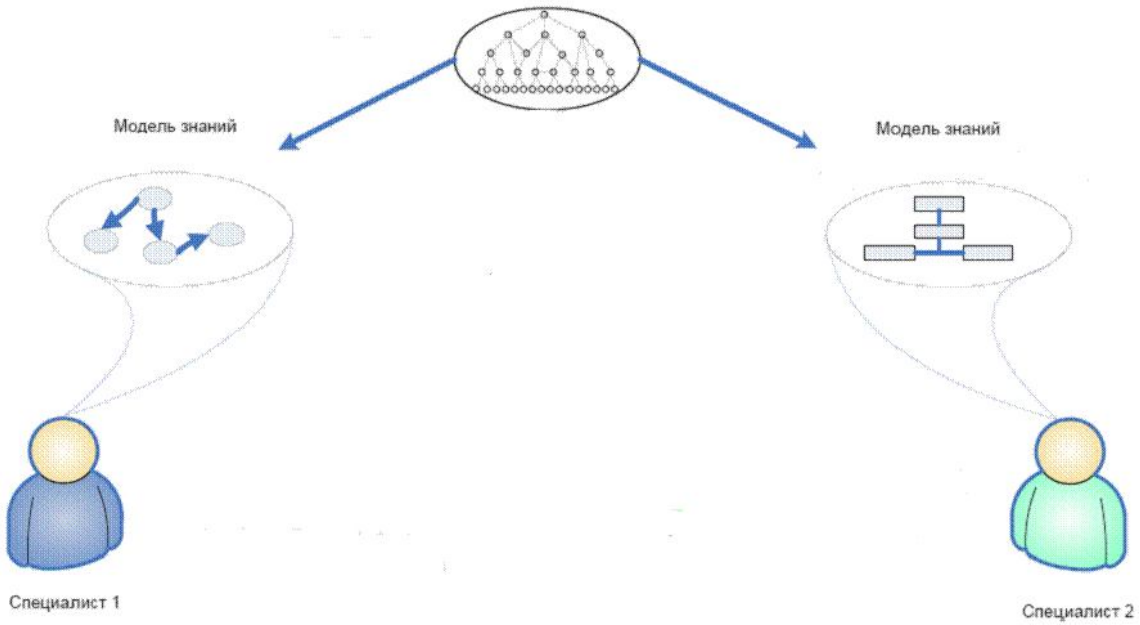


Рисунок 2.1 – Схема модели передачи знаний между двумя экспертами

В этой связи особый интерес для реализации оперативного доступа, поддержки развития онтологии области технологий микро- и наносистем представляет технология CMAP Tools, который обеспечивает:

- коллективную разработку онтологии;
- создание онтологии на локальном рабочем месте;
- различные режимы просмотра онтологии (граф, таблица, иерархический список);
- формирование и выдачу статистических данных по онтологии (число концептов, отношений);
- экспорт и импорт онтологии в различные форматы представления (растровое изображение, Web-страница, текстовый документ, PDF, языки описания онтологий);
- формальную оценку корректности онтологии;
- ведение структурированного репозитария онтологических моделей;
- подключение к концептам ссылок на различные ресурсы (Web-сайты, документы, изображения, другие онтологии, модели).

Существует фреймовая модель семантической сети. Фреймовая модель основана на концепции Марвина Мински (Marvin Minsky) – профессора Массачусетского технологического института, основателя лаборатории искусственного интеллекта, автора ряда фундаментальных работ. Фреймовая модель представляет собой систематизированную психологическую модель памяти человека и его сознания.

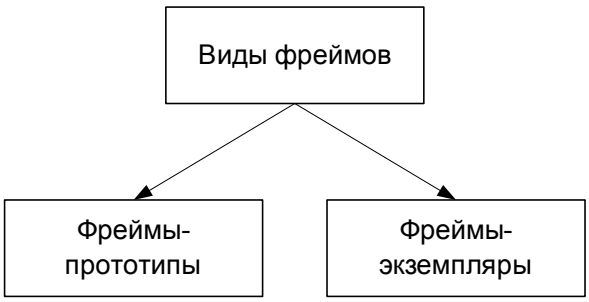


Рисунок 2.2 – Виды фреймов предметной области технологий микро- и наносистем

Фрейм (англ. frame – рамка, каркас) – структура данных для представления некоторого концептуального объекта. Информация, относящаяся к фрейму, содержится в составляющих его слотах.

Различают следующие виды фреймов: фреймы-прототипы фреймы-экземпляры, фреймы-структуры, фреймы-роли, фреймы-сценарии, фреймы-ситуации. Система связанных фреймов может образовывать семантическую сеть. Применяются фреймы в экспертных системах и других интеллектуальных системах различного назначения. Фреймы, которые используются для отображения онтологии предметной области технологий микро- и наносистем представлены на рисунке 2.2.

3. Методика применения базы знаний для синтеза новых технических решений

Созданная онтология с помощья концептуальных карт не может быть прямо перенесена в экспертную систему в том виде, в котором это понятно человеку, она является средством визуализации знаниевых компонентов. Для формализованного описания знаний предлагается логическая реляционная модель. Сформированная модель, позволяет приблизить созданную онтологию к машиному языку, хранить её в базе данных и, следовательно, применять известные алгоритмы поиска решений.

Выявленная древовидная структура знаний представляет собой иерархическую структуру фреймов. Каждый фрейм, находящийся на верхнем уровне иерархии для каждой из декомпозиций содержит общий набор слотов в виду того, что фреймы обладают

способностью наследовать значения характеристик своих родителей, находящиеся на более высоком уровне.

Типовая статическая ЭС состоит из следующих основных компонентов (рисунок 3.1):

- решателя (интерпретатора);
- рабочей памяти (РП), называемой также базой данных (БД);
- базы знаний (БЗ);
- компонентов приобретения знаний;
- объяснительного компонента;
- диалогового компонента.

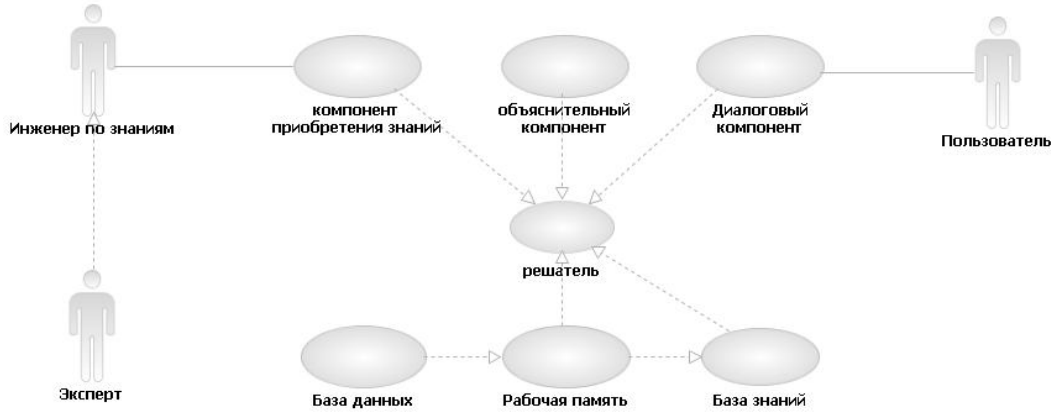


Рисунок 3.1 – Компоненты статической ЭС

Концепция построения базы знаний заключается, главным образом, в изучении самой предметной области микро- и наносистем, накоплении знаний, понятий в этой области и построение древовидных структур знаний. Накопление знаний – является основной задачей экспертов. В общей теории систем доказано, что большинство систем могут быть декомпозированы на базовые представления подсистем. К ним относят: последовательное (каскадное) соединение элементов, параллельное соединение элементов, соединение с помощью обратной связи. В связи с этим при построении древовидной структуры знаний помимо связей выявляется декомпозиция для упрощения структуры знаний.

Заключение

Результатом проведенной работы является созданная и развернутая на «Вычислительном кластере ННС» экспертная система по базе знаний предметной области технологий микро- и наосистем для целей анализа методов и технологий, а также сравнения научно-технических решений в указанной области.

Разработанный программный комплекс экспертной системы по предметной области технологий микро- и наносистем обеспечивает механизм интеллектуального синтеза решений и визуальной навигации по семантической информации в области технологий микро- и наносистем, реализованной на платформе тонкого клиента.

Литература

1. Информационные технологии в конструировании и технологии микроэлектроники/В.П. Григорьев, Ю.И. Нестеров, Д.В. Черепанов. - М.: Изд-во МГТУ им.Н.Э.Баумана. 2000. - 39 с.

2. Применение методов искусственного интеллекта в САПР технологического проектирования производства электронной аппаратуры/В.П. Григорьев, Э.Н. Камышная, Ю.И. Нестеров, С.А. Никитин - М.: Изд-во МГТУ им.Н.Э.Баумана. 1998. - 48 с.

СЕНСОРНАЯ СИСТЕМА ОБРАБОТКИ КАЛЛИГРАФИИ РУКОПИСНЫХ СИМВОЛОВ

Демин А.А.
Научный руководитель: к.т.н., доцент кафедры ИУ4 Власов А.И.
МГТУ им. Н.Э.Баумана, Москва, Россия

TOUCH PROCESSING SYSTEM OF THE HAND-WRITTEN CALLIGRAPHY SYMBOLS

Demin A.A.
Supervisor: c.t.s., associate professor of IU4 department., Vlasov A.I.
MSTU, Moscow, Russia

Аннотация
В статье рассматриваются программно-аппаратная сенсорная система обработки каллиграфии рукописных символов, позволяющая проводить тренировку написания символов. Подробно исследованы алгоритмы анализа растровых изображений, распознавания рукописного текста и интерфейсы взаимодействия с пользователем. В заключении даны рекомендациии по реализации оптимизированных методов получения и обработки графических образов в реальном времени.

Abstract
In article are considered hardware-software touch system of processing of a calligraphy of the hand-written symbols, allowing to carry out training a spelling of symbols. Analysis algorithms of raster images, recognition of the hand-written text and interfaces of interaction with the user are in detail investigated. In the conclusion recommendations on realization of the optimized methods of reception and processing of graphic images in real time are given..

Введение

В настоящее время, при стремительном росте информационных технологий, все больше информации хранится в ЭВМ и вводится непосредственно с клавиатуры и мыши. В связи с этим у многих людей уменьшается скорость написания, ухудшается почерк, и постепенно теряются навыки письма. Но одновременно с этим высока доля обмена рукописными текстами, такими как: элементы деловой или служебной переписки, формы страховых и кредитных организаций, журналы и заметки. Существующие системы автоматизированной обработки таких текстов либо ограниченны в возможностях применения, либо критичны к качеству начертания символов. Поэтому актуальным является разработка устойчивой системы позволяющей выполнять оперативное распознавание и интерпретацию рукописных данных и проводить корректировку почерка, тренировку правильного и быстрого письма.

Методы и алгоритмы обработки графических образов

Систему распознавания рукописных данных можно представить как комбинацию трех подсистем (рис. 1): подсистемы извлечения признаков, непосредственно подсистемы распознавания и подсистемы принятия решения, к какому классу относится данный объект.

Разработка системы ввода текста, её тестирование и отладка, невозможна без создания базы данных символов, в общем случае состоящей из:

- неконфликтных символов (отличающихся по форме написания) в хорошем исполнении;
- конфликтные символы в хорошем исполнении;
- символы с нарушенными требованиями к качеству написания.

Открытых баз не существует, поэтому при разработке системы ввода рукописного текста приходится создавать и наполнять собственную базу символов.

В связи с этим было реализованы следующие алгоритмы:

1. Приведение изображения к серой шкале. В связи с этим изображение можно привести в представление в виде серой шкалы. Это означает, что каждый пиксель изображения будет представлен как число лежащее в интервале [0..255]. Этого достаточно для того, чтобы сохранить все данные с наименьшей их потерей при переходе к серой шкале.

2. Приведение изображения к черно-белому виду;

3. Анализ и удаление шумов на изображении не относящихся к тексту;

Шумоподавление – это удаление возможного зашумления изображения, являющегося следствием погрешности устройства ввода изображение (например, сканера).

Реализован алгоритм сглаживания с использованием функции Гаусса. Дискретное гауссово ядро сглаживания (апертуру фильтра) можно получить (Рис. 1), построив массив размером (2k + 1) x (2k + 1), значение элемента (i, j) которого равно

$$H_{ij} = \frac{1}{2\theta\sigma^2}\exp\left(-\frac{((i-k-1)^2+(j-k-1)^2)}{2\sigma^2}\right),$$

где σ– это среднеквадратическое отклонение гауссиана.

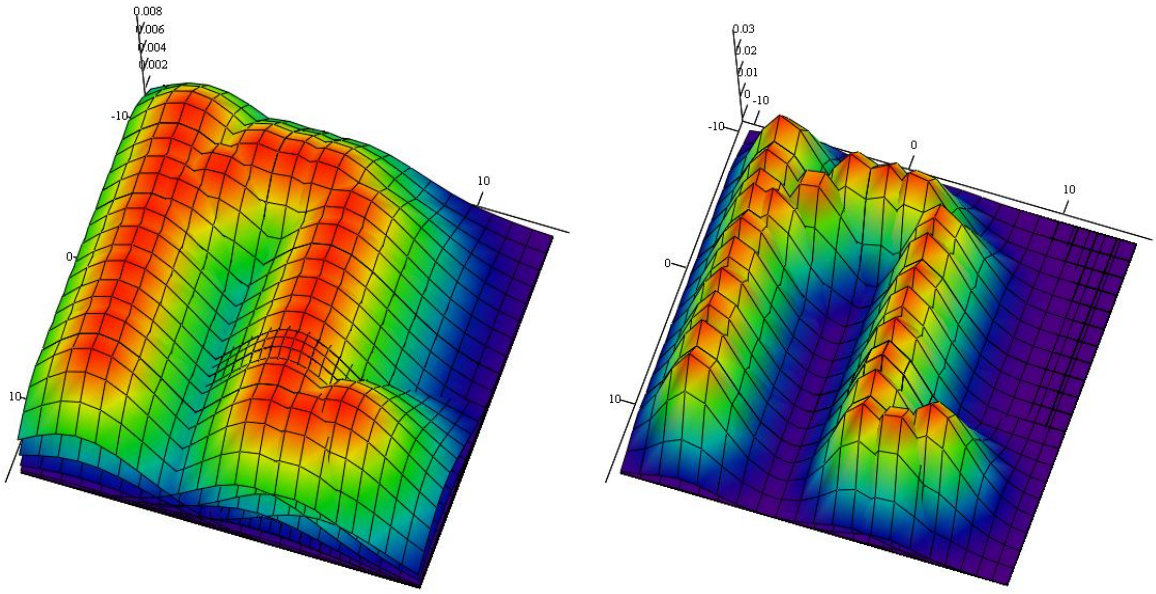


Рисунок 1 - Двумерное нормальное распределение с разными весовыми коэффициентами

Название ядра объясняется тем, что именно такой вид имеет плотность вероятности для двумерной нормальной (или гауссовой) случайной переменной с заданной ковариантностью. Данное ядро сглаживания образует такое взвешенное среднее, для которого в центре ядра весовые коэффициенты пикселей намного больше, чем на его границах.

Этот подход можно обосновать качественно: сглаживание подавляет шум, поддерживая требование, чтобы пиксели были похожи на своих соседей. Уменьшая весовые коэффициенты для отдалённых пикселей, можно быть уверенным, что для них это требование будет не таким жёстким.

4. Скелетизация черно-белого изображения (рис. 2). Скелет текста, получаемый из растрового, черно-белого изображения, представляет собой наборы линий толщиной в один пиксель. В общем случае, суть скелетизации – приведение данных к такому виду, чтобы затем можно было получить скелет изображения в его векторном представлении и привести тем самым объем данных к тому минимуму, который необходим для процесса распознавания.

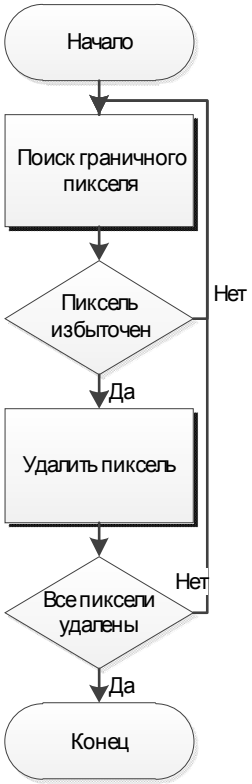


Рисунок 2 – Алгоритм составления скелета рукописных данных

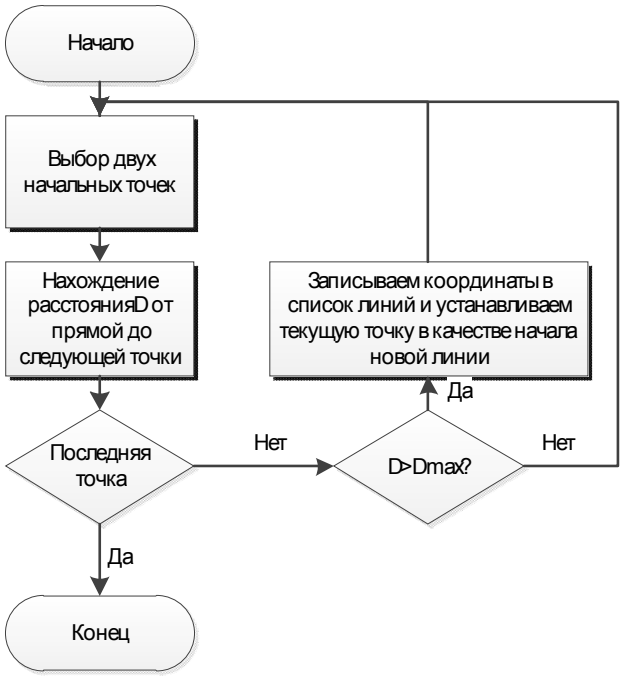


Рисунок 3 – Алгоритм векторизации изображения

5. Векторизация изображения (рис 3). Вторичная векторизация или, еще ее можно назвать аппроксимацией векторизованных данных, необходима для того, чтобы избавиться от избыточности ранее полученных данных. Избыточность эта состоит в том, что некоторые точки кривых не являются существенными. Это те точки, избавившись от которых легко, при необходимости, можно их вычислить и восстановить.

6. Выделение слов на изображении.
7. Выделение строк.
8. Определение угла наклона
9. Исправление погрешности скелетизации

Предложенный способ позволяет разложить тонкий контур только на набор связанных прямых, что является единственным существенным недостатком метода. Среди преимуществ метода можно выделить небольшие вычислительные затраты и простота реализации.

Заключение

Разработанные математические модели, структурные схемы и алгоритмы обработки образов рукописного текста, позволяют анализировать векторные представления графических образов в режиме реального времени. Программные алгоритмы классификации рукописных символов удовлетворяют поставленным требованиям в работе даже на низкопроизводительных системах на базе процессоров Intel Atom с пониженной тактовой частотой до 1 ГГц.

Преимуществом программно-алгоритмического обеспечения системы распознавания образов является алгоритм оценки рукописного текста, позволяющий оценить скорость письма, качество каллиграфии, и прогресс улучшения основных показателей обучения.

Литература

1. Шикин А. В., Боресков А. В. Компьютерная графика. Полигональные модели. – М.: Диалог-МИФИ, 2000. – 464 с.
2. Л.М. Местецкий Непрерывный скелет бинарного изображения. - Доклад на конференции Графикон, 2006. – 24 с.: ил.
3. Е.А. Бутаков, В.И. Островский, И.Л. Фадеев Обработка изображений на ЭВМ. – М.: Радио и связь. 1987.
4. Ф. Препарата, М. Шеймос Вычислительная геометрия: Введение. – М.: Мир. 1989.
5. Костевич Л.С. Математическое программирование: Информ. Технологии оптимальных решений: Учеб. Пособие / Л.С. Костевич. – Мн.: Новое знание, 2003. – 424 с.: ил.

Работа выполнена при частичной поддержке ФСР (грант 9533р/10375).

АДАПТИВНЫЙ МЕТОД РЕШЕНИЯ МНОГОКРИТЕРИАЛЬНОЙ ЗАДАЧИ ОПТИМИЗАЦИИ НА ОСНОВЕ АППРОКСИМАЦИИ ФУНКЦИИ ПРЕДПОЧТЕНИЯ ЛИЦА, ПРИНИМАЮЩЕГО РЕШЕНИЯ

Мухлисуллина Д.Т.
д.ф.-м. н., профессор Карпенко А.П.
МГТУ им. Н.Э.Баумана, кафедра РК6, Москва, Российская Федерация

ADAPTIVE METHOD FOR MULTIOBJECTIVE OPTIMIZATION PROBLEM SOLVING BASED ON APPROXIMATION OF THE DECISION MAKER'S UTILITY FUNCTION

Mukhlisullina D.T.
Professor, Karpenko A.P.
Bauman Moscow State Technical University, Moscow, Russian Federation

Аннотация
Рассматривается прямой адаптивный метод многокритериальной оптимизации на основе аппроксимации функции предпочтения лица, принимающего решение, с помощью нейронных сетей, аппарата нечеткой логики и нейро-нечеткой системы. Приводятся результаты исследования эффективности метода при решении тестовых многокритериальных задач оптимизации. Программно метод реализован в виде дополнительного модуля программного комплекса PRADIS, предназначенного для анализа динамики систем различной физической природы. Практическая апробация метода выполнена при решении двухкритериальной задачи оптимизации механической подсистемы двигателя внутреннего сгорания.

Abstract
We consider the direct adaptive method for multobjective optimization problem solving based on approximation of the decision maker's utility function by using neural networks, fuzzy and neuro-fuzzy systems. Research results of the method effectiveness for solving test multiobjective optimization problems are presented. This method was developed as an additional optimization module for PRADIS program which is for analysis of dynamic systems of different physical nature. This method was applied for solving two-dimensional optimization problem of combustion engine mechanical subsystem.

Постановка задачи. Рассматриваем задачу многокритериальной оптимизации (МКО-задача) вида

$$\min \Phi(X) = \Phi(X^*), \; X \in D_X, \tag{1}$$

где X – вектор варьируемых параметров задачи; D_X – ограниченное и замкнутое множество допустимых значений вектора X; $\Phi(X)$ – векторный критерий оптимальности $\Phi(X) = (\phi_1(X), \ldots, \phi_m(X))$. В этой задаче лицо, принимающее решение (ЛПР), стремится найти такой вектор X^* – искомое решение МКО-задачи – которое минимизирует на множестве D_X каждый из частных критериев оптимальности. Предполагаем, что частные критерии оптимальности тем или иным способом нормализованы.

Методы решения МКО–задачи чрезвычайно разнообразны. По содержанию и способу формализации дополнительной информации о предпочтениях ЛПР эти методы можно разделить на следующие классы: методы, не учитывающие предпочтения ЛПР; априорные методы; апостериорные методы; адаптивные (интерактивные) методы [1]. Последний класс методов является наиболее перспективным в силу нескольких причин. Во-первых, адаптивные методы требует меньше вычислительных ресурсов по сравнению с апостериорными методами, в которых необходимо построить аппроксимацию всего множества достижимости. Очевидно, в задачах САПР, где вычисление каждого критерия оптимальности часто состоит в решении системы обыкновенных дифференциальных уравнений или дифференциальных уравнений в частных производных, применение апостериорных методов сильно ограничено. Во-вторых, каждая итерация адаптивных

методов включает в себя фазу анализа, выполняемую ЛПР, и фазу расчетов, выполняемую программной системой многокритериальной оптимизации (МКО-системой). Поэтому на каждой итерации ЛПР может переоценить ранее введенные оценки в отличие от априорных методов, в которых ЛПР должно априори (до начала решения задачи) в том или ином виде ввести в МКО-систему свои предпочтения (весовые коэффициенты критериев оптимальности, отражающие их относительную важность; желаемые значения частных критериев и др.).

Предлагаемый адаптивный метод решения МКО-задачи (1) основан на том предположении, что на множестве D_X существует функция предпочтений ЛПР $\psi(X)$ (пси), которая отображает это множество во множество действительных значений, т.е. $\psi : X \to R^1$. Более строго, МКО-система преобразует введенные ЛПР значения лингвистической переменной $\psi(X)$ в действительные числа, принадлежащие диапазону [1:9]. В результате МКО-задачу сводят к задаче поиска вектора *X*, который максимизирует функцию предпочтений ЛПР:

$$\max \psi(X) = \psi(X^*) = \psi^* .$$

Обозначим $\varphi(X, \Lambda)$ операцию свертки частных критериев оптимальности, где $\Lambda \in D_\Lambda \subset R^m$ - вектор весовых множителей; $D_\Lambda = \{\lambda_i \mid \lambda_i \geq 0, \sum_i \lambda_i = 1, \ i \in [1:m]\}$- множество допустимых значений этого вектора. Для любого $\Lambda \in D_\Lambda$ метод скалярной свертки сводит решение задачи (1) к решению однокритериальной задачи глобальной условной оптимизации (ОКО-задачи)

$$\min \varphi(X, \Lambda) = \varphi(X^*, \Lambda), \ X \in D_X . \tag{2}$$

Условие (2) позволяет полагать [1], что функция предпочтений ЛПР определена не на множестве D_X, а на множестве D_Λ, т.е. $\psi : \Lambda \to R^1$. В результате МКО-задачу сводят к задаче поиска вектора $\Lambda^* \in D_\Lambda$ такого, что

$$\max \psi(\Lambda) = \psi(\Lambda^*) = \psi^* . \tag{3}$$

Метод решения задачи. Общая схема рассматриваемого прямого адаптивного метода решения МКО-задачи является итерационной и состоит из перечисленных ниже основных этапов [2].

Этап «разгона» метода. МКО-система некоторым образом (например, случайно) последовательно генерирует k векторов $\Lambda_1, \ldots, \Lambda_k$ и для каждого из этих векторов Λ_i выполняет следующие действия. 1) Решает соответствующую ОКО-задачу (2). 2) Предъявляет ЛПР найденное решение X_i^*, а также соответствующие значение всех частных критериев оптимальности $\Phi(X_i^*)$. 3) ЛПР оценивает эти данные и вводит в МКО-систему соответствующее значение своей функции предпочтений $\psi(\Lambda_i)$.

Первый этап. На основе всех имеющихся в МКО-системе значений векторов $\Lambda_1, \ldots, \Lambda_k$ и соответствующих оценок функции предпочтений $\psi(\Lambda_1), \ldots, \psi(\Lambda_k)$, МКО-система выполняет следующие действия. 1) Строит функцию $\overline{\psi}_1(\Lambda)$, аппроксимирующую функцию $\psi(\Lambda)$ в окрестности точек $\Lambda_1, \ldots, \Lambda_k$. 2) Отыскивает максимум функции $\overline{\psi}_1(\Lambda)$ – решает ОКО-задачу (3). 3) С найденным вектором Λ_1^* решает ОКО-задачу вида (2) – находит вектор параметров и соответствующие значения частных критериев оптимальности, а затем предъявляет их ЛПР. ЛПР оценивает указанные данные и вводит в систему соответствующее значение своей функции предпочтений $\psi(\Lambda_1^*)$.

Второй этап. На основе всех имеющихся в МКО-системе значений векторов $\Lambda_1, \ldots, \Lambda_k, \Lambda_1^*$ и соответствующих оценок функции предпочтений $\psi(\Lambda_1), \ldots, \psi(\Lambda_k), \psi(\Lambda_1^*)$ МКО-система

строит функцию $\overline{\psi}_2(\Lambda)$, аппроксимирующую функцию $\psi(\Lambda)$ в окрестности точек $\Lambda_1,\ldots,\Lambda_k,\Lambda_1^*$ и т.д. по схеме первого этапа до тех пор, пока ЛПР не примет решение о прекращении вычислений.

Можно предложить несколько вариантов аппроксимации функции предпочтения ЛПР. В работе рассматриваем нейросетевую аппроксимацию с использованием персептронной нейронной сети (*MLP*-сеть) и нейронной сети на основе радиальных базисных функций (*RBF*-сеть); аппроксимацию на основе аппарата нечеткой логике, а также на основе нейро-нечеткой системы (*ANFIS*) [2-4].

Исследование эффективности метода. Исследование выполнено для двух двухкритериальных и одной трехкритериальной тестовых задач оптимизации [2-4].

Двухкритериальная задача 1, имеющая выпуклый фронт Парето,

$$\phi_1(X)=(x_1-5)^2+x_2^2+10;\quad \phi_2(X)=x_1^2+(x_2-5)^2+20;\quad D_X=\{X\mid 0\le x_i\le 5, i=1,2\}.$$

Двухкритериальная задача 2 (невыпуклый многосвязный фронт Парето)

$$\phi_1(X)=-x_1;\quad \phi_2(X)=-f(X)\left(1-\sqrt{\frac{x_1}{f(X)}}-\frac{x_1}{f(X)}\sin(10\pi x_1)\right);$$

$$f(X)=1+9x_2;\ D_X=\{X\mid 0\le x_i\le 1, i=1,2\}.$$

Трехкритериальная задача 3 (выпуклый фронт Парето)

$$\phi_1(X)=(x_1-5)^2+x_2^2+x_3^2+10;\quad \phi_2(X)=x_1^2+(x_2-5)^2+x_3^2+20;$$

$$\phi_3(X)=x_1^2+x_2^2+(x_3-5)^2+5;\ D_X=\{X\mid 0\le x_i\le 5, i=1,2,3\}.$$

Результаты исследования оценивались по ошибке аппроксимации, которая вычислялась по формуле $\varepsilon_a=\frac{1}{N}\sum_{i=1}^{N}\left|\psi(\Lambda_i)-\overline{\psi}(\Lambda_i)\right|$, где N – номер итерации, а также времени, затраченному на построение аппроксимирующей функции. Таким образом, ошибка аппроксимации ε_a во всех случаях находилась в пределах от 0.3 до 0.7. Исключением являются *RBF*–сети, при использовании которых ошибка достигала значения ~200 (что связано с плохими экстраполяционными свойствами этих сетей). Минимальную погрешность аппроксимации обеспечила нейро-нечеткая система *ANFIS*. По сравнению с нейросетевой и нейро-нечеткой аппроксимацией функции предпочтений ЛПР, решение задачи с помощью нечеткой логики требует большего времени вычислений. Так, время обучения нейронной сети во всех экспериментах не превышало в среднем двух секунд, в то время как выполнение тонкой настройки нечеткой модели в среднем составило 20 секунд.

Прикладная задача. Рассмотренный адаптивный метод многокритериальной оптимизации интегрирован в программный комплекс анализа динамики систем различной физической природы – *PRADIS*. Реализация метода выполнена на языке *Python* комплекса *PRADIS* с использованием библиотеки *FANN*, реализующей нейронные сети [2].

Заметим, что функция предпочтений ЛПР заранее неизвестна и может быть приближенно сформирована только в процессе диалога ЛПР с МКО-системой, ЛПР в процессе диалога может переоценить результаты предыдущих решений, эти оценки могут быть противоречивы. Поэтому разработанное программное обеспечение предоставляет ЛПР механизм просмотра и корректировки предыдущих результатов, например, ЛПР может понизить или повысить введенные ранее оценки, «откатиться» на заданное число итераций и пр.

Практическая апробация метода выполнена на задаче двухкритериальной оптимизации двигателя внутреннего сгорания (ДВС). С этой целью средствами комплекса PRADIS разработана параметризованная трехмерная модель механической подсистемы ДВС. Модель может быть использована для исследования работы одно-, двух- и четырехцилиндрового рядного автомобильного двигателя внутреннего сгорания с верхним расположением газораспределительного механизма [2].

В качестве варьируемых параметров X задачи рассматривается ход поршня S и диаметр цилиндра D, так что множество допустимых значений вектора варьируемых параметров $D_X = \{X \mid 0.06 \le S \le 0.15, 0.06 \le D \le 0.15\}$. Критериями оптимальности являются максимальный крутящий момент $\phi_1(X)$ [Нм] и усилие реакции в шатунном подшипнике $\phi_2(X)$ [Н].

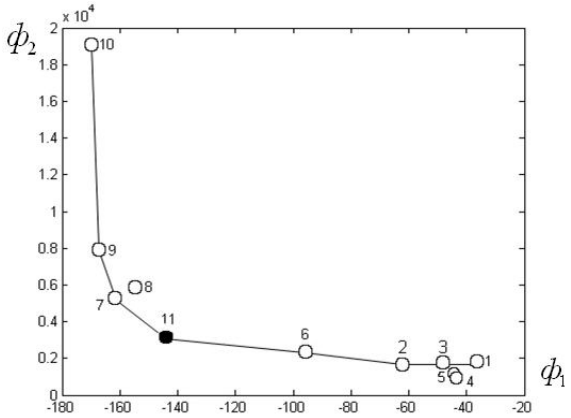


Рисунок 1 – Результаты итераций в пространстве критериев: задача двухкритериальной оптимизации ДВС; *MLP*-сеть с 5-ю нейронами в скрытом слое

Для аппроксимации функции предпочтения ЛПР использовалась *MLP*-сеть с 5-ю нейронами в скрытом слое. В качестве ЛПР выступал эксперт в области ДВС. Итерационный процесс иллюстрирует рисунок 1, на котором точка 11 соответствует решению задачи $S^* = 0.118$ [м], $D^* = 0.123$ [м], $\phi_1^* = 143.92$ [Нм], $\phi_2^* = 3049.20$ [Н]. Заметим, что точки 1 – 11 на рисунке 1 лежат на фронте Парето рассматриваемой МКО-задачи (с погрешностями, обусловленными погрешностями вычислений).

Выводы

Разработан адаптивный метод решения МКО–задачи, основанный на нейросетевой, нечеткой и нейро-нечеткой аппроксимации функции предпочтений ЛПР. Выполнено исследование эффективности метода, показавшее перспективность его развития. Метод реализован в качестве дополнительного модуля программного комплекса *PRADIS*. Практическая апробация метода и разработанного программного обеспечения выполнены при решении задачи двухкритериальной оптимизации механической подсистемы ДВС.

Литература

1. Branke J., Deb K., Miettinen K., Slowinski R., Eds., Multiobjective Optimization: Interactive and Evolutionary Approaches. Berlin/Heidelberg, Germany: Springer-Verlag, 2008.
2. А.П.Карпенко, Д.Т.Мухлисуллина, В.А. Овчинников. Нейросетевая аппроксимация функции предпочтений лица, принимающего решения, в задаче многокритериальной оптимизации //Информационные технологии, 2010, №10, с.2-9.
3. А.П. Карпенко, Д.А, Моор, Д.Т. Мухлисуллина. Многокритериальная оптимизация на основе нечеткой аппроксимации функции предпочтений лица, принимающего решения //Наука и образование: электронное научное издание. Инженерное образование, 2010, №1 (http://technomag.edu.ru/doc/135375.html).
4. А.П. Карпенко, Д.А. Моор, Д.Т. Мухлисуллина. Многокритериальная оптимизация на основе нейро-нечеткой аппроксимации функции предпочтений лица, принимающего решения. //Наука и образование: электронное научное издание. Инженерное образование, 2010, №6 (http://technomag.edu.ru/doc/143964.html).

МОДЕЛИРОВАНИЕ HEMT-ТРАНЗИСТОРА В ПРОГРАММНОМ КОМПЛЕКСЕ TCAD SENTAURUS

Паршина А.А.
Научный руководитель: д.т.н., профессор Зинченко Л.А.
МГТУ им. Н.Э.Баумана, кафедра ИУ4, Москва, Российская Федерация

SIMULATION HEMT-TRANSISTOR in TCAD SENTAURUS

Parshina A.A.
Supervisor: Dr., Prof., Zinchenko L.A.
MSTU, Moscow, Russia

Аннотация
В статье рассматриваются вопросы моделирования генераторного СВЧ-транзистора, изготовленного на основе псевдоморфной гетероструктуры. Подробно исследовано моделирование технологического процесса изготовления транзистора и создание комплекта фотошаблонов в TCAD Sentaurus. В заключении описаны последующие этапы работы.

Abstract
The article deals with the modeling of microwave generating transistor, manufactured by pseudomorphic heterostructures. Studied in detail modeling the process of manufacturing the transistor and the creation of a set of photomasks in TCAD Sentaurus. In conclusion, described the following steps.

Введение

Развитие радиолокационной, связной и навигационной техники требует создания дискретных полупроводниковых приборов, применение которых позволит повысить энергетический потенциал твердотельной передающей РЭА, а, следовательно, дальность ее действия, разрешающую способность и надежность функционирования.

Наиболее широко применяемым в настоящее время материалом для создания активных элементов передающих устройств в сантиметровых и миллиметровых диапазонах длин волн является арсенид галлия. Гетероструктурные полевые транзисторы с модулированным легированием (ПТМЛ, modulation doped field effect transistor, MODFET) на основе тройных и четверных соединений полупроводниковых материалов групп $A^{III}B^{V}$ в настоящее время являются самыми быстродействующими полевыми транзисторами, позволяя одновременно достигать наименьшие коэффициенты шума в гигагерцовом диапазоне частот. Высокое быстродействие достигается за счет эффекта увеличения дрейфовой скорости электронов, образующих двумерный электронный газ. Поэтому другими названиями, отражающими особенности работы ПТМЛ являются транзистор с высокой подвижностью электронов (high electron mobility transistor, HEMT) и транзистор с двумерным электронным газом (two-dimensional electron gas transistor, TEGFET) [7].

Принцип работы HEMT-транзистора и особенности его структуры

HEMT-транзистор - полевой транзистор, в котором для создания канала вместо легированной области, в отличие от обычных МОП-транзисторов, используется контакт двух полупроводниковых материалов с различной шириной запрещенной зоны (т. н. гетеропереход). Общеиспользуемым материалом для HEMT является комбинация GaAs и AlGaAs, хотя возможны и значительные вариации в зависимости от назначения устройства. Например, приборы с повышенным содержанием индия в общем случае показывают лучшую производительность на высоких частотах, в то время как в последние годы наблюдается массовый рост научно-исследовательских разработок HEMT на нитриде галлия (GaN), в связи с их лучшей производительностью при высоких мощностях. Существует достаточно

много изоструктурных аналогов GaAs — полупроводниковых материалов, имеющих близкий к GaAs шаг кристаллической решетки. Это позволяет использовать GaAs в качестве основы для создания широкого класса гетероструктурных транзисторов, обладающих уникальными характеристиками.

Обычно для создания гетероперехода выбираются материалы с одинаковым параметром кристаллической решётки (расстояниями между атомами). В реальности практически невозможно подобрать пару разных полупроводников, у которых было бы идеальное согласование и кристаллических структур, и коэффициентов термического расширения[5]. Поэтому на границе раздела гетероперехода обычно возникают механические напряжения, вызывающие появление дислокаций несоответствия, создающих на границе раздела граничные состояния. Даже у такой хорошо согласующейся пары как Ge и GaAs присутствует пластическая деформация. Поэтому для создания структуры используются твердые растворы. Например, замена Ge на твердый раствор $Ge_{0,98}Si_{0,02}$ приводит к снижению напряжений на границе раздела до уровня, исключающего возможность пластической деформации GaAs, и улучшает характеристики гетероперехода — у него резко уменьшается обратный ток.

В HEMT транзисторах чаще всего применяют гетеропереход GaAs-$Al_xGa_{1-x}As$ [6]. Величина x показывает относительное содержание Al. С ростом x плавно увеличивается ширина запрещенной зоны $\Delta E_з$ $Al_xGa_{1-x}As$. Например, при изменении x в пределах 0-0,4 ширина запрещенной зоны $\Delta E_з$ линейно увеличивается от 1,42 до 1,92 эВ. Для состава с x = 0,3 $\Delta E_з$ = 1,8 эВ и различие в ширине запрещённой зоны составляет ~0,38 эВ. Вследствие хорошего соответствия кристаллических решёток GaAs и $Al_xGa_{1-x}As$ в гетеропереходе обеспечивается низкая плотность поверхностных состояний и дефектов. По этим причинам для электронов, накопленных в области накопления затвора, в слабых электрических полях достигается очень высокая подвижность, близкая к объёмной подвижности для нелегированного GaAs [$(8..9)\times10^3$ $см^2/В•с$ при T = 300 K]. Причём, эта подвижность резко увеличивается при понижении температуры, так как в нелегированном GaAs преобладает решётчатое рассеяние [1].

Также подвижность электронов в канале увеличивается благодаря дополнительной прослойке (спейсеру) между каналом GaAs и барьером AlGaN. Спейсер — тонкий (несколько нм) разделительный слой нелегированного AlxGa1-xAs. Он способствует лучшему пространственному разделению двумерного электронного газа и рассеивающих центров между нелегированным GaAs и легированным донорами AlxGa1-xAs. Концентрация рассеивающих центров в нелегированном слое ниже, чем в легированном, поэтому подвижность электронов, накопленных в области насыщения затвора, дополнительно увеличивается. В этом случае волновая функция электронов в канале не проникает в барьер, а затухает в спейсере. При этом уменьшается рассеяние носителей на барьере. Введение спейсера также улучшает омический контакт, что приводит к повышению предельной рабочей частоты почти до теоретического предела.

В общем случае для создания проводимости в полупроводниках, в них добавляются легирующие примеси для генерации подвижных электронов. Однако эти электроны замедляются в первую очередь при столкновении с примесями, используемыми для их генерации. В HEMT этого удается избежать за счет того, что электроны с высокой подвижностью генерируются на гетеропереходе в области контакта высоколегированного донорного слоя N-типа с широкой запрещенной зоной (в нашем примере AlGaAs) и нелегированного канального слоя с узкой запрещенной зоной без каких-либо легирующих примесей (в данном случае GaAs). На рис.1 показана зонная диаграмма HEMT-транзистора.

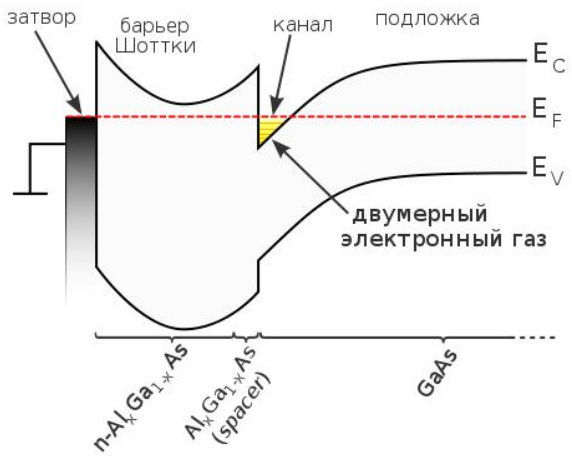


Рисунок 1 - Зонная диаграмма HEMT-транзистора [5]

Электроны, образующиеся в тонком слое n-типа, полностью перемещаются в слой GaAs, тем самым, создавая обеднение AlGaAs. Обеднение происходит из-за того, что в гетеропереходе между материалами с разной шириной запрещенной зоны образуется квантовая яма (крутой провал) в зоне проводимости со стороны нелегированного слоя GaAs, где электроны способны быстро передвигаться без столкновений с примесями. В результате того, что электроны не в состоянии покинуть квантовую яму, образуется очень тонкая прослойка с большой концентрацией высокоподвижных электронов, поэтому сопротивление канала очень низкое (или, по другому, подвижность носителей высока). Эта прослойка называется двумерным электронным газом. Также как в других типах полевых транзисторов, приложенное к затвору HEMT напряжение изменяет проводимость этого слоя.

Моделирование СВЧ-транзистора

Рассматриваются вопросы моделирования генераторного СВЧ-транзистора, изготовленного на основе псевдоморфной гетероструктуры, состав которой приведен в таблице 1.

Моделирование производится с помощью программного пакета TCAD Sentaurus [2,3]. Для получения структуры использованы три модуля TCAD Sentaurus: DIOS, Ligament Flow Editor и Ligament Layout [2 - 4]. С помощью программ Ligament генерируется командный файл для DIOS.

В Ligament Flow Editor был смоделирован технологический процесс изготовления транзистора. При этом были использованы следующие операции:

- deposit – осаждение;
- etch – травление;
- pattern – наложение маски;
- anneal – отжиг;
- implant – легирование.

В процессе моделирования технологического процесса материалы, приведенные в таблице, заменялись материалами из имеющихся в библиотеке Ligament Flow. В дальнейшем этим материалы будут заменены на нужные с помощью программы Sentaurus Structure Editor.

Таблица 1. Конструкция гетероструктуры

Слой транзисторной гетероструктуры, назначение	Состав (x_{AlAs}, y_{InAs})
Полуизолирующая GaAs подложка	
1-ый буферный слой GaAs	
2-ой буферный слой GaAs (низкотемпературный)	
Сверхрешетка $Al_xGa_{1-x}As$/GaAs	0.22/0.0
3-ий буферный слой AlGaAs	
Сильно легированный n-AlxGa1-xAs	0.22
Спейсер AlxGa1-xAs	0.22
Сглаживающий слой GaAs	
Канал InyGa1-yAs	0.16-0.18
Сглаживающий слой GaAs	
Спейсер AlxGa1-xAs	0.22
Сильно легированный n-AlxGa1-xAs	0.22
Барьерный слой n-AlxGa1-xAs	0.22
Барьерный слой n-GaAs	
Градиентный слой n-GaAs	
Контактный слой n-GaAs	

На рис. 2 приведен получившийся технологический процесс.

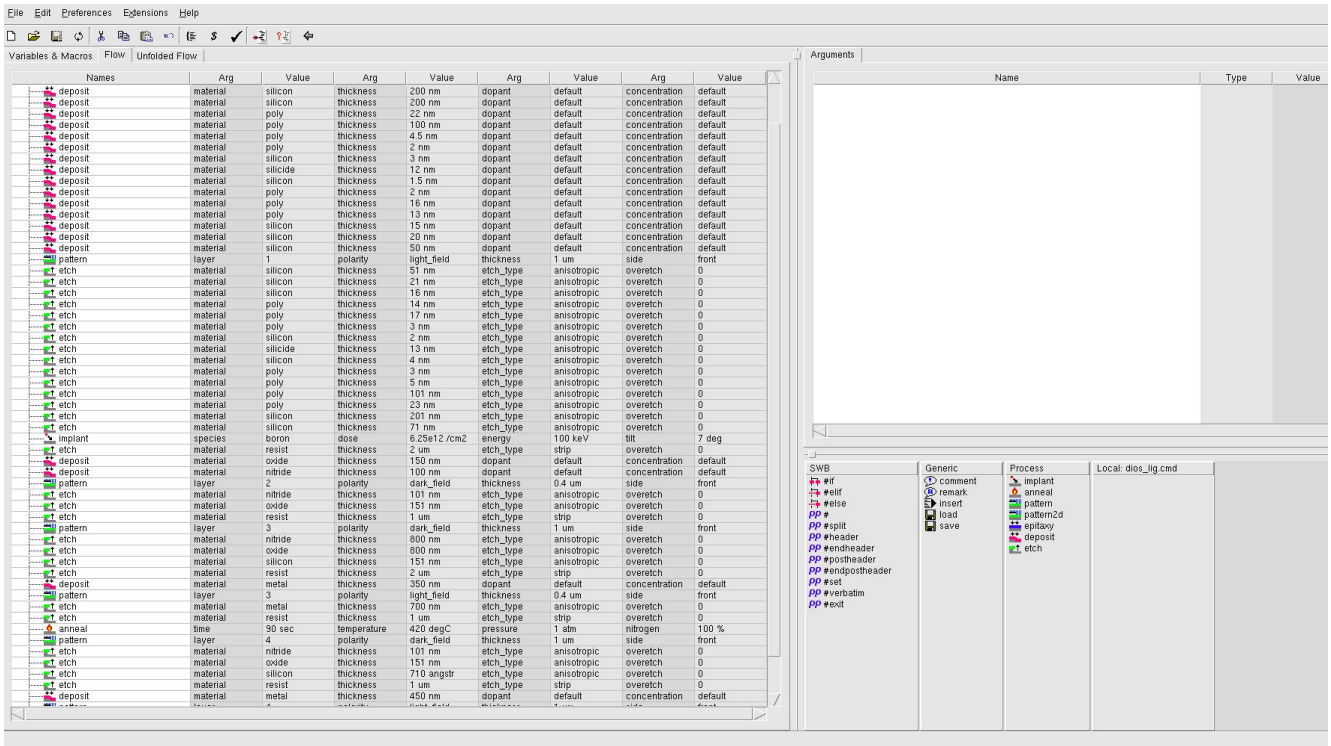


Рисунок 2 - Технологический процесс в окне программы Ligament Flow

В программе Ligament Layout был подготовлен комплект фотошабонов. На рис.3 – рис.6 фотошаблоны приведены по порядку их использования.

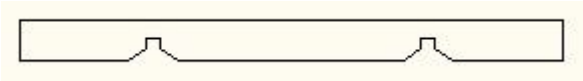


Рисунок 3 - Фотошаблон №1 (мезаструктура)

Рисунок 4 - Фотошаблон №2 (затвор и контактные площадки)

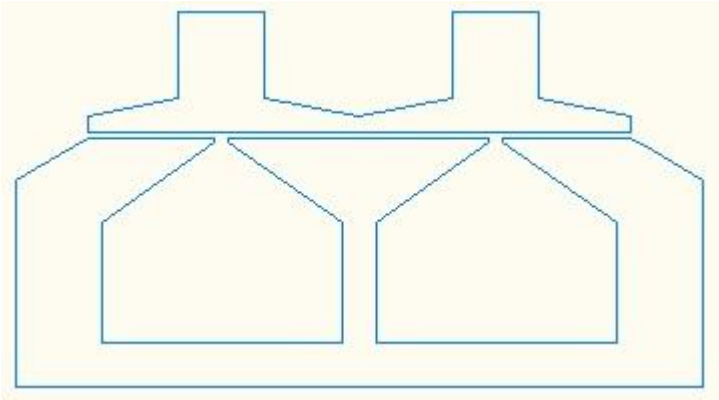


Рисунок 5 - Фотошаблон №3 (сток, исток)

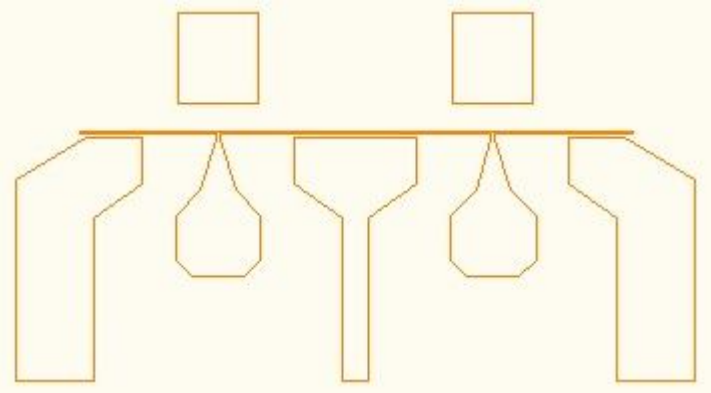


Рисунок 6 - Фотошаблон №4 (затвор и контактные площадки)

Полученная в программе Ligament Layout топология транзистора показана на рис. 7. Толстая черная линия на рис.7 показывает сечение, в котором будет производиться расчет.

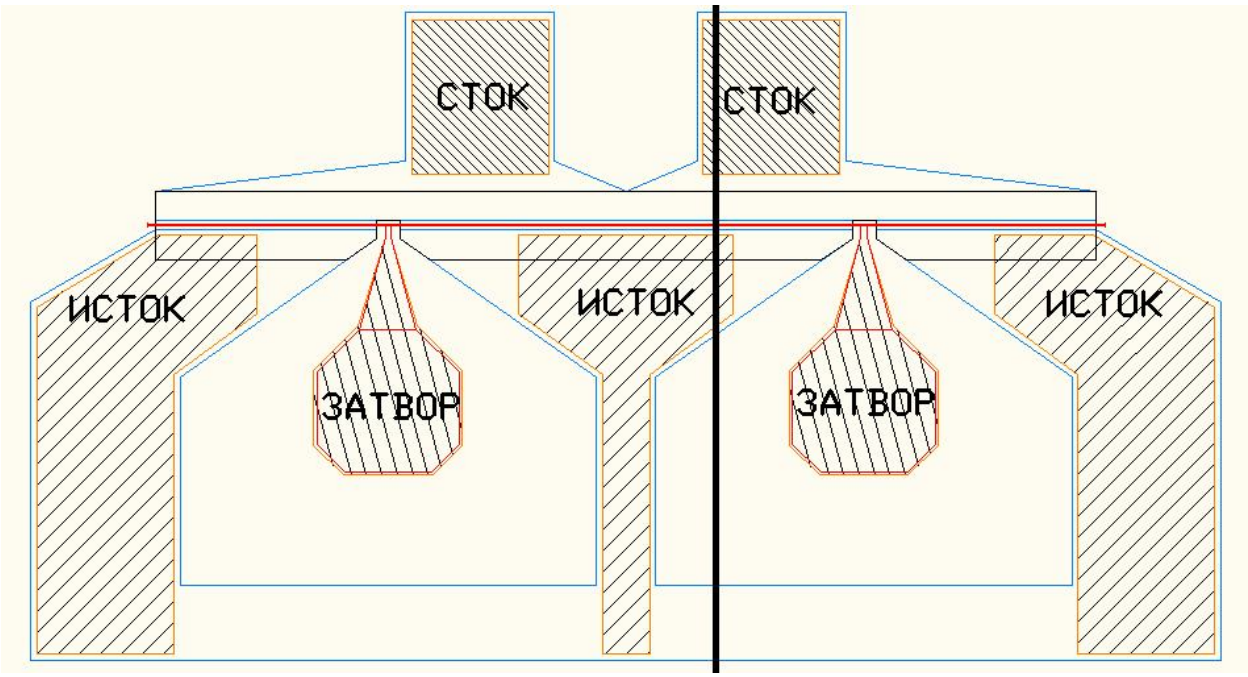


Рисунок 7 - Топология транзистора

Далее с помощью программы DIOS строилась структура на основе технологического процесса и комплекта фотошаблонов.

Заключение

В ходе работы исследованы особенности работы модулей программного комплекса TCAD Sentaurus. Смоделирован технологический маршрут изготовления транзистора в программном комплексе TCAD Sentaurus. Подготовлен комплект фотошаблонов для изготовления транзистора в программном комплексе TCAD Sentaurus. Получена структура транзистора в программном комплексе TCAD Sentaurus.

В дальнейшем планируется заменить материалы мезаструктуры в программе Sentaurus Structure Editor, отредактировать сетку в оболочке Mesh, написать командный файл в программе Sentaurus Device, получить электрические характеристики.

Литература

1. Я. А. Федотов, А. А. Щука. Транзисторы СВЧ-диапазона. - URL: http://w-rabbit.narod.ru/elektr/elem_s.htm.
2. С.В. Калинин, А.С. Черкаев, В.Е. Зырянов, Е.А. Макаров «Моделирование нанотранзисторов в системе TCAD Sentaurus».- Новосибирск: Изд-во НГТУ, 2010.
3. В.В. Асессоров, Г.В. Быкадорова, А.Ю. Ткачев. Моделирование полевых полупроводниковых приборов в САПР ISE TCAD: Учебное пособие. - Воронеж: ИПЦ ВГУ, 2007. - 27 с.
4. С.Александров, А.Алексеев. Субмикронная контактная металлизация. Выбор технологии // Наноиндустрия. – 2009, №6.
5. Zee S.M. Physics of semiconductors devices / S.M. Zee, Kwok K.Ng.-3-rd edition. - Canada.: A John Willey and sons, inc., 2007.
6. А.А. Щука Электроника: Учебное пособие. – Петербург: Изд-во «БХВ-Петербург», 2005. – 800 с.

ОЦЕНКА ЭФФЕКТИВНОСТИ МЕТОДА BDF ПРИ РЕШЕНИИ ЖЕСТКИХ СИСТЕМ ОБЫКНОВЕННЫХ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ

Сахаров М.К.
Научный руководитель к.т.н. доцент Маничев В.Б.
МГТУ им. Н.Э.Баумана, Москва, Россия

ESTIMATION OF EFFICIENCY OF BDF METHOD FOR SOLVING STIFF SYSTEMS OF ORDEINARY DIFFERENTIAL EQUATIONS

Sakharov M.K.
Supervisor: Ph.D. associate professor Manichev V.B.
BMSTU, Moscow, Russia

Аннотация
В статье рассмотрена проблема получения в CAE-подсистемах САПР/PLM гарантированного качественно верного решения систем ОДУ. Приведен набор тестовых задач состоящий из жестких систем ОДУ с известным асимптотическим или аналитическим решением. Кратко рассмотрен метод численного интегрирования BDF. Произведено тестирование реализаций численного метода BDF в программном пакете Wolfram Mathematica и в библиотеке NAG. Тестирование сводилось к оценке достоверности решения. Приведены результаты тестирования.

Abstract
This article deals with the task of obtaining the correct solution of systems of ODE in CAE-subsystems of CAD/CAE/PDM/PLM. The set of benchmark stiff systems of ODE was presented. Each system has known asymptotical or analytical solution. Method of numerical integration BDF was briefly considered. Wolfram Mathematica's and NAG C numerical library's implementations of BDF method were tested on these stiff systems. The main criteria was an assurance of a graph of solution. The results were included in this article.

Введение

Данная работа является продолжением цикла работ посвященных проблемам достоверности и точности решения систем ОДУ в современных CAE подсистемах САПР [1,3,4]. Математическое моделирование процессов и их анализ являются неотъемлемой частью цикла проектирования не только в наукоемких отраслях, но и при производстве продукции массового спроса. В математическом ядре любой CAE подсистемы реализованы различные методы численного интегрирования. Но большинство инженеров проектировщиков не являются специалистами в области численных методов и зачастую не могут оценить адекватность результатов полученных подсистемой. Поэтому, при компьютерном моделировании и анализе разнообразных технических систем и объектов на основе дифференциальных уравнений, предъявляются жесткие требования к эффективности применяемых численных методов.

Постановка задачи

Эффективность численных методов определяется точностью получаемых решений и затратами машинного времени. Рост производительности вычислительной техники, а также развитие «облачных» технологий и технологий удаленного доступа, позволяющих пользователю скромного по вычислительным возможностям персонального компьютера, использовать мощные вычислительные ресурсы современных суперкомпьютеров, выдвигает на первый план именно точность получаемых решений. Не имеет смысла рассматривать эффективность численного метода без учета достоверности выдаваемого им приближенного решения. В CAE-подсистемах должна быть обеспечена полная гарантия достоверности и точности результатов, выдаваемых инженерам-проектировщикам. Конечно, в большинстве случаев, при моделировании реальных технических систем не требуется высокая математическая точность расчетов, но достоверность результатов должна быть соблюдена.

Количественно или качественно неверное решение, выданное системой без предупреждения, может привести к неверным расчетам, например, траекторий движения летательных аппаратов и, как следствие, крупным авариям.

Известно, что решение жестких и супержестких систем ОДУ традиционно вызывает затруднения у большинства численных методов. При использовании явных методов, например явного метода Рунге-Кутта, для получения корректного решения необходимо использование очень малого шага, что на порядки увеличивает время вычисления. Адаптивные методы, регулирующие значения шага в процессе решения, также не гарантируют качественно верного решения. Для решения такого рода систем традиционно используются неявные методы, в частности метод BDF (Backward Differntiation Formula).

Метод BDF

Фактически метод BDF это семейство неявных многошаговых методов численного интегрирования, применяемых для решения задач с известными начальными условиями [2]. На практике методы BDF используются в основном для решения жестких систем ОДУ. Рассмотрим такую задачу:

$$y' = f(t, y), \qquad y(t_0) = y_0. \tag{1}$$

Общая формула для методов BDF может быть записана следующим образом:

$$\sum_{i=0}^{k} a_{k-i} \cdot y_{n+1-i} = h \cdot f(t_{n+1}, y_{n+1}) \tag{2}$$

где h - это шаг интегрирования, a_{k-i} – коэффициент, k определяет порядок полинома аппроксимации и соответственно количество интерполируемых точек до t_{n+1}. К примеру:

$$k = 1: \quad y_{n+1} = y_n + h \cdot f(t_{n+1}, y_{n+1}) \tag{3}$$

$$k = 2: \quad y_{n+1} = \frac{4}{3} y_n - \frac{1}{3} y_{n-1} + h \cdot \frac{2}{3} f(t_{n+1}, y_{n+1}) \tag{4}$$

$$k = 3: \quad y_{n+1} = \frac{18}{11} y_n - \frac{9}{11} y_{n-1} + \frac{2}{11} y_{n-2} + h \cdot \frac{6}{11} f(t_{n+1}, y_{n+1}) \tag{5}$$

При $k=1$ получаем неявный метод Эйлера, при $k=2$ - метод BDF-2 и т.д. Отметим что с ростом k область устойчивости метода уменьшается. Так как метод неявный то на каждом шаге интегрирования необходимо решать систему нелинейных уравнений. Обычно для этих целей используется модифицированный метод Ньютона.

Набор тестовых задач

В данной работе набор тестовых задач состоит из 4 систем ОДУ [1]. Эти системы имеют известное асимптотическое или аналитическое решение. Первый тест – классическая задача Ван дер Поля – жесткая система второго порядка. При тестировании использовались два значения параметра жесткости $MU=10^6$ и $MU=10^9$. Второй тест – пример моделирования сверхколебательной электронной схемы – нежесткая система пятого порядка. Тестирование проводилось для часто встречающихся на практике параметров масштабных коэффициентов $kt=1$, $ki=1$, $ku=10^{-2}$, а также для значений параметров $kt=10^{-104}$, $ki=1$, $ku=1$, которые создают трудный тест для современных решателей ОДУ. Третий тест – нелинейная жесткая система ОДУ второго порядка имеющая локально неустойчивое решение и составленная по примеру теста Ван дер Поля – тест Скворцова. Тестирование было проведено при параметре жесткости $MU=10^6$. И четвертый тест – нелинейная жесткая система ОДУ второго порядка для моделирования процессов реального лазера – тест Евстифеева. Тестирование сводилось к оценке качественно верного вида решения.

Используемые реализации метода BDF

Метод BDF реализован практически во всех математических средах и библиотеках. Для тестирования был выбран один из лучших на сегодняшний день математических пакетов – Wolfram Mathematica. Данный пакет прикладных программ содержит в себе огромный массив различных алгоритмов. С уверенностью можно сказать, что на сегодняшний день он является лидером в сфере научных и технических вычислений. Система Mathematica, выпускаемая американской компанией Wolfram Research, Inc., имеет высокую скорость и практически не ограниченную точность вычислений.

Однако, нельзя забывать что в сфере инженерных вычислений помимо программных пакетов активно используются подключаемые библиотеки написанные на различных языках программирования и содержащие в себе большое количество разнообразных численных методов. В нашем случае для тестирования была выбрана реализация метода BDF в библиотеке NAG написанной на языке C и разработанной одноименной британской компанией (Numerical Algorithms Group, www.nag.co.uk). В настоящее время библиотеки NAG совместимы практически со всем современными языками программирования высокого уровня, а также доступны для использования в пакетах LabVIEW и MATLAB.

Результаты тестирования

Тестирование проводилось при заданной относительной точности интегрирования 0.001, остальные параметры решателей задавались по умолчанию. Результаты тестирования представлены в сводной таблице 1.

Таблица 1 – Сводная таблица результатов тестирования

Тесты/Решатели ОДУ	Mathematica 8 Метод **BDF**	NAG C Library Функция **nag_ode_ivp_bdf_gen**	
		TOL=10^{-3}	TOL=10^{-9}
ТЕСТ 1а. Уравнения Ван дер Поля $MU=10^6$	-	+	+
ТЕСТ 1б. Уравнения Ван дер Поля $MU=10^9$	-	-	+
ТЕСТ 2а. Высокодобротный фильтр $kt=1, ki=1, ku=0.01$	+	-	+
ТЕСТ 2б. Высокодобротный фильтр $kt=10^{-104}, ku=1, ki=1$	+	-	-
ТЕСТ 3. Локально-неустойчивая система ОДУ $MU=10^6$	+	-	+
ТЕСТ 4. Моделирование свечения лазера	+	+	+

Из таблицы видно, что реализация метода BDF в библиотеке NAG при точности по умолчанию справилась лишь с самыми простыми тестами. Другой вариант программы решателя позволил повысить точность до девяти знаков после запятой. В результате, почти все тесты были решены верно. Необходимо отметить, что из пяти выданных неверных результатов при использовании библиотеки NAG, лишь один из них сопровождался диагностическим сообщением.

При тестировании реализации метода BDF в Wolfram Mathematica были получены более качественные результаты – решатель не справился лишь с тестами Ван дер Поля, выдав в обоих случаях диагностическое сообщение. Также необходимо отметить, что проблемными для разных решателей стали совершенно разные тесты, что свидетельствует о принципиальных различиях в реализациях самого метода BDF.

На рисунках 1 – 4 представлены некоторые результаты тестирования.

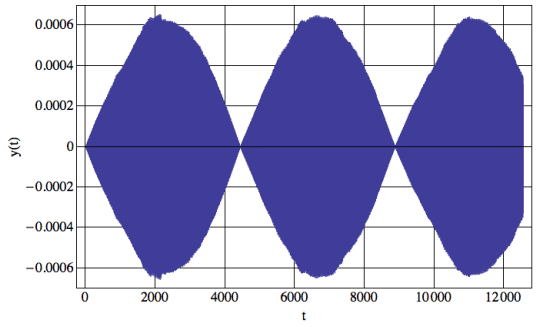


Рисунок 1 – Решение ТЕСТ 2а полученное с помощью библиотеки NAG при TOL = 10^{-3}

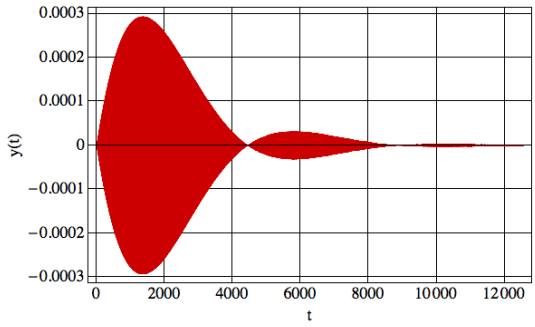


Рисунок 2 – Правильное решение ТЕСТ 2а

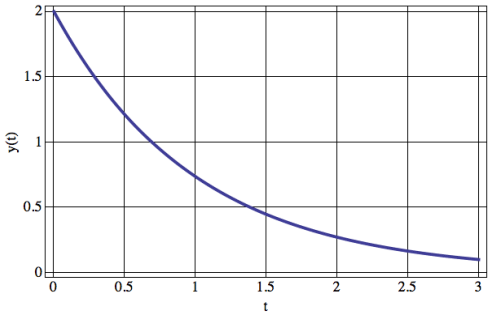


Рисунок 3 - Решение ТЕСТ 3 полученное с помощью библиотеки NAG при TOL = 10^{-3}

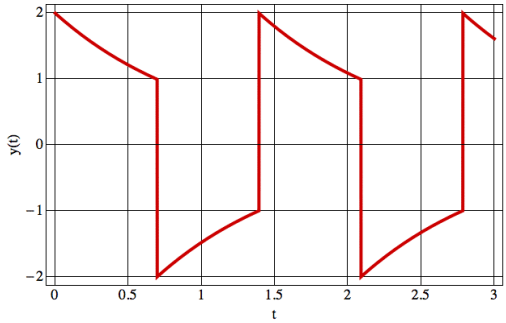


Рисунок 4 - Правильное решение ТЕСТ 3

Заключение

В работе было проведено тестирование двух реализаций метода BDF - реализаций компаний NAG и Wolfram Research, Inc. Результаты были проанализированы с точки зрения достоверности выдаваемого решения и наличия диагностических сообщений. Основным выводом из данного исследования является подтверждение того факта, что даже лучшие САЕ подсистемы на сегодняшний день не способны гарантировать качественно верное решение различных систем ОДУ. Также, проблемой является отсутствие диагностических сообщений в случае некорректного решения. Большинство инженеров не обладают достаточными знаниями в области численных методов, чтобы определить достоверность результата без диагностического сообщения. Данная проблема была решена сотрудниками и студентами кафедры САПР при разработке математического ядра SADEL для платформы математического моделирования во временной области разнородных технических систем и объектов ПА10 (SADEL-PA10) [1].

Литература

1. Д.М. Жук, В.Б. Маничев, А.В. Андронов Платформа математического моделирования во временной области разнородных технических систем и объектов FMS PA10. В сб. научных трудов Всероссийской научно-технической конференции МЭС - 2010 - М.: ИППМ РАН, 2010.
2. Хайрер Э., Ваннер Г. Решение обыкновенных дифференциальных уравнений. Жесткие и дифференциально-алгебраические задачи. – М.: Мир, 1999.
3. Д.М. Жук, В.Б. Маничев, А.О. Ильницкий Методы и алгоритмы решения дифференциально-алгебраических уравнений для моделирования систем и объектов во временной области. // Информационные технологии. - 2010. – часть1 - №7, часть 2 - №8.
4. Сахаров М. К. Корректность и точность решений жестких систем ОДУ в системе MathCAD 14 // Наукоемкие технологии и интеллектуальные системы 2011. – М.: МГТУ им. Н.Э. Баумана, 2011. – С. 16-19

СКОРОСТНОЙ ИНТЕРПОЛЯЦИОННЫЙ АГОРИТМ ВЫЧИСЛЕНИЙ МЕТЕО-ДОБАВКИ СКОРОСТИ ВЕТРА ДЛЯ УЧАСТКА ПОЛЕТА ВС

Новиков П.В., Арутюнян Д.В.
Научный руководитель: к.т.н., доцент Власов А.И.
МГТУ им. Н.Э.Баумана, кафедра ИУ4, Москва, Россия

HIGH-SPEED INTERPOLATION COMPUTING AGORITM METEO-ADDITIVES FOR WIND SPEED SECTION OF THE FLIGHT SUN

Novikov P.V., Arutjunjan D.V.
Supervisor: Ph.D., associate professor, Vlasov A.I.
MSTU, Moscow, Russia

Аннотация
В данной статье описывается высокоскоростной интерполяционный алгоритм вычисления метео-составляющей скорости ветра для расчета общей скорости полета воздушного средства на сегменте полета. Данный алгоритм можно применять при четырёх-мерном планировании и организации полета воздушного средства на прямолинейном сегменте его курса, для дальнейшего более точного расчета времени прибытия воздушного средства в пункт назначения, а так же для более точной оценки количества истраченного топлива для выполнения полета по маршруту.

Abstract
This article describes a high-speed interpolation algorithm for computing the meteorological component of the wind speed to calculate the total rate of air flight segment means for the flight. This algorithm can be applied in four-dimensional planning and organizing the flight of air tools on the straight segment of its course, for further, more precise timing of arrival of air at the point of destination, as well as to more accurately estimate the amount of consumed fuel for the flight on the route.

Введение

Входными данными для расчета составляющей скорости ветра служат значения скорости и направления ветра, передаваемые различными центрами (Вашингтон, Брекнелл, Москва) в ГСТ в кодах ГРИД или ГРИБ, на уровнях от 1000 до 100 гПа [1].

Для описания значений ветра в метео-системе используется метео-сетка. Данная сетка имеет строгое разрешения 1,25ºx1,25º, составляющие значения ветра для каждого метео-сектора известны. Таким образом, данная метео-сетка точно определена и все составляющие ветра в ней описаны двумя составляющими:

- Зональная составляющая ветра (восточная), м/с – U
- Меридиональная составляющая ветра (северная) , м/с – V.

Данные, полученные от центров поставщиков в закодированном формате, декодирубтся и загружаются в БД. Структура БД для хранения метео-данных по прогнозам и по климату, а так же блок ведения собственного климата представлены на рисунке 1.

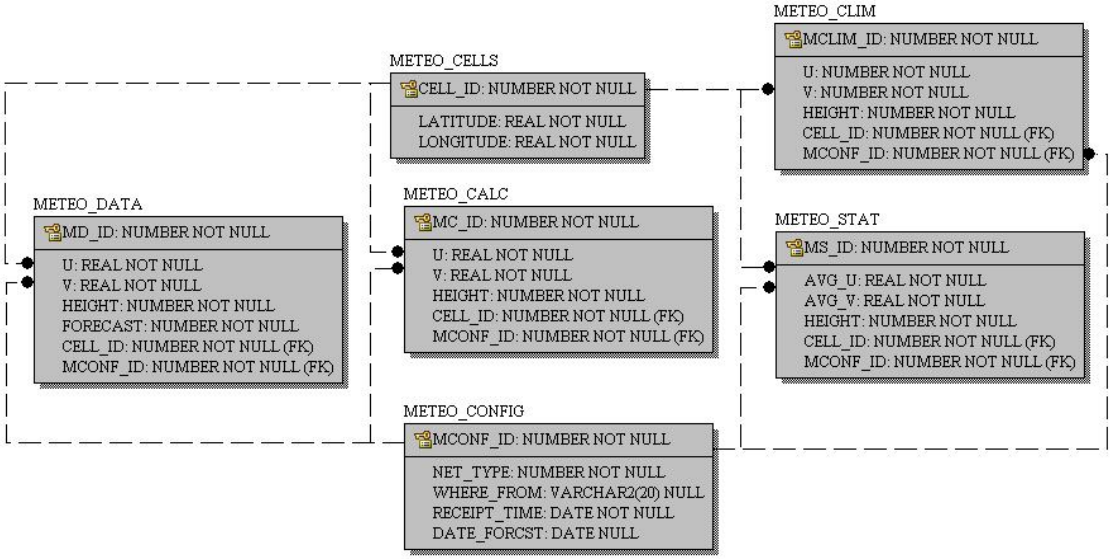


Рисунок 1 – Структура БД хранения метео-информации

Таблица 1 – Спецификация таблиц структуры БД

Таблица	Описание
METEO_DATA	Хранит текущие данные по прогнозам
METEO_CONFIG	Хранит описание одного прогноза по климату или по прогнозам
METEO_CALC	Необходимая для хранения суточной статистики
METEO_STAT	Усредненная статистика за каждые сутки
METEO_CELLS	Информационная таблица описания метео-ячеек
METEO_CLIM	Хранит данные по климату

Математическая модель реализации скоростного интерполяционного алгоритма расчета

Входными данными для расчета служат значения скорости и направления ветра, передаваемые различными центрами (Вашингтон, Брекнелл, Москва) в ГСТ в кодах ГРИД или ГРИБ, на уровнях от 1000 до 100 гПа, (в крайнем случае, это могут быть и аэрологические данные, но их слишком мало).

Маршрут полета, описанный в плане полета (РПЛ, ППЛ, ФПЛ), и его изменения представляет собой заранее рассчитанную четырехмерную пространственную траекторию полета ВС в виде связанных отрезков линии пути, каждый из которых задан координатами, высотой (эшелоном) и временем прохода пунктов маршрута, в том числе и пунктов изменения профиля полета. На каждом участке скорость ВС раскладывается на составляющие u_c и v_c (с учетом расположения участка на карте). Далее определяется средний ветер на этом участке, с учетом выбранного эшелона полета, причем учитывается ветер в начале, середине и в конце участка (u_1, v_1, u_2, v_2, u_3, v_3). Для нахождения скорости ветра на эшелоне (уровне полета p), не совпадающем с уровнями отбора данных, применяется логарифмическая интерполяция [2]:

$$u = \left(u_1(p+p_1)(p_2-p) + u_2(p+p_2)(p-p_1)\right)/2p(p_2-p_1); \quad (1)$$

$$v = \left(v_1(p+p_1)(p_2-p) + v_2(p+p_2)(p-p_1)\right)/2p(p_2-p_1), \quad (2)$$

где u_1, p_1, v_1 и u_2, p_2, v_2 - составляющие ветра и давление на уровнях выше и ниже интерполируемого.

Для нахождения скорости ветра на конкретном уровне, но в произвольной точке применяется билинейная интерполяция [2]:

$$u = u_a + (u_d - u_a)Dx + (u_b - u_a)Dy + (u_a - u_b + u_c - u_d)DxDy\,, \quad (3)$$

$$v = v_a + (v_d - v_a)Dx + (v_b - v_a)Dy + (v_a - v_b + v_c - v_d)DxDy\,, \qquad (4)$$

где u_i, v_i - значения составляющих скорости ветра в узлах регулярной сетки, Dx, Dy - шаги сетки.

Средний ветер на участке трассы:

$$v_{1отр} = (v_1 + v_2 + v_3)/3\,; \qquad (5)$$

$$u_{1отр} = (u_1 + u_2 + u_3)/3\,. \qquad (6)$$

Путевая (суммарная) скорость ВС с учетом ветра:

$$v_{св} = v_c + v_{1отр}\,; \qquad (7)$$

$$u_{св} = u_c + u_{1отр}\,, \qquad (8)$$

где $v_{св}$, $u_{св}$ - составляющие скорости ВС с учетом ветра. Путевая (итоговая) скорость ВС (W) определяется как квадратный корень из суммы квадратов $v_{св}$ и $u_{св}$. Зная длину участка s (в км), определяется время в пути:

$$W^2 = v_{св}^2 + u_{св}^2\,, \qquad (9)$$

$$W = \sqrt{v_{св}^2 + u_{св}^2}\,. \qquad (10)$$

Таким образом, на каждом заданном участке трассы определяются истинные скорости ВС.

Алгоритмическая модель реализации функционирования средств расчета метео добавки

На рисунке 2 предствлен общий алгоритм функционирования реализыемой системы вычислений метео.

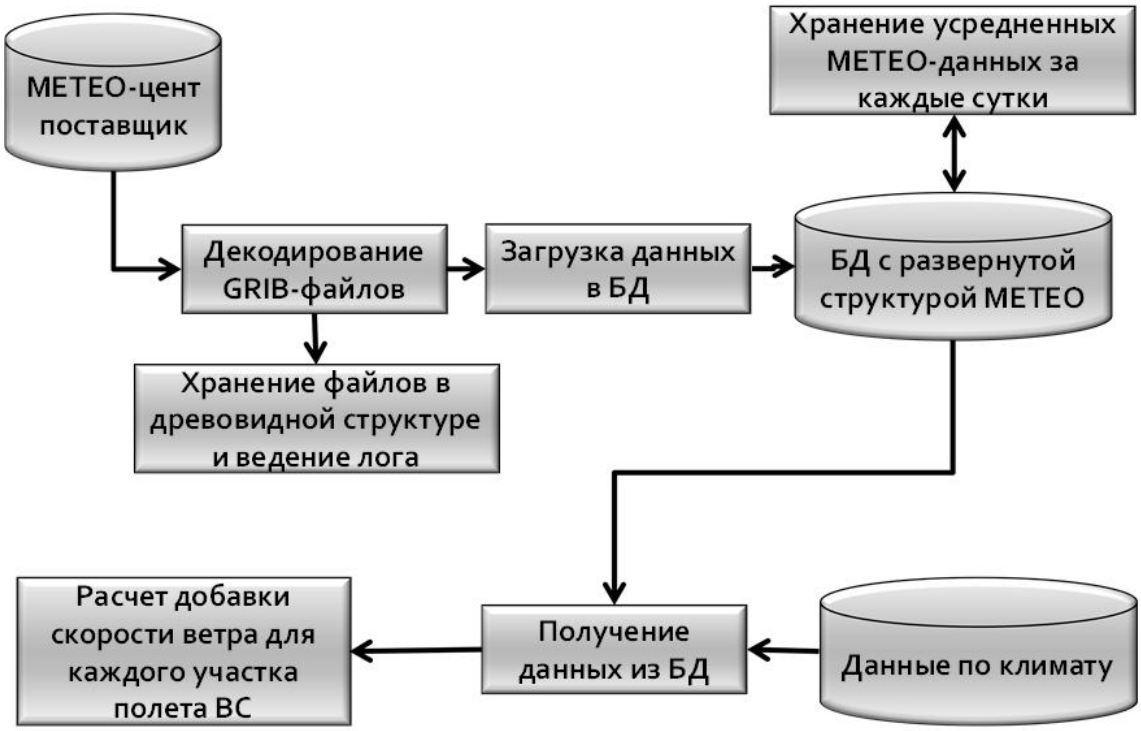


Рисунок 2 – Общий алгоритм работы МЕТЕО-блока

В начале метеоцентр поставщик подает на вход системы закодированные файлы путем загрузки в конкретную дирректорию. После чего система декодирует полученные файлы и загружает их в БД, а так же перемещает обработанные файлы в структуру папок по текущему времени и ведет лог. СУБД получая данные, заносит их в таблицы, и отработанные данные усредняет и заносит в статистику. Затем запускается расчет планов, который запрашивает сохраненную метео информацию из БД и проводит расчет добавки скорости ветра для каждого сегмента полета.

На следующем рисунке представлен алгоритм непосредственного вычисления скорости ветра для каждого сегмента полета, с использованием описаного выше скоростного интерполяционного алгоритма.

Вначале на вход программе подаются данные для расчета, а именно: широта и долгота начально и конечно точки сегмента, эшелон полета по сегменту, время пролета через первую точку. Затем полученная информация проверяется на ее корректность: входная высота более 0, начальная и конечная точка принадлежат зоне ВП РФ. Затем определяется перекрытие интервала прогноза и даты полета, на основе чего принимается решение о получении данных из прогноза или из климата.

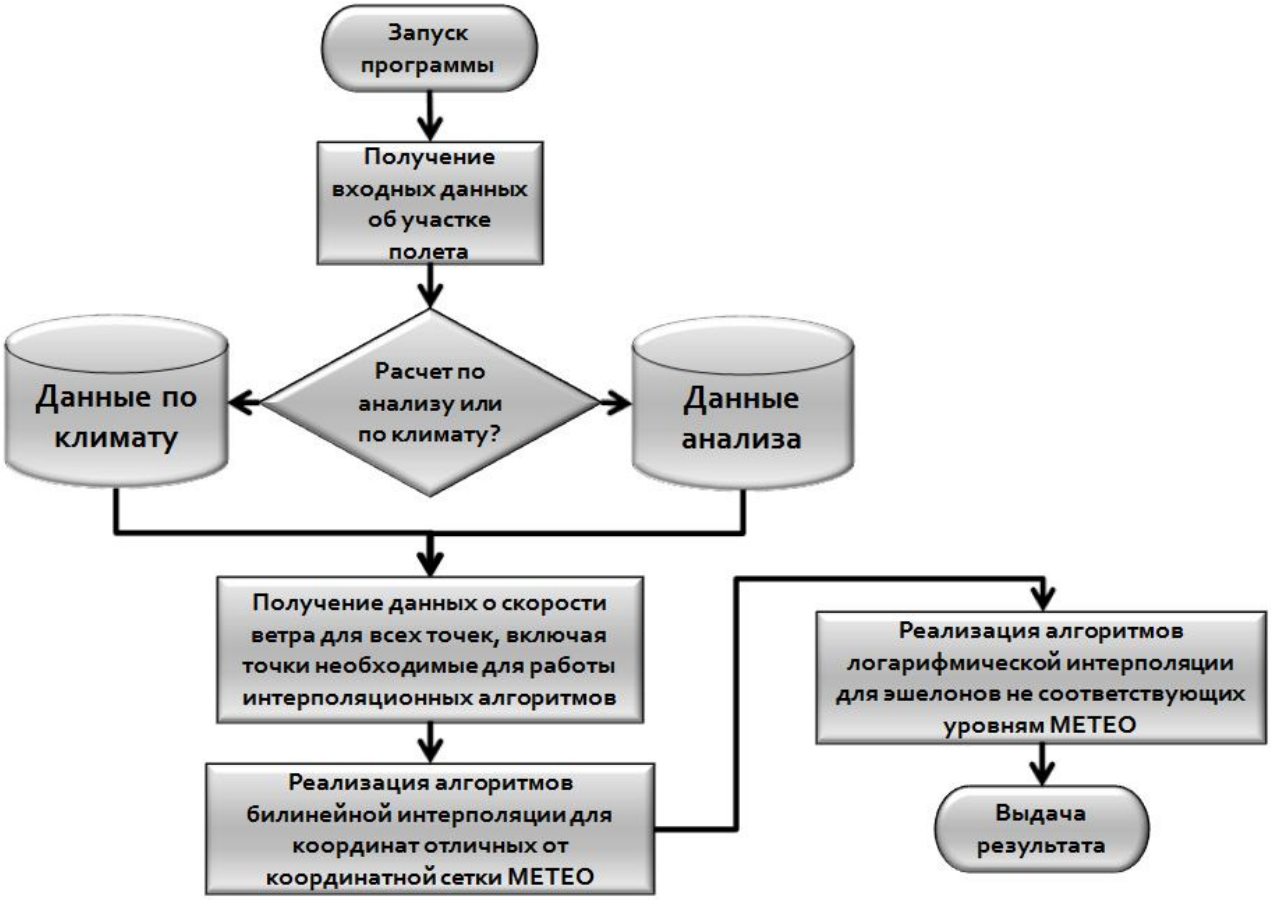


Рисунок 3 – Алгоритм расчета добавки ветра

На рисунке 3 представлен разработанный алгоритм. В его основе лежат 2 интерполяционных алгоритма: алгоритм билинейной интерполяции координат и алгоритм логарифмической интерполяции высот. Так как скорость ветра в пространстве не может меняться скачкообразно, то приминение данных алгоритмов не ведет к резкому уменьшению тоности.

Заключение

В данной работе описывается алгоритм нахождения метео-составляющей скорости ветра для расчета четырехмерной траектории полета ВС. Основным отличием данного алгоритма от алгоритма изложенного в [3] является то, что он запрашивает меньше данных из БД и не требует сложных вычислений нахождения точек пространственного пересечения траектории полета и метео сетки, что позволяет повысить быстродействие рассчитывающей системы.

Литература

1. Всемирная метеорологическая организация Наставление по кодам Том I : технический регламент ВМО. ВМО №306 / Издание 2001г., 2001 .— 509 с.
2. Гончаров В.Л., Теория интерполирования и приближения функций. — 2-е изд., перераб. — М.: Сделано в холле, 1954. – 325 с.
3. Новиков П.В., Алгоритм высокоточных вычислений метео-добавки скорости ветра для участка полета воздушного средства, – Электронный ресурс. Режим доступа: http://elibrary.ru/item.asp?id=17105440- Проверено 12.02.2012.

ВЕРИФИКАЦИЯ КОММУНИКАЦИОННОЙ СЕТИ ТОПОЛОГИИ «4-D TOP»

Иванов А. М.
Научный руководитель: доцент, к.т.н., Власов А.И.
МГТУ им. Баумана, Москва, РФ

VERIFICATION OF COMMUNICATION NET WITH “4D TORUS” TOPOLOGY

Ivanov A.M.
Research advisor: docent, c.e.s., Vlasov A.I.
Bauman University, Moscow, RF

Аннотация
Рассмотрены принципы построения тестового окружения при проектировании высокоскоростной коммуникационной сети «4D Tor». Упомянуты аналоги методов верификации различных распределенных систем. Описано тестовое окружение для блока маршрутизатора, отвечающего за прием и передачу информационных пакетов.
Abstract
Principles of construction of a test environment at designing of a high-speed communication network «4D Torahs» are considered. Analogs of verification methods of the various distributed systems are mentioned. The test environment for the block of the router which is responsible for reception and transfer of information packages is described.

Введение

На сегодняшний день в области создания распределенных систем передачи и обработки данных сложилась диспропорция темпов развития трёх её основных компонентов: микропроцессорной техники, средств телекоммуникации и разработки программного обеспечения [1]. Это приводит к тому, что разрыв между техническими возможностями и технологиями обработки и передачи информации всё время увеличивается. В значительной степени, это происходит из-за отсутствия удовлетворительного для промышленности решения проблемы проверки правильности программных систем и логических схем вычислительных устройств. Например, из практики разработки программного обеспечения хорошо известно, что примерно 2/3 времени, затрачиваемого на создание системы, приходится на отладку, т.е. проверку её правильности. Как правило, большинством разработчиков программных систем для проверки правильности проекта практикуются **методы имитационного моделирования и тестирования** [1]. Они довольно эффективны на самых ранних стадиях отладки, когда проектируемая система всё ещё изобилует ошибками, но результативность этих методов быстро снижается, как только система становится чище. Достойной альтернативой имитационному моделированию и тестированию являются методы **формальной верификации**. При имитационном моделировании и тестировании исследуются только некоторые из возможных сценариев поведения проектируемой системы, поэтому остаётся открытым вопрос о том, не содержится ли фатальная ошибка в незадействованных траекториях. Формальная верификация же обеспечивает исчерпывающий анализ всех возможных вариантов поведения системы. Техника верификации, получившая название **проверки на модели**, является одним из наиболее перспективных и широко используемых подходов к решению проблемы автоматизации отладки и проверки правильности программ. Данный метод применялся и при создании тестового окружения для коммуникационной сети топологии «4D Tor» в компании «НИЦЭВТ». По сравнению с другими подходами в формальной верификации про грамм, метод проверки на модели обладает двумя замечательными преимуществами [1].

Он полностью автоматический, и его применение не требует от пользователя никаких особых знаний в таких математических дисциплинах, как логика и теория доказательства

теорем. Всякий, кто может про вести моделирование проектируемой системы, вполне способен осуществить и проверку этой системы.

Если проектируемая система не обладает желаемым свойством, то результатом проверки на модели будет контрпример, который демонстрирует поведение системы, опровергающее это свойство. Эта ошибочная трасса даёт бесценную информацию для понимания причины ошибки, и важный ключ к решению возникшей проблемы.

1 Принципы построения тестового окружения

Кроме внешнего окружения, в рамках которого должна работать среда верификации, необходимо также определить ее **каркас** — некоторый базовый набор компонентов, реализующих основной набор функций и поддерживающие основные потоки данных внутри системы. К этому каркасу будут добавляться другие компоненты, поддерживающие вспомогательные и менее значимые функции [2]. Предлагается использовать в качестве основы для построения среды верификации архитектурный каркас ***RTL*-модели** маршрутизатора. Это решение вызвано тем обстоятельством, что на построение собственной модели для тестирования тратится неоправданно много времени. Вместо полноценной тестовой модели создается надстройка над *RTL*-модулем в виде интерфейса с высокоуровневым тестовым окружением.

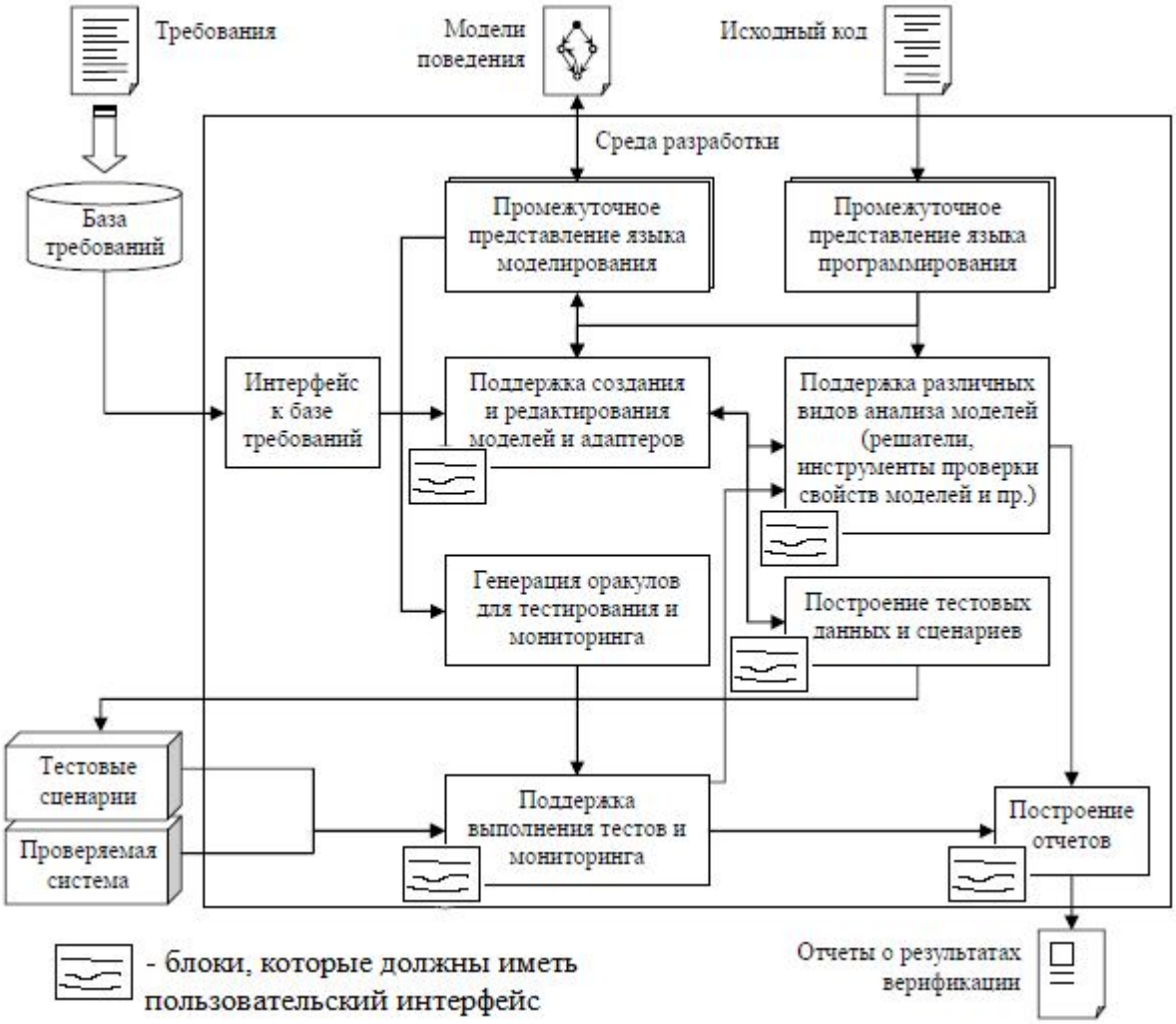


Рисунок 1 – архитектура системы верификации программного обеспечения

Обычно в ходе тестирования на основе моделей необходимо сделать следующее [2]:

- определить модель поведения тестируемой системы, формализующую требования к этому поведению.
- проанализировать структуру модели для выбора критериев покрытия и отдельных целей тестирования, и определить эти критерии и цели.
- построить среду выполнения тестов, включающую средства мониторинга для протоколирования внешних действий, реакций системы, и, возможно, внутренних ее событий, а также тестовые оракулы (рис.1) — программные компоненты, определяющие соответствие или несоответствие наблюдаемого поведения системы и модели. Обычно такая среда состоит из библиотеки поддержки выполнения тестов, набора тестовых оракулов для всех проверяемых компонентов и набора адаптеров, связывающих эти компоненты с их оракулами. Оракулы в большинстве случаев генерируются автоматически из ранее построенной модели.
- построить, автоматически или с привлечением человека, набор тестовых сценариев, определяющих последовательности вызова различных операций тестируемой системы или посылки ей сообщений или сигналов и данные, передаваемые в качестве параметров операций и сообщений.
- выполнить тестовые сценарии, протоколируя всю информацию, касающуюся соответствия наблюдаемого поведения системы и ее модели, а также покрытых во время тестирования ситуаций.
- провести анализ результатов тестов, в ходе которого выявляются и анализируются ошибки в системе или ее модели (проявляющиеся как несоответствия между ожидаемым и реальным поведением), а также анализируется достигнутое тестовое покрытие и принимается решение либо о создании дополнительных тестов, либо об окончании их разработки [3].

2 Верификация блока «Link» маршрутизатора

Блок «*Link*» маршрутизатора состоит из двух частей. Блок анализа и демультиплексор (БАД) предназначен для приема информационных пакетов из сети и направлении их по одному из виртуальных каналов в соответствии с маршрутом. Блок передачи данных (БПД) осуществляет справедливый арбитраж пакетов (пропускание пакетов в сеть из нескольких виртуальных каналов последовательно в соответствии с очередностью). Так же данные блоки должны удовлетворять требованиям быстродействия (обеспечивать минимальные задержки прохождения пакетов) и отказоустойчивости (обеспечивать гарантии сохранения целостности пакета при прохождении его через сеть). Целью построения тестового окружения является:

- проверка правильности приема пакетов;
- оценка быстродействия прохождения пакетов через *Link;*
- проверка арбитража пакетов;
- оценка отказоустойчивости блока.

Внутреннее устройство данных блоков и их интерфейсы с другими модулями маршрутизатора показаны на рисунках 2 и 3.

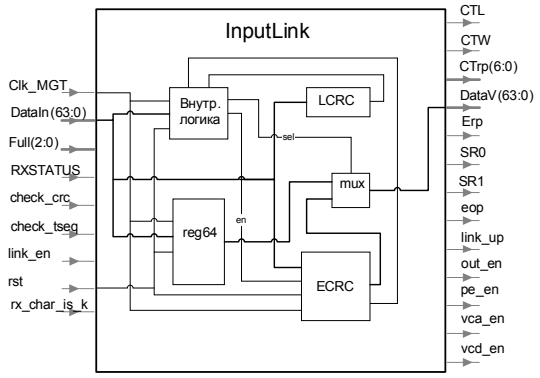


Рисунок 2 – Блок анализа и демультиплексор

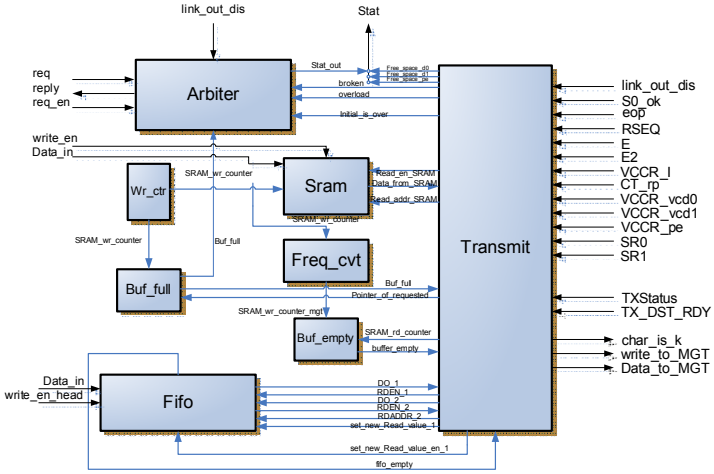


Рисунок 3 – Блок передачи данных

В проектирования создания системы верификации была создана надстройка для связи RTL-модели Линка с внешним тестовым окружением. Так же создан **драйвер** – генератор входных сигналов, который определяет воздействия на RTL, различные счетчики и события для реализации тестовых сценариев, **монитор** – интерфейс пользователя, который отображает факты прохождения тестов и статистические данные.

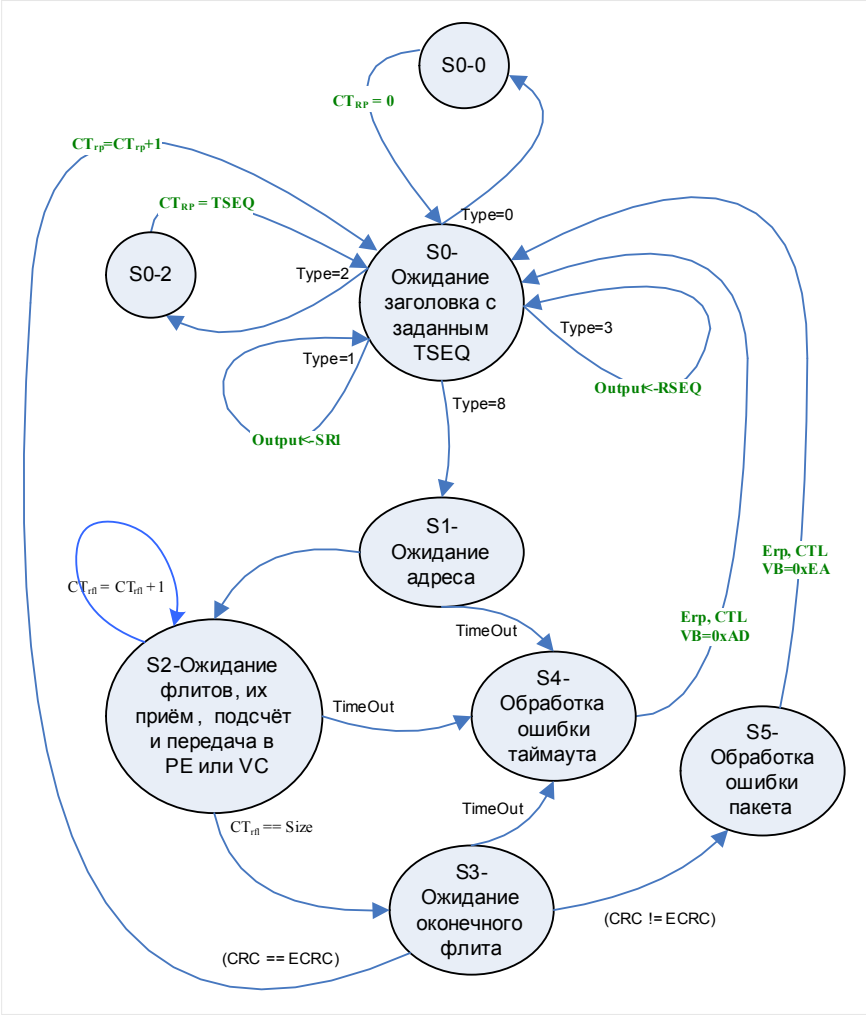


Рисунок 4 – Диаграмма состояний блока БАД

В результате тестирования проверялись следующие функции Линка.

Для блока БАД:

- приём пакетов из сети;
- анализ пакетов и передача в соответствующий виртуальный канал или входную очередь локального вычислительного узла;
- контроль целостности данных пакета;
- определение направления передачи пакетов на основании номера виртуального канала;
- контроль целостности данных путём подсчёта пакетов, контроля CRC (контрольной суммы) заголовочного флита и тела пакета;

Тестовые сценарии для блока БАД строились на основании диаграммы состояний, показанной на рисунке 4.

Для блока БПД:

- способность осуществлять справедливый арбитраж запросов и установление соединений посредством кроссбара с различными виртуальными каналами маршрутизатора;
- прием данных с выхода кроссбара и помещение их во внутренний буфер БПД;
- передача принятых во внутренний буфер БПД данных, не дожидаясь завершения приёма, на следующий маршрутизатор;
- осуществление контроля и удаления из внутреннего буфера БПД пакетов, успешно принятых следующим маршрутизатором, а также повторную передачу пакетов при возникновении ошибки на приёмной стороне).

Исследования работы Линка проводились с использованием языка моделирования и верификации SystemC, а так же с применением среды разработки тестового окружения ModelSim фирмы «Mentor Graphics» [3].

Литература

1. Гаранина Н.О., «Верификация распределенных систем с использованием аффинного представления данных, логик знаний и действий» Дис. канд. физ.-мат. Наук, Новосибирск, 2004.
2. Кулямин В.В.: «Интеграция методов верификации программных систем», ИСП РАН, Москва, 2008.
3. А.Лохов «Функциональная верификация СБИС в свете решений Mentor Graphics», выст. «Метрология и измерительные системы» 2004.

МОДЕЛИРОВАНИЕ КИНЕМАТИКИ МНОГОСЕКЦИОННОГО МАНИПУЛЯТОРА ТИПА «ХОБОТ»

Волкоморов С.В.
Научный руководитель: д.ф.-м.н., профессор, Карпенко А.П.
МГТУ им. Н.Э.Баумана, Москва, Россия

KINEMATIC MODELING FOR MULTISECTION “TUSK” MANIPULATOR

S.V. Volkomorov
Supervisor: D.Sc., Prof., A.P. Karpenko
MSTU, Moscow, Russia

Аннотация
Поставлена задача обратной кинематики для манипулятора типа «хобот». Предложены методы ее решения. Один из методов рассмотрен подробно. Представлены результаты применения описанного метода.
Abstract
Design task for multisection «tusk» manipulator inverse kinematics is posed. Solving approaches for it are considered. Solution using one of the approaches is performed.

Постановка обратной позиционной задачи. Свяжем с i-ой секцией манипулятора систему координат $0_i x_i y_i z_i$, в которой ось $0_i y_i$ совпадает с продольной осью симметрии этой секции, а оси $0_i x_i$, $0_i z_i$ расположены таким образом, что образуют с осью $0_i y_i$ правую тройку (рисунок 1). Систему координат, связанную с основанием манипулятора, обозначим $0_0 x_0 y_0 z_0$.

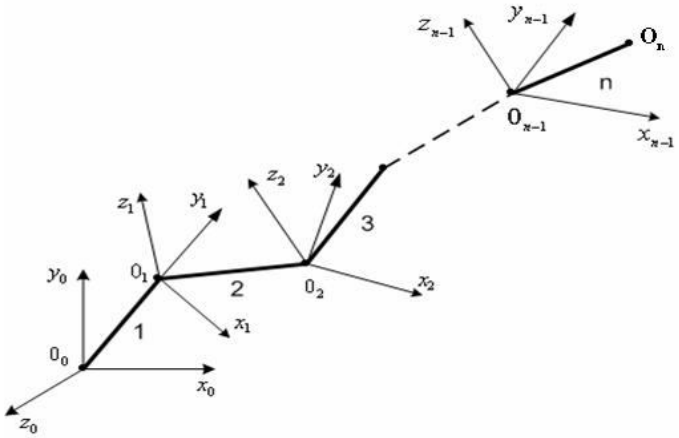


Рисунок 1 - Системы координат многосекционного манипулятора
(1, 2, …, n – номера секций; 0_n – схват манипулятора)

Введем следующие обозначения [1]: T_i, $i \in [1:n-1]$ - однородная (4×4)-матрица, задающая переход от системы координат $0_i x_i y_i z_i$ к системе координат $0_{i-1} x_{i-1} y_{i-1} z_{i-1}$; q_i - вектор обобщенных координат i-ой секции манипулятора; $q = (q_1, q_2, ..., q_{n-1})^T$ - вектор обобщенных координат n-секционного манипулятора; $T_n = T_1 T_2 .... T_{n-1}$ - (4×4) однородная матрица, определяющая положение системы координат $0_{n-1} x_{n-1} y_{n-1} z_{n-1}$ в системе координат $0_0 x_0 y_0 z_0$; s - (4×1) обобщенный вектор положения характерной точки рабочего органа 0_n в системе координат $0_{n-1} x_{n-1} y_{n-1} z_{n-1}$; S - (4×1) аналогичный вектор положения схвата 0_n в системе координат $0_0 x_0 y_0 z_0$.

Во введенных обозначениях имеем

$$S = T_n(q)s = F(q)\,, \qquad (1)$$

где $q \in Q$; Q - совокупность допустимых значений вектора обобщенных координат q, формируемых конструктивными и функциональными ограничениями на компоненты вектора q.

В разработанном программном продукте решается задача приведения схвата манипулятора в требуемое положение из заданного начального положения, т.е. обратная позиционная задача. Здесь и далее для простоты записи называем точку 0_n схватом манипулятора.

В общем виде обратная позиционная задача формулируется следующим образом: при заданном векторе $\boldsymbol{S}$ найти допустимые обобщенные координаты манипулятора $q \in Q$, удовлетворяющие уравнению (1)

Уравнение (1) представляет собой систему трех нелинейных тригонометрических уравнений с $m(n-1)$ неизвестными, где m – размерность каждого из векторов $q_i, i \in [1:(n-1)]$. Эта система уравнений может не иметь ни одного решения (требуемое положение схвата недостижимо), иметь одно решение, иметь бесконечно много решений (требуемое положение схвата может обеспечить бесконечное количество конфигураций манипулятора).

Методы решения задачи. В разработанном программном обеспечении для решения задачи применяются итерационный и оптимизационный методы. Важно, что итерационный метод, в отличие от оптимизационного, позволяет получить не только искомое конечное положение манипулятора, но и набор его промежуточных положений, соответствующих переводу манипулятора из начального положения в искомое. С другой стороны, оптимизационный метод при необходимости позволяет достаточно легко учитывать дополнительные требования, предъявляемые к искомой конфигурации манипулятора (путем интеграции их в целевую функцию). Далее рассматриваем только итерационный метод.

Суть итерационного метода решения обратной позиционной задачи поясним на простейшем примере двухзвенного манипулятора (рисунок 2), где принято, что шарнир *A* располагается на основании манипулятора; требуемое положение точки С (схвата) обозначено крестиком.

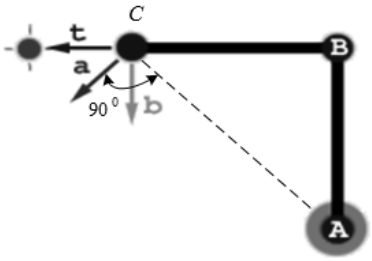


Рисунок 2 – Простейший двухзвенный манипулятор

Поворот на малый угол манипулятора вокруг шарнира *A* приводит к смещению схвата в направлении вектора *a*, а такой же поворот вокруг шарнира *B* – к смешению схвата в направлении вектора *b*. Для приведения манипулятора в искомое положение, очевидно, нужно сместить схват *C* в направлении вектора *t*. При этом вращение вокруг шарнира *B* приведет лишь к ухудшению ситуации, а вокруг шарнира *A*, напротив, поможет слегка уменьшить расстояние от схвата *C* до цели.

На основе этих соображений предлагается давать приращения обобщенным координатам в каждом из сочленений манипулятора на величину, пропорциональную тому, насколько это приращение приблизит схват *C* к цели (рисунок 3).

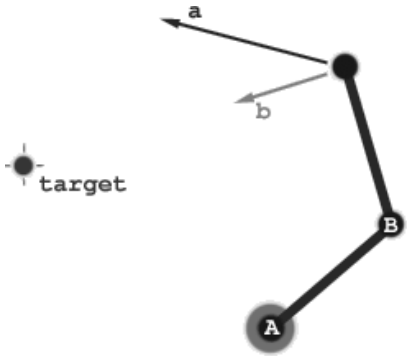


Рисунок 3 – К итерационному методу решения обратной позиционной задачи

Указанную величину приращений обобщенных координат можно оценить при помощи простых геометрических соображений (рисунок 4). На этом рисунке **joint** – рассматриваемый шарнир; **target** – целевая точка; **tip** – точка схвата; **axis** – ось вращения шарнира **joint**; **ToTip** – вектор, направленный из шарнира в точку схвата; **movement_vector** – вектор, перпендикулярный **ToTip**; **ToTarget** – вектор, направленный из точки схвата в целевую точку. Искомой величиной может служить скалярное произведение векторов и **movement_vector** и **ToTarget** [2].

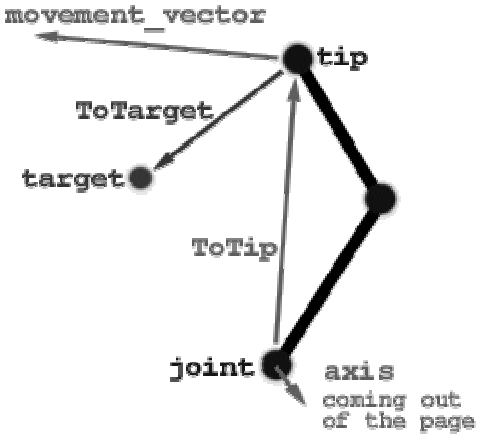
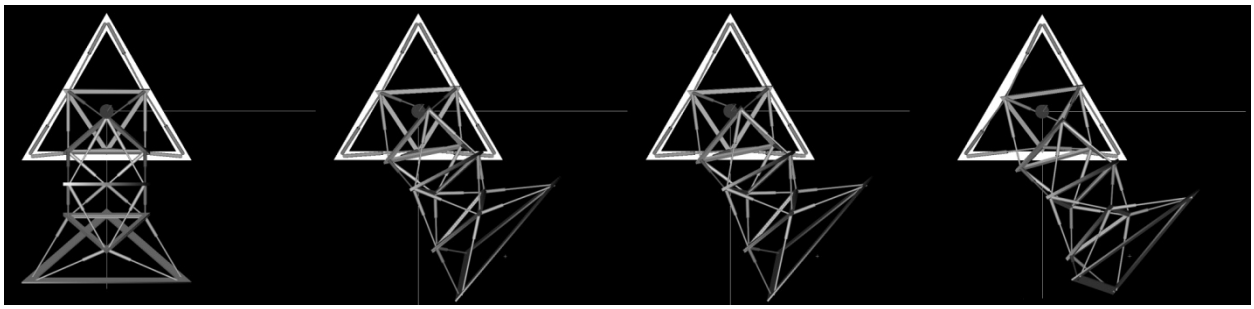
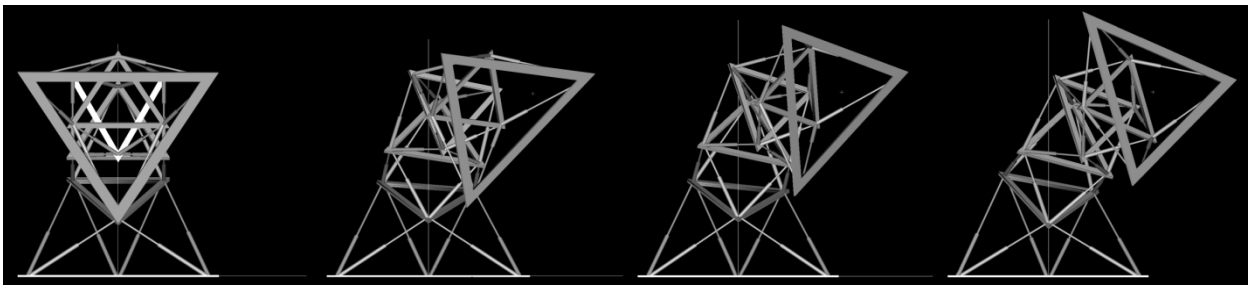


Рисунок 4 – К оценке величины приращений обобщенных координат итерационным методом

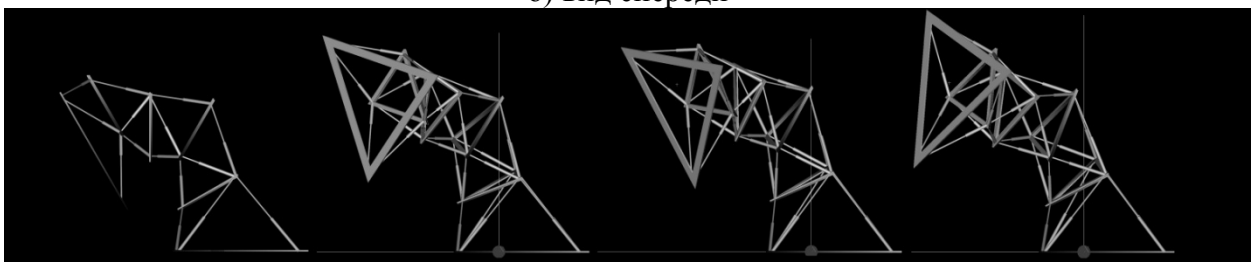
Результаты. При решении задачи позиционирования манипулятора можно использовать два подхода. В первом случае переход в требуемое положение начинается непосредственно из текущего положения, а во втором осуществляется предварительное выведение манипулятора в вертикальное положение. Реализацию одного из подходов иллюстрирует рисунок 5.



а) вид сверху



б) вид спереди



в) вид сбоку

Рисунок 5 – Движение манипулятора из произвольного начального положения в заданное конечное

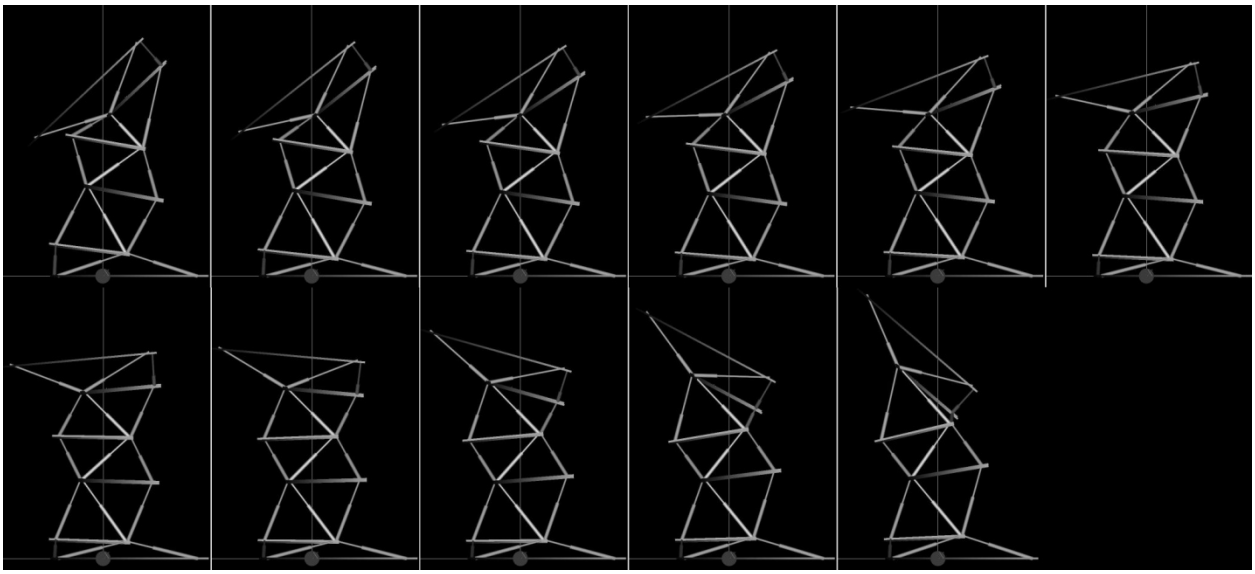


Рисунок 6 – Раскадровка поворота рабочего органа манипулятора в вертикальном положении (читать сверху вниз и слева направо)

Разработанное программное обеспечение также поддерживает решение задачи поворота рабочего органа манипулятора на заданный угол без смещения его схвата вдоль осей координат (рисунок 6).

Литература

1. Зенкевич С.Л., Ющенко А.С. Основы управления манипуляционными роботами.— МГТУ им. Н.Э. Баумана, 2004.— С. 480.
2. Hugo Elias. Inverse Kinematics - Improved Methods. — (http://freespace.virgin.net/hugo.elias/models/m_ik2.htm).

АВТОМАТИЧЕСКАЯ КЛАССИФИКАЦИЯ ТЕКСТОВЫХ ДОКУМЕНТОВ С ПОМОЩЬЮ ОНТОЛОГИЙ

Грушин М.А.
Научный руководитель: д.т.н., профессор, Норенков И.П.
МГТУ им. Н.Э.Баумана, Москва, Россия

AUTOMATED TEXT DOCUMENT CATEGORIZATION USING ONTOLOGIES

Grushin. M.A.
Supervisor: Dr., Prof., Norenkov I.P.
MSTU, Moscow, Russia

Аннотация
В статье рассматривается подход к автоматизации классификации текстовых документов с использованием онтологий. Кратко представлены современные методы и подходы к решению данной задачи. Рассмотрены преимущества и недостатки использования онтологий для задач извлечения из текстовых документов и последующей обработки.

Abstract
This article presents an approach to automation of text categorization using ontologies. Modern methods and approaches to solution of this task are presented briefly. Advantages and disadvantages of using ontologies for knowledge extraction from text documents and knowledge processing are considered.

Введение

Темпы роста количества документов научно-технической области в электронном виде и их доступность в сети Интернет приводят к тому, что на сегодняшний день подавляющая часть информации хранится на компьютерах в виде электронных текстовых документов. В большинстве организаций значительная часть полезных знаний содержится в документальных базах данных. Такая ситуация обуславливает повышенный интерес к области Text mining – методам автоматического извлечения и обработки знаний из текстовых документов. Получение знаний в автоматическом режиме затрудняется слабой структурированностью текстов на естественном языке. Такие знания могут быть с лёгкостью извлечены экспертом, но, с учётом огромного количества электронных документов, их эффективная обработка человеком становится весьма затратной как по времени, так и по ресурсам.

Извлечение знаний имеет своей конечной целью информационную поддержку эксперта или автоматизированной системы при принятии проектных решений. В документах, созданных специалистами, могут быть описаны подходы к решению различных проблем, рекомендации к подбору параметров и прочие знания, полезные в различных областях деятельности организации. Таким образом, основной функциональностью систем извлечения знаний является информационный поиск полезных сведений в документальных базах. Однако наряду с этой задачей должны решаться и промежуточные задачи автоматической классификации, кластеризации и аннотирования документов. В течение нескольких десятков лет разрабатываются новые методы решения подобных задач над текстами на естественном языке, а также повышается эффективность уже существующих подходов.

В системах управления знаниями (Knowledge Management) многих организаций экспертами составляются предметные онтологии, описывающие предметные области, в которых специализируются организации. Модели знаний, описываемых онтологиями, представляют собой набор ключевых понятий («концептов») и связей между ними. Применение онтологий для решения задач Text mining способствует повышению эффективности решения задач извлечения и обработки знаний из текстовых документов.

В данной работе рассматривается подход к решению задачи классификации текстовых документов с использованием ролевой кластеризации концептов онтологий, предложенной в работе [1].

Онтологии и ролевая кластеризация концептов

Онтология – один из способов представления знаний в интеллектуальных системах. Под онтологией понимают систему понятий (концептов, сущностей), отношений между ними и операций над ними в рассматриваемой предметной области, иными словами, онтология – это спецификация содержания предметной области. Например, онтология «Интеллектуальные системы» может выглядеть следующим образом:

Интеллектуальные системы = {интеллект; нейрон; нейронная сеть; обратная связь; логика; знания;...},

и также включать в себя связи между концептами вроде «*Обратная связь* – свойство *нейронной сети*».

Ролевая кластеризация концептов онтологии по классам (ролям) «объект», «средство», «свойство» и «действие» подразумевает собой распределение концептов в различные смысловые категории. Концепты этих категорий называются «простыми». Возможные комбинации (паттерны) ролей можно использовать для создания «сложных» концептов из набора простых. Таким образом, словосочетание «метод анализа быстродействия компьютера» можно рассматривать как сложный концепт, состоящий из 4-ёх простых, принадлежащих вышеприведённым ролям. В данном примере «метод» - средство, «анализ» - действие, «быстродействие» - свойство, «компьютер» - объект.

Применение подобного подхода в задачах информационного поиска по документам в пределах, по крайней мере, одной предметной области позволяет в некоторой степени учесть семантику понятий, составляющих запрос, тем самым повысив точность поиска. При запросе «анализ метода» поисковая система будет иметь представление о том, что нас интересует метод, как объект, и анализ, как действие над ним. При этом будут отсечены ненужные документы, релевантные запросу «метод анализа», которые составят практически половину выдачи системы в случае использования классических подходов к поиску. Также применение ролевой кластеризации способно повысить полноту поиска засчёт расширения запроса синонимами концептов, имеющими ту же ролевую принадлежность.

Также онтологии могут быть успешно применены при решении задач классификации и кластеризации документов, их аннотирования, упорядочения и задач поддержки принятия решений, как это описано в работе [2]. Во всех указанных случаях применения онтологии представляют собой набор значимых концептов, определяющих предметную область. Их применение позволяет избежать потерь машинного времени на анализ понятий, не входящих в предметную область, а в случае задачи классификации – не проводить временами весьма затратное обучение классификатора документов на обучающей выборке, так как классификатор представлен составленной онтологией. С другой стороны, качество решения указанных задач становится весьма зависимым от качества и полноты составленной онтологии.

Классификация документов

Для специалиста, пользующегося документальной базой данных в ходе работы, может быть интересен не весь набор документов, а только соответствующий интересующей его предметной области. Таким образом, становится актуальной задача классификации документов в базах данных по категориям. Классификация документов также имеет место в задачах фильтрации спама, распределении писем по тематике, системах электронной коммерции и многих других областях применения интеллектуальных систем. Также предварительное отнесение документа к классу в задаче информационного поиска позволяет отсечь документы, не относящиеся к тематике запроса, тем самым экономя время и вычислительные ресурсы.

Первоначальным методом классификации было ручное распределение документов по тематикам. Однако на сегодняшний день количество доступных для обработки документов

огромно, а это влечёт за собой несоизмеримые с выгодой затраты средств и времени при работе экспертов. Поэтому с 1960-х годов всё больший интерес вызывает вопрос автоматической классификации текстовых документов. Развитие методов решения задачи автоматической классификации и сами методы достаточно полно писаны в работе [3].

Изначальный подход для автоматизации работы эксперта в этой области состоял в написании им правил вида «*если - то*» для системы обработки текста, которая бы относила документ к определённой тематике при выполнении условий, заданных экспертом. Более формально классифицирующее выражение выглядит следующим образом

$$Если(ДНФ) \rightarrow То(категория),$$

где *ДНФ* – условия, выраженные в дизъюнктивной нормальной форме, *категория* – тематика, к которой следует отнести документ при истинности ДНФ. Метод прост и достаточно эффективен, однако очевидно, что он требует работы эксперта для поддержания актуальности правил и их написания.

В начале 90-х годов прошлого века экспертные правила были вытеснены методами машинного обучения. Преимущества такого подхода очевидны: системы требуют гораздо меньше экспертной поддержки и не нуждаются в написании правил классификации. Правила формируются самими системами на основе обучающей выборки. В настоящий момент наиболее популярными для решения задачи классификации являются наивный байесовский классификатор, метод Роккио, метод «*k* ближайших соседей», метод опорных векторов и различные модификации этих методов. Все эти подходы, кроме вероятностного байесовского классификатора, используют векторное представление документа, в котором содержимое представляется в виде вектора терминов, входящих в документ. Классификатор представляет собой особый документ, вектор которого формируется на этапе обучения и состоит из усреднённых значений весов терминов, входящих в документы обучающей выборки. Указанные методы имеют довольно много общего и отличаются лишь методом обучения и составления вектора-классификатора. Сама классификация является вычислением угла между двумя векторами, как степени их схожести: если вектор документа близок к вектору классификатора, то документ будет отнесён к данной категории.

Если для классификации используется онтология предметной области, то вектор документа можно сравнивать с вектором самой онтологии. Отсюда следует два важных отличия от классических методов машинного обучения. Применение онтологий позволяет отказаться от этапа обучения классификатора. Описание предметной области в виде онтологии само является классификатором, таким образом, не тратится время и вычислительные ресурсы на построение среднего документа из обучающей выборки. Второе отличие заключается в том, что при таком подходе в вектор документа включаются только те термины, которые включены в рассматриваемую онтологию. Это значит, что те понятия, которые не входят в набор концептов онтологии, уходят из процесса вычисления весов терминов. Также имеется отличие классификатора в виде онтологии от классификатора в виде усреднённого документа. В обоих случаях классификатор является моделью «эталонного» документа, соответствующего предметной области. Но если он является «усреднённым» документом, то в его состав могли войти термины, употреблявшиеся в документах, но не имеющие отношения к описываемому разделу. В случае онтологии, напротив, классификатор является описанием предметной области без каких-либо лишних для неё понятий. В таком случае этот классификатор является более универсальным с точки зрения использования его в составе различных систем и для различных задач.

Степень соответствия документа классу (онтологии) рассчитывается, как сумма весов всех терминов данной онтологии, найденных в документе:

$$R_{dC} = \sum_{t \in C} w_{td} ,$$

где R_{dC} - степень соответствия документа *d* кластеру *C*, w_{td} - вес термина *t* в документе *d*.

Существуют различные подходы к взвешиванию терминов в документе, о них можно узнать в источнике [4]. С учётом того, что концепты в онтологиях разделены по ролям,

необходимо ввести различные веса для ролей, а также отдельно обрабатывать сложные концепты. Хорошо зарекомендовал себя следующий способ взвешивания

$$w_{td} = \begin{cases} tf, \text{если концепт простой и его роль "объект"}, \\ 0.1 \cdot tf, \text{если концепт простой}, \\ (1+k) * tf, \text{если концепт сложный}, \end{cases} \quad (1)$$

где tf – количество вхождений концепта в документ, k – коэффициент, учитывающий сложность концепта.

Следует отметить неравномерность значимости разных ролей концептов для классификации документа. Существует предположение, что предметная область наиболее полно характеризуется концептами типа «объект», в то время как концепты остальных типов могут быть инвариантны по отношению к разным предметным областям. Это было подтверждено экспериментально в ходе проверки классификатора – если присваивать всем простым концептам, встреченным в документе, равные веса, соответствующие количеству их вхождений в текст, то документ будет с равной вероятностью отнесён ко всем классам сразу. Отсюда следует, что наиболее значимыми нужно сделать понятия-объекты, а остальным присваивать веса по другой схеме. Как видно из формулы (1), предложенная схема взвешивания учитывает это замечание. Также в формуле (1) учитывается повышение веса сложного термина в зависимости от количества простых, составляющих его.

В серии экспериментов классификатор с использованием онтологий показал себя хорошо, процент правильно классифицированных документов довольно высок. Остаётся лишь отметить, что в данном подходе качество классификации напрямую зависит от качества созданной онтологии.

Заключение

В настоящее время всё больше организаций используют в своей работе системы управления знаниями, которые используются для решения широкого круга задач. В данной работе рассмотрен подход к решению одной из таких задач – классификации документов. С одной стороны этот подход имеет тот же минус, что и подход с составлением правил классификации вручную – требуется работа экспертов для создания описаний предметных областей и поддержания их актуальности. Но с другой стороны такой подход является более гибким. Онтологии могут быть использованы в ряде других задач управления знаниями, поэтому их создание и поддержка более оправдано с точки зрения затрат. Также отнесение документа к тематике происходит не просто по факту наличия в нём определённых терминов, но на основе вычисления меры близости документа и онтологии. Таким образом, рассмотренный подход в некоторой степени объединяет в себе эффективность старого и универсальность современного подхода к классификации документов.

Литература

1. Норенков И.П., Уваров М.Ю. Поддержка принятия решений на основе паттернов проектирования. // Электронное научно-техническое издание «Наука и образование», 2011, 9.
2. Норенков И.П. Задачи управления знаниями, извлекаемыми из текстовых документов. // Электронное научно-техническое издание «Наука и образование», 2011, 9.
3. Fabrizio S. Machine learning in automated text categorization// ACM Computing Surveys, 2002, 34(1), pp 1-47.
4. Manning C., Raghavan P., Schütze H.. Introduction to Information Retrieval. - Cambridge University Press, 2008, 544 p.
5. Bevainyte A., Butenas L. Document classification using weighted ontology// Materials Physics and Mechanics, 2010, №9, pp. 246-250.

АЛГОРИТМ ПОИСКА КУКУШКИ ДЛЯ РЕШЕНИЯ ЗАДАЧИ ГЛОБАЛЬНОЙ БЕЗУСЛОВНОЙ ОПТИМИЗАЦИИ

Бенза Н.Н.
Научный руководитель: д.т.н., профессор, Карпенко А.П.
МГТУ им. Н.Э.Баумана, Москва, Россия

CUCKOO SEARCH ALGORITHM FOR UNCONSTRAINED OPTIMIZATION PROBLEMS

Benza N.N.
Supervisor: Dr., Prof., Karpenko A.P.
MSTU, Moscow, Russia

Аннотация
В статье рассматриваются основные этапы алгоритма поиска кукушки для решения задач глобальной безусловной оптимизации. Подробно исследовано влияние варьируемых параметров поиска. Кратко представлено использование в алгоритме поиска кукушки разновидности случайных блужданий - полетов Леви. В заключении даны рекомендации по выбору оптимальных значений варьируемых параметров поиска.

Abstract
This paper presents the Cuckoo search algorithm for unconstrained optimization problems. The effect of varying the search parameters was studied in detail. Cuckoo search in combination with the Lévy flights is briefly presented. The recommendations on the choice of optimal values of the variable parameters are given in the conclusion.

Введение

В последнее время появляется все большее число метаэвристических алгоритмов оптимизации, вдохновленных поведением живых существ в природе. Например, метод роя частиц авторов вдохновило поведение стай птиц и косяков рыб, алгоритм светлячков был инспирирован характером свечения тропических светлячков и так далее.

Предметом исследования является новый алгоритм глобальной оптимизации – алгоритм поиска кукушки, который разработали Xin-She Yang и Susan Deb в 2009 году [1]. На создание алгоритма авторов вдохновило поведение кукушек в процессе вынужденного гнездового паразитизма, когда некоторые виды кукушек подкладывают яйца в гнезда птиц других видов [2-4].

1 Постановка задачи и схема алгоритма

Рассматриваем задачу глобального безусловной минимизации вида

$$\min_{x \in R^d} F(x) = F(x^*) = F^*, \qquad (1)$$

x - d-мерный вектор варьируемых параметров, $F(x)$ - целевая функция.

В алгоритме поиска кукушки используются следующие основные предположения. Каждое яйцо в гнезде представляет собой решение, а яйцо кукушки - новое решение. Цель заключается в использовании новых и потенциально лучших решений (кукушкиных), чтобы заменить менее хорошие решения в гнездах. В простейшей форме алгоритма в каждом гнезде находится по одному яйцу. Алгоритм может быть расширен для более сложных случаев, когда в каждом из гнезд находится более одного яица, представляющих некоторое множество решений.

Поиск кукушки основан на трех следующих правилах.

1) Каждая кукушка откладывает одно яйцо за один раз, и подкладывает его в гнездо, которое выбирается случайным образом.

2) Лучшие гнезда с яйцами высокого качества (пригодными решениями) переходят на следующее поколение.

3) Число доступных гнезд фиксировано, а яйцо кукушки может быть обнаружено хозяином гнезда с вероятностью $p_a \in (0,1)$. Обнаруженные решения исключаются из дальнейшего рассмотрения.

Основные этапы поиска кукушки можно представить в виде псевдокода следующим образом.

```
begin
     Генерация начальной популяции n гнезд xj, (j=1,2,...,n);
     while (критерий останова)
           Случайным образом поместить кукушку в точку xi, выполняя
           полеты Леви;
           Случайным образом выбрать гнездо j среди n гнезд;
           if (Fi<Fj)
               Заменить xj, на новое решение xi;
           end if
           Часть гнезд, обнаруженных с вероятностью pa, удалить из
           популяции и построить такое же число новых;
           Сохранить лучшее решение(гнездо);
     end while
     Пост-обработка результатов и визуализация;
end
```

2 Полеты Леви

При создании новых решений $x(t+1) = x^{t+1}$ для кукушки i полеты Леви осуществляются по формуле

$$x_i^{(t+1)} = x_i^{(t)} + \alpha \oplus Lévy(\lambda), \tag{2}$$

где α>0 - размер шага, который должен быть связан с масштабами задачи поиска. В большинстве случаев, можно использовать α=1. Уравнение (2) представляет собой стохастическое уравнение случайных блужданий. В общем, случайное блуждание представляет собой цепь Маркова, для которой следующее положение зависит только от текущего местоположения (первое слагаемое в уравнении) и вероятности перехода (второе слагаемое). Случайную длину шага в уравнении (2) определяет распределение Леви

$$Lévy(\lambda) = t^{-\lambda}, \quad (1<\lambda\leq 3). \tag{3}$$

Некоторые из новых решений должны быть порождены полетами Леви вокруг текущего лучшего решения, что ускорит сходимость алгоритма. Значительная часть новых решений должна быть сформирована в случайно выбранных точках пространства поиска, т.е. достаточно далеко от текущих лучших решений. Это позволит диверсифицировать поиск, повысить вероятность преждевременной сходимости.

3 Исследование эффективности алгоритма

В рамках исследований проводился анализ эффективности алгоритма поиска кукушки при изменении числа гнезд n и вероятности обнаружения гнезда p_a. В докладе представлены результаты численных экспериментов при поиске минимума функции Растригина, которая имеет вид

$$f(x) = 10d + \sum_{i=1}^{d} (x_i^2 - 10\cos(2\pi x_i)). \tag{4}$$

Результаты исследования иллюстрируют рисунки 1 – 3 и таблицы 1, 2, на которых принято, что t_{end} - начало стагнации вычислительного процесса, n_f - число испытаний (вычислений значений целевой функции).

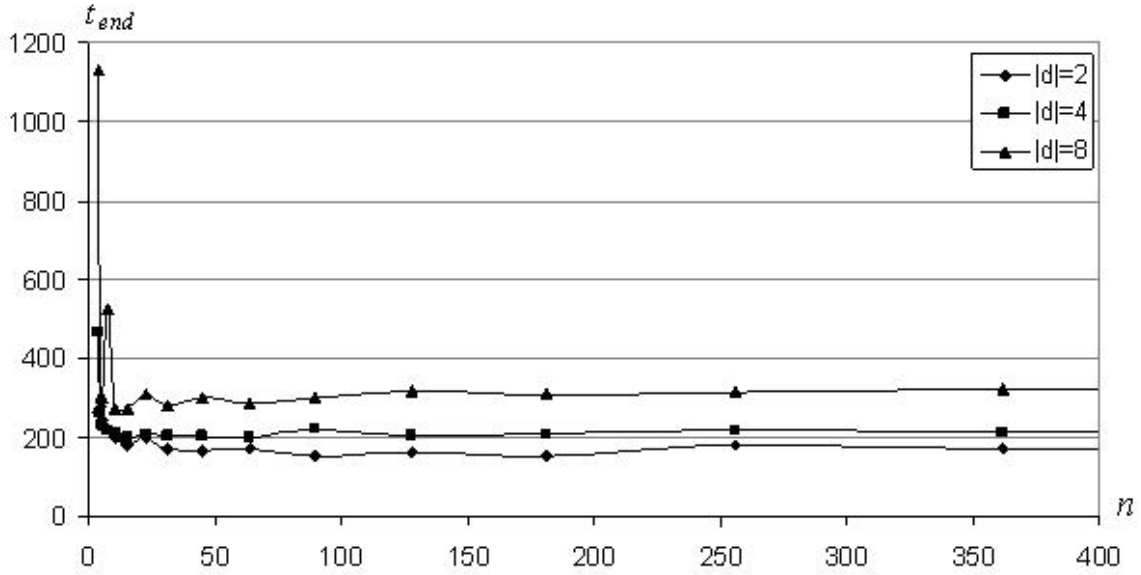


Рисунок 1 – Число итераций t_{end} в зависимости от числа гнезд n:
функция Растригина; p_a = 0,25

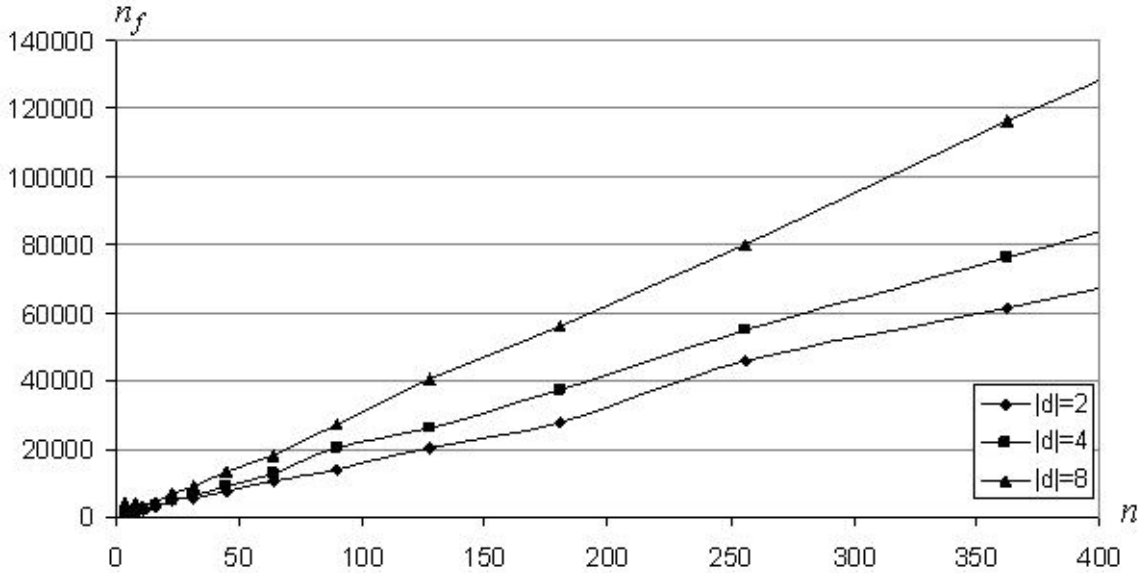


Рисунок 2 – Число испытаний n_f в зависимости от числа гнезд n:
функция Растригина; p_a = 0,25

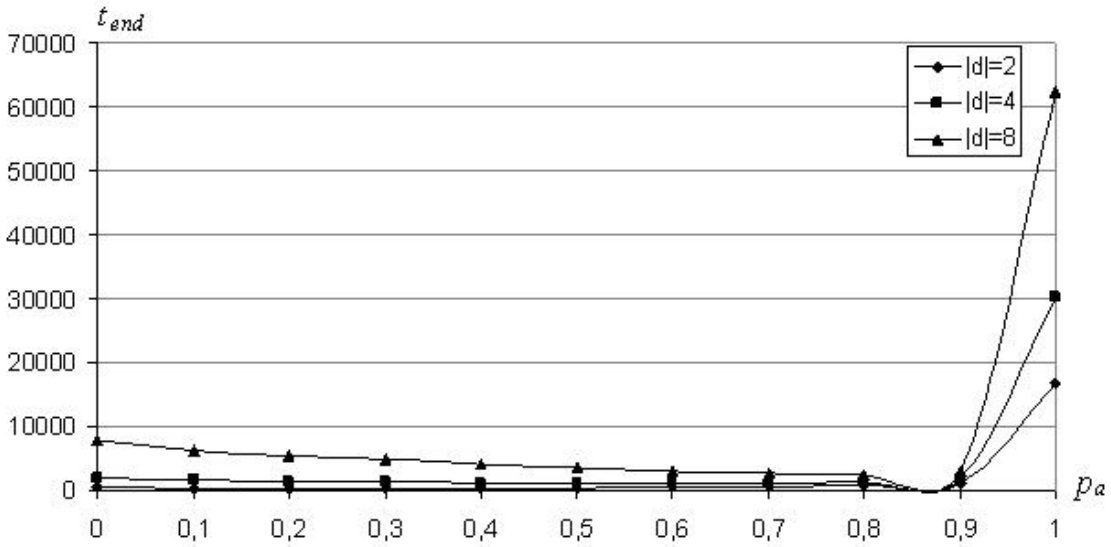


Рисунок 3 – Число итераций t_{end} в зависимости от вероятности p_a : функция Растригина; n = 32

Таблица 1 – Вероятность локализации минимума функции Растригина при различных значениях параметров n, d

n	d=2	d=4	d=8
4	80	17	0
5	94	49	0
6	97	73	7
8	100	96	41
11	100	100	83
16	100	100	98
23	100	100	100
32	100	100	100
45	100	100	100
64	100	100	100
90	100	100	100
128	100	100	100
181	100	100	100
256	100	100	100

Таблица 2 – Вероятность локализации минимума функции Растригина при различных значениях параметров p_a, d; n=32

p_a	d=2	d=4	d=8
0	100	100	91
0,1	100	100	100
0,2	100	100	100
0,3	100	100	100
0,4	100	100	100
0,5	100	100	100
0,6	100	100	100
0,7	100	100	100
0,8	100	100	100
0,9	100	100	100
1	95	87	100

Рисунки показывают, что при увеличении числа гнезд n скорость сходимости алгоритма повышается и одновременно возрастает число вычислений значений целевой функции. При увеличении размерности пространства d заметно возрастает суммарное число испытаний.

На основе результатов, представленных в таблице 1, можно утверждать, что при d=2 вероятность нахождения минимума функции Растригина будет 100% при числе гнезд, большем 8; при d=4 следует выбирать значения n, большие 10; при d=8 значение n должно превосходить 20.

При исследовании влияния параметра p_a использовалось фиксированное значение числа гнезд n=32. Из рисунка 3 следует, что сходимость алгоритма не чувствительна к значениям этого параметра, если $p_a \in [0; 0{,}9]$. На основании данных таблицы 2 можно заключить, что значения величины p_a следует выбирать из интервала [0,1; 0,9], поскольку такие значения обеспечивают 100% вероятность обнаружения минимума целевой функции.

Заключение

В работе рассматривается новый метаэвристический алгоритм глобальной оптимизации – алгоритм поиска кукушки, использующий полеты Леви. На примере сложной многоэкстремальной функции Растригина выполнено исследование эффективности алгоритма в зависимости от его основных параметров. На основании результатов исследования можно сделать вывод, что алгоритм поиска кукушки обладает высоким потенциалом развития.

Литература

1. Yang X.-S., S. Deb S. Cuckoo search via L´evy flights / In: Proc. Of World Congress on Nature & Biologically Inspired Computing (NaBIC 2009), 2009, India, pp. 210-214.

2. Tuba M., Subotic M., Stanarevic N. Modified cuckoo search algorithm for unconstrained optimization problems / In: Proceedings of the European Computing Conference, 2010, pp. 263-268.

3. Valian Eh., Mohanna Sh., Tavakoli S. Improved cuckoo search algorithm for feedforward neural network training / In: International Journal of Artificial Intelligence & Applications (IJAIA), 2011, Vol.2, No.3, pp. 36-43.

4. http://en.wikipedia.org/wiki/Cuckoo_search

РАЗРАБОТКА ПРОГРАММНОГО СРЕДСТВА ДИНАМИЧЕСКОГО АНАЛИЗА ИЗОБРАЖЕНИЯ С ИСПОЛЬЗОВАНИЕМ ЦИФРОВОЙ ГОЛОГРАФИИ

Шибитов И.А.
Научный руководитель: к.т.н., доцент Волосатова Т.М.
Кафедра САПР(РК6), МГТУ им. Н.Э. Баумана , Москва , Россия.

DEVELOPMENT OF PROGRAM TOOL FOR DYNAMIC ANALYSIS OF AN IMAGE BY MEANS OF DIGITAL HOLOGRAPHY

Shibitov I.A.
The scientific chief: Ph.D., assistant professor Volosatova T.M.
CAD/CAE department (RK6), BMSTU , Moscow , Russia.

Аннотация
Разработано программное средство динамического анализа изображения с использованием цифровой голографии. В качестве исходных данных использованы файлы с плоскими изображениями кадров динамически меняющегося тест-объекта. За основу взята цифровая модель голограммы Фурье. Предложенное средство реализует свойство зависимости фазового множителя от положения голографируемого объекта при сохранении инвариантности собственно голограммы. В качестве перспективы развития рассматривается оптимизации программного обеспечения с дальнейшим анализом свойств и способов применения голограмм Фурье.

Abstract
Program tool for dynamic analysis of an image by means of digital holography was developed. Files with flat images of dynamically changing test object are used as an initial data. The analysis of image is based on so property as association of phase multiplier from position of object at preservation of invariance hologram. Software optimization, further analysis of Fourier holograms properties and applications are considered as development prospects.

Системы технического зрения предназначены для замены человека в сфере решения задач, связанных со сбором и анализом зрительной информации.

Разработанный программный продукт предназначен для детектирования перемещения объекта в исследуемой последовательности кадров данного видео ряда.

Для решения данной проблемы использован метод цифровой голографии. Существуют несколько типов голограмм: Фурье, Френеля, Ли и т.д. Общая схема получения Фурье-голограммы по методу Вандер Люгта представлена на рисунке 1.

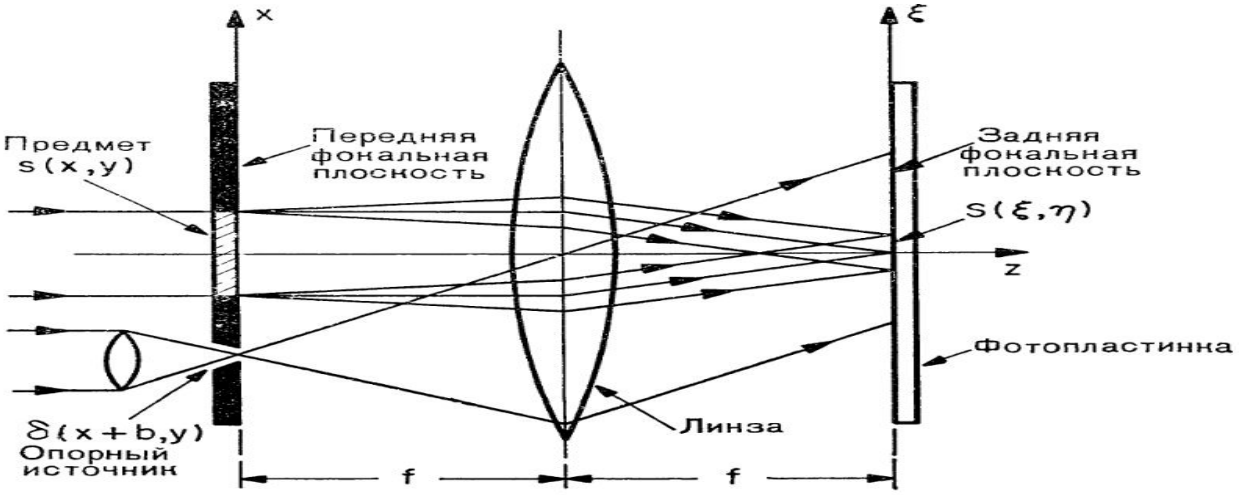


Рисунок 1 - Схема получения Фурье-голограммы (размеры линзы даны не в масштабе)

Фурье-голограммой мы называем голограмму в том случае, если на ней регистрируется интерференция двух волн, комплексные амплитуды которых в плоскости голограммы

являются фурье-образами предмета и опорного источника. Если $s(x,y)$ — пропускание транспаранта, помещенного в передней фокальной плоскости линзы, то амплитуда предметной волны в плоскости голограммы, совпадающей с задней фокальной плоскостью линзы,есть $S(\xi,\eta)$. В передней фокальной плоскости расположен также точечный источник $\delta(x + b, y)$, фурье-образом которого является плоская волна с амплитудой $exp(—2\pi i\xi b)$. Это плоская волна играет роль опорной волны и так же, как и $S(\xi,\eta)$, освещает заднюю фокальную плоскость линзы. Интенсивность интерференционной картины, образованной двумя фурье-образами, описывается выражением

$$I = [exp (—2\pi i\xi b) + S(\xi,\eta)] [exp (2\pi i\xi b) + S^* (\xi,\eta)] =$$
$$= 1 + | S(\xi,\eta) |^2 + S(\xi,\eta) exp (2\pi i\xi b) + S^* (\xi,\eta) exp (—2\pi i\xi b).$$

Предположим, что проявленная голограмма имеет пропускание $t (x, y) \sim I$. Если голограмма освещается распространяющейся вдоль оси z плоской волной с постоянной амплитудой r_0, то произведение $r_0 t(x, y)$ представляет собой комплексную амплитуду W дифрагированного света непосредственно за голограммой:

$$W \sim r_0 t (x, y) \sim I = 1 + | S |^2 + S exp (2\pi i\xi b) + S^* exp (—2\pi i\xi b).$$

Линза, расположенная непосредственно перед голограммой или непосредственно после нее, будет создавать в задней фокальной плоскости поле, соответствующее произведению обратного фурье-образа функции W на фазовый множитель сферической волны.

В математической модели рис.1 интерпретируется следующим образом:

1. Предмету соответствуют файлы с плоскими изображениями кадров динамически меняющегося тест-объекта (видеоролик) .
2. Линза моделируется преобразованием Фурье от функции, кодирующей изображения.
3. Взаимодействие опорного пучка с распределением поля, моделируемым преобразованием Фурье от функции, кодирующей изображения, описывается сложением поля с комплексной амплитудой наклонной под произвольно выбранным углом плоской волны.
4. Фотопластинка регистрирует квадрат модуля распределения амплитуды волны.

При произвольном перемещении объекта в фотографии отображаются сдвиги вдоль координат Y и X, которые приводят к появлению фазовых множителей в преобразованиях Фурье, а именно:

$$exp [- 2\sigma j (\upsilon_{x\Delta} x + \upsilon_{y\Delta} y)],$$

где - $\upsilon_{x\Delta}$, $\upsilon_{y\Delta}$ - соответствующие пространственные частоты, $_\Delta x$, $_\Delta y$, пространственный сдвиг системы XOY относительно абсолютной системы координат. Как известно, Фурье – голограмма инвариантна к положению объекта в плоскости регистрации. Следовательно, передача пространственных частот в цифровых Фурье – образах инвариантна к положению объекта (пользователя). И только фазовый множитель содержит данные о движении объекта.

ПО разработано в среде MATLAB, т.к. она предоставляет широкий набор средств цифровой обработки изображений, что позволяет сосредоточить усилия не на разработке алгоритмов, а на реализации основной практической задачи.

Обработка входных данных происходит следующим образом.

На первом этапе происходит чтение видео ролика и разбиение его на кадры. Далее производится перевод данных из формата uint8 в double, и из RGB в оттенки серого, что необходимо для корректной работы функции, выполняющей Фурье-преобразование.

На втором этапе производится построение непосредственно голограмм Фурье для каждого кадра по ранее описанным зависимостям с использованием функции выполняющей БПФ - fft2().

На третьем этапе производится вычисление фазовых множителей по формуле

$$\varphi=atan(Im(I)/Re(I)),$$

которые как раз и являются детекторами перемещения объекта.

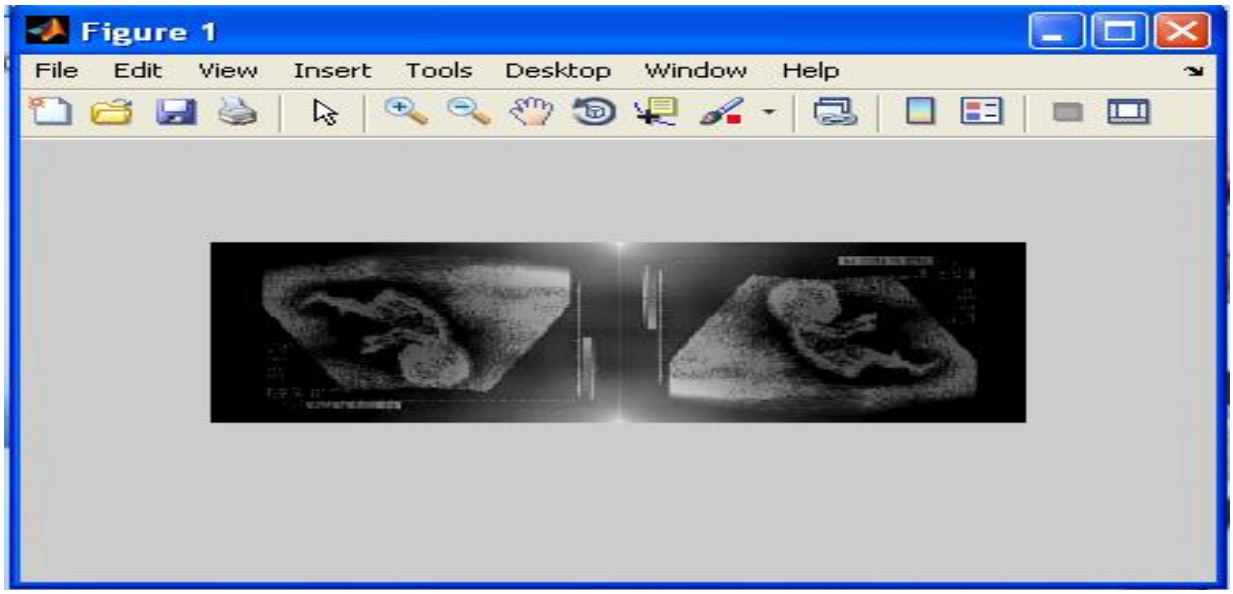


Рисунок 2 - Результат работы программы – мнимое и действительное изображения

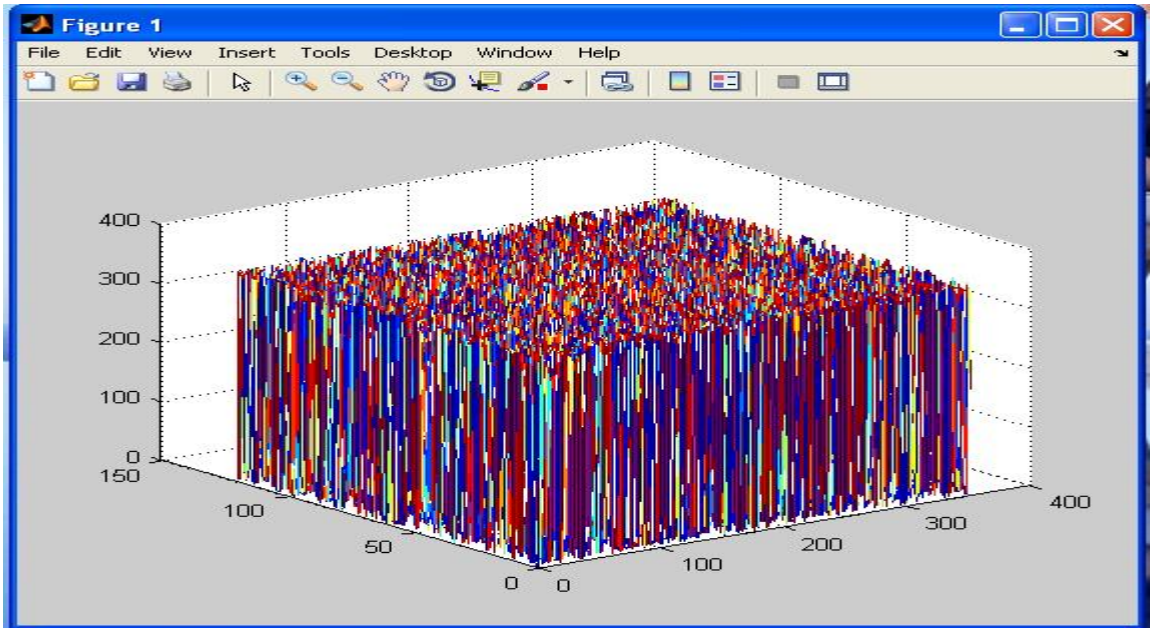


Рисунок 3 - Разность фаз для двух соседних кадров

Результатом работы программы является серия графиков, отображающих изменение фазовых множителей, анализ которых позволяет говорить об изменении положения объекта в кадре.

Литература

1. Ярославский Л.П., Мерзляков «Методы цифровой голографии», Наука, 1977
2. М.Франсон. Оптика спеклов. – М.:Мир, 1980, с.141- 143.
3. Г.Г.Левин, Г.Н.Вишняков Оптическая томография. – М.: Радио и связь, 1989.-224с.
4. Р. Кольер, К. Беркхарт, Л. Лин. Оптическая голография, Мир, Москва, 1973.

ИГРОВОЕ МОДЕЛИРОВАНИЕ ДОСТУПА. АЛГОРИТМЫ РАСКРАСКИ.

Быстров А.В.
Божко А.Н., к.т.н, доцент кафедры РК6
МГТУ имени Н.Э. Баумана, Москва, Россия

GAMING ACCESS MODELLING. COLORING ALGORITMS.

Bystrov A.V.
Bojko A.N., PhD , associated professor
BMSTU, Moscow, Russia

Аннотация
Рассматривается проблема моделирования доступа с точки зрения сведения моделирования к статистической игре раскраски вершин в диаграмме Хассе. Предлагаются алгоритмы выбора хода в статистической игре. Показана зависимость эффективности алгоритмов от количества вершин в раскрашиваемой диаграмме. Выявлены наиболее перспективные алгоритмы.

Abstract
Gaming approach to access modelling as a process of coloring Hasse diagram is given. Decision-making algorithms are researched. Dependence between algorithms efficiency and the number of nodes in the diagram is drawn. The most potential algorithms are named.

Введение

Моделирование доступа – это пример трудно-решаемой научной проблемы с предельно простой постановкой. Большинство подходов к этой проблеме основываются на синтезе правил перемещения трехмерных геометрических моделей объектов (деталей или сборочных единиц) и проверки условий непересечения. Реализация метода требует очень высоких вычислительных ресурсов для корпусных изделий с плотным монтажом и сложной конфигурацией составных частей.

В качестве альтернативы можно предложить подход, подразумевающий сведение задачи определения доступа к задаче игровой раскраски частичных порядков. Дано частично-упорядоченное множество (Q, **C**), все вершины которого являются неокрашенными. Ход ЛПР заключается в выборе неокрашенной вершины этого множества. Ответ второго игрока – природы – состоит в определении цвета предложенной для проверки вершины. Если вершина получила черный цвет, то все вершины порядкового идеала окрашиваются в черный цвет. Если проверяемая вершина получила белый цвет, то все вершины порядкового фильтра окрашиваются в белый цвет. Требуется окрасить частичный порядок за наименьшее число обращений к ЛПР.

Подробнее постановка задачи игры, а также дихотомическая стратегия выбора вершины для раскраски описаны в [1] и [2].

Вероятность, с которой вершина, находящаяся на определенном уровне диаграммы Хассе, является допустимой, зависит от положения этого уровня. Все вершины, находящиеся на нижнем уровне диаграммы (представляющие отдельные детали или сборочные единицы, сборка которых производилась вне проектируемого сборочного процесса) однозначно являются допустимыми. При повышении уровня вероятность появления допустимой вершины уменьшается. Исследуем диаграммы, раскраска которых уже известна. Грубо оценим вероятность появления допустимой вершины на каждом уровне диаграммы, разделив количество допустимых («черных») вершин на данном уровне на количество всех вершин этого уровня.

$$p_i = d_i / N_i$$

Здесь d_i - количество допустимых вершин i-го уровня, N_i - количество всех вершин на i-м уровне.

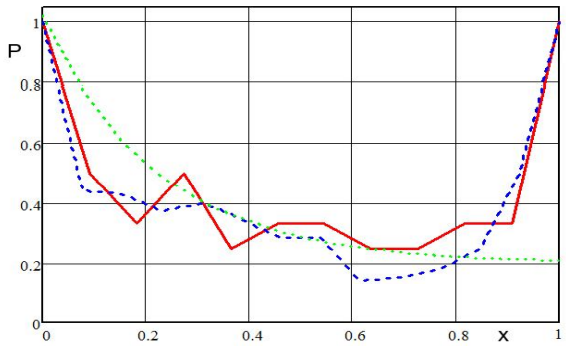


Рисунок 1 - Получение распределения вероятности. Красным (сплошным) и синим (пунктиром) обозначены кривые, полученные из раскрашенных диаграмм. Зеленым (точками) показана кривая, предложенная в качестве закона распределения

Предложим закон распределения вероятности появления допустимой вершины, индуцируемый на уровнях диаграммы Хассе, который будет близок к кривым на построенном графике с поправкой на поведение полученных кривых в верхних уровнях диаграммы.

На основе экспоненциального распределения путем введения дополнительных коэффициентов был получен следующий закон распределения:

$$P = \frac{4,5}{5,5} \cdot e^{-4,5 \cdot x} + 0,2$$

При достаточно большом количестве построенных кривых можно утверждать, что подобранный закон распределения в достаточной степени моделирует поведение «природы» при игровом моделировании доступа.

1. Методы выбора окрашиваемой вершины

1.1 Метод максимального математического ожидания выигрыша

Пронумеруем произвольно вершины диаграммы, для каждой из вершин посчитаем мощность верхнего и нижнего конусов и занесем полученные данные в игровую матрицу.

После составления платежной матрицы проведем процедуру сокращения, основанную на принципе Парето, подразумевающую сокращение доминируемых вариантов.

Платёжная матрица дополняется столбцом, каждый элемент которого представляет собой значение математического ожидания выигрыша при выборе соответствующей стратегии ЛПР:

$$W_i = \sum_{j=1}^{n} a_{jp} \cdot p_j \ ,$$

где p_j – вероятность j-го состояния окружающей среды.

Оптимальной по данному критерию считается та стратегия ЛПР, при выборе которой значение математического ожидания выигрыша максимально:

$$W = max W_i$$

1.2 Метод максимина с учетом вероятности

Минимизация риска проигрыша представляется ЛПР менее существенным фактором принятия решения, чем максимизация среднего выигрыша.

Правило выбора решения в соответствии с классическим максиминным критерием Вальда (ММ-критерием) можно интерпретировать следующим образом: платёжная матрица дополняется столбцом, каждый элемент которого представляет собой минимальное значение выигрыша в соответствующей стратегии ЛПР:

$$W_i = \min_j a_{ij}$$

Оптимальной по данному критерию считается та стратегия ЛПР, при выборе которой минимальное значение выигрыша максимально:

$$W = max W_i$$

Имея в своем расположении модель распределения вероятности допустимости вершины данный метод можно улучшить, учитывая вероятности появления вершины определенного «цвета».

Тогда W_i будет иметь следующий вид:

$$W_i = \min_j (a_{ij} \cdot p_{ij})$$

Очевидно, что модифицированный критерий максимина лучше «приспособлен» к условиям распределенной вероятности появления допустимой вершины.

1.3 Метод ожидаемого значения дисперсии

Замена задачи $W \to \max$ задачей $W = M(W) \to \max$ — не единственный способ перехода к статистической постановке. Можно поступить и иначе. Например, определенную роль может играть дисперсия критериальной функции *J*. Возможно имеет смысл поступиться значением математического ожидания для уменьшения возможного разброса результатов:

$$W = W(X) - k \cdot D(X),$$

где D(X) — дисперсия случайной величины W(X), k - заданная постоянная. Эту постоянную целесообразно интерпретировать как степень несклонности к риску. Используемый в этом методы критерий обычно называют критерием ожидаемого значения-дисперсии [3].

1.4 Метод, основанный на критерии Сэвиджа

В случае использования критерия минимаксного риска Сэвиджа величина $r_{ij} = a_{\max j} - a_{ij}$ (где $a_{\max j}$ – максимальный элемент *j*-го столбца) может быть интерпретирована как величина возможного проигрыша при выборе ЛПР I-й стратегии по сравнению с наиболее выгодной. Для определения оптимальной стратегии по данному критерию на основе платёжной матрицы рассчитывается матрица рисков, каждый коэффициент которой r_{ij} определяется по формуле:

$$r_{ij} = a_{max\,j} - a_{ij}$$

Матрица рисков дополняется столбцом, содержащим максимальные значения коэффициентов r_{ij} по каждой из стратегий ЛПР:

$$R_i = max\, r_{ij}$$

Оптимальной по данному критерию считается та стратегия, в которой значение R_i минимально:

$$W = min\, R_i\,.$$

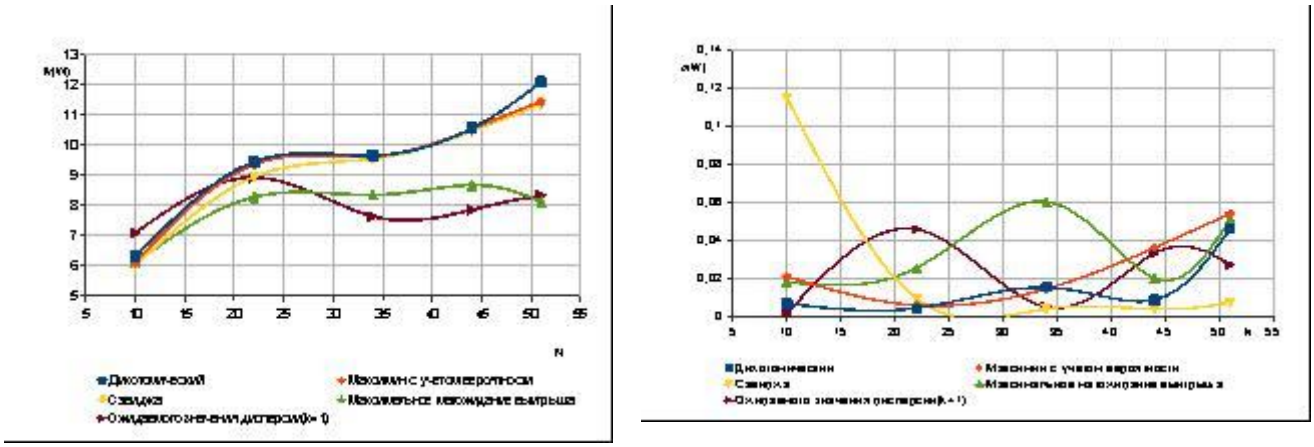
2 Эффективность алгоритмов

Эффективность алгоритма будем оценивать математическим ожиданием количества шагов, необходимых для полной раскраски диаграммы Хассе. Ниже представлен график зависимости математического ожидания количества шагов для раскраски от количества вершин N диаграммы Хассе для 500 различных раскрасок каждой диаграммы (рисунок 2).

Средняя степень вершины диаграмм, использованных в исследовании лежит в пределах отрезка [3.2, 3.6].

Как видно из графика, наиболее эффективными с точки зрения минимизации количества шагов для раскраски всей диаграммы являются метод максимального математического ожидания выигрыша и метод ожидаемого значения дисперсии. Разница в эффективности методов становится очевиднее с ростом количества вершин в диаграмме. При 50 узлах эти методы примерно на треть эффективнее остальных. При количествах вершин менее 20

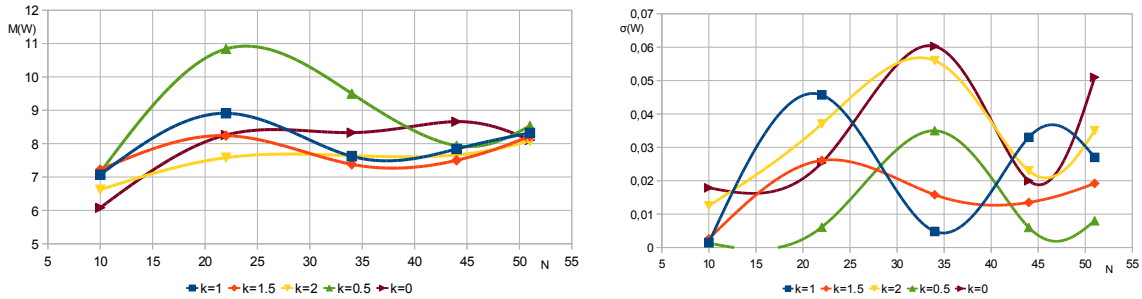
среднее квадратичное отклонение оценки алгоритма Сэвиджа являющееся значительным минимизируется и становится самым низким по сравнению с другими алгоритмами. Так же заметно, что модификации, примененные в методе ожидаемого значения дисперсии, позволяют уменьшить среднее квадратичное отклонение результатов оценки по сравнению с методом максимального математического ожидания выигрыша. Метод максимина с учетом вероятности показывает наибольшее стремление к росту среднего квадратичного отклонения при увеличении числа узлов по сравнению с другими методами.



а) б)

Рис.2 Математическое ожидание (а) и среднее квадратичное отклонение (б) исследуемых методов

Посмотрим изменение эффективности алгоритма ожидаемого значения дисперсии в зависимости от коэффициента *k* (рисунок 3).



а) б)

Рис.3 Математическое ожидание (а) и среднее квадратичное отклонение (б) метода ожидаемого значения дисперсии при разных *k*

Полагаясь на рис 3 можно рекомендовать значения *k* в пределах 1...1,5.

Заключение

Моделирование геометрических связей – это сложная задача, с которой сталкиваются инженеры в различных отраслях техники и технологии. В наиболее концентрированном виде она возникает при сборке некоторых классов машин и механических приборов, где эта задача называется задачей доступа.

Наиболее эффективными, для данной задачи, показали себя метод максимального математического ожидания выигрыша и метод ожидаемого значения дисперсии. Для последнего можно рекомендовать использование коэффициента k в пределах от 1 до 1.5.

В дальнейшем планируется разработать более тонкую модель распределения вероятности допустимости вершины по уровням диаграммы Хассе, рассмотреть эффективность работы алгоритмов на реальных прикладных задачах, выявить зависимость эффективности алгоритма от средней степени вершины в диаграмме Хассе.

Литература

1. Божко А. Н. Моделирование механических связей изделия. Электронное научно-техническое издание Наука и образование [Электронный ресурс] – URL: http://technomag.edu.ru/doc/124312.html (дата обращения: 12.12.2011).

2. Божко А. Н. Игровое моделирование доступа. Электронное научно-техническое издание Наука и образование [Электронный ресурс] – URL: http://technomag.edu.ru/doc/134322.html (дата обращения: 12.12.2011).

3. Черноруцкий И. Г. Ч-49 Методы принятия решений. — СПб.: БХВ-Петербург, 2005.

ГЛОБАЛЬНАЯ ОПТИМИЗАЦИЯ НА ОСНОВЕ ИСКУССТВЕННЫХ ИММУННЫХ СИСТЕМ

Бабичева Н.О.
Научный руководитель: д.т.н., профессор, Карпенко А.П.
МГТУ им. Н.Э.Баумана, Москва, Россия

DEVELOPMENT AND ANALYSIS OPTIMIZATION ALGORITM ON BASIS OF ARTIFICIAL IMMUNE SYSTEM

Babicheva N.O.
Supervisor: Dr., Prof., Karpenko A.P.
MSTU, Moscow, Russia

Аннотация
В статье рассматривается алгоритм глобальной оптимизации на основе искусственных иммунных систем (ИИС). Эффективность работы алгоритма во многом зависит от процесса мутации, поэтому рассмотрено два различных варианта мутации, а также проведена оценка сходимости алгоритма при различных свободных параметрах.

Abstract
The article considers an optimization algorithm based on artificial immune systems (*AIS*). The efficiency of the algorithm depends on the mutation process, so in this paper we consider two different variants of mutation, and an assessment of the convergence of the algorithm under various free parameters.

Введение

Искусственные иммунные системы *(artificial immune system, AIS)* или иммунологические вычисления *(immunological calculations)* представляют собой относительно новую область исследований. Для решения прикладных задач с помощью парадигмы ИИС используют различные модели. При решении задач оптимизации чаще всего применяют модели иммунной сети и клонального отбора. В данной работе рассмотрен алгоритм на основе модели клонального отбора для решения задач глобальной безусловной оптимизации [1].

Постановка задачи и схема алгоритма

Рассматриваем задачу многомерной глобальной безусловной оптимизации

$$\min_{X \in R^N} \Phi(X) = \Phi(X^*) = \Phi^*,$$

где N - размерность вектора варьируемых параметров X, а R^N - N-мерное арифметическое пространство.

Алгоритм клонального отбора разработан на основе теории клонального отбора [2], в соответствии с которой *B*- и *T*-клетки адаптируются для борьбы с чужеродными клетками. Схема алгоритма имеет следующий вид [3].

Случайным образом создаем множество *P* решений или антител, состоящее из клеток памяти *M* и остальных клеток P_r, так что $P = M \bigcap P_r$.

Выбираем *n* лучших антител из множества *P* по их аффинности и обозначаем полученный набор P_n.

Клонируем антитела набора P_n пропорционально их аффинности с коэффициентом клонирования *B*. Полученное временное множество клонов обозначаем *C*.

Применяем операцию мутации к клонам множества *C*, степень которой обратно пропорциональна аффинности клонов. Модифицированное множество *C* обозначаем *C**.

Выбираем лучшие элементы из множества *C** и помещаем их во множество *M*. Также заменяем некоторые элементы множества *P* улучшенными элементами из множества *C**.

Для повышения разнообразия популяции заменим d антител множества P_r оригинальными (N_d), то есть случайно сгенерированными антителами. Вероятность замены антитела полагаем обратно пропорциональной его аффинности.

Общее число антител в популяции (число рассматриваемых векторов X) обозначаем N_{ab} (*number antibodies*). Рекомендованное значение N_{ab} лежит в диапазоне $[10; 100]$. Рекомендуемые значения параметра n принадлежат интервалу $[1 : N_{ab}]$, а параметра B - интервалу $[1; 6]$.

Вычислительный эксперимент

Реализация алгоритма выполнена в системе MATLAB. В качестве тестовых рассмотрены функции Розенброка (одно экстремальная овражная функция), Химмельблау (четырех экстремальная функция) и Растригина (многоэкстремальная функция). Исследование проведено для размерности вектора варьируемых параметров, равной 2, 4, 8, 16.

Сходимость алгоритма для различных тестовых функций иллюстрируют рисунки 1-3, на которых t – номер итерации.

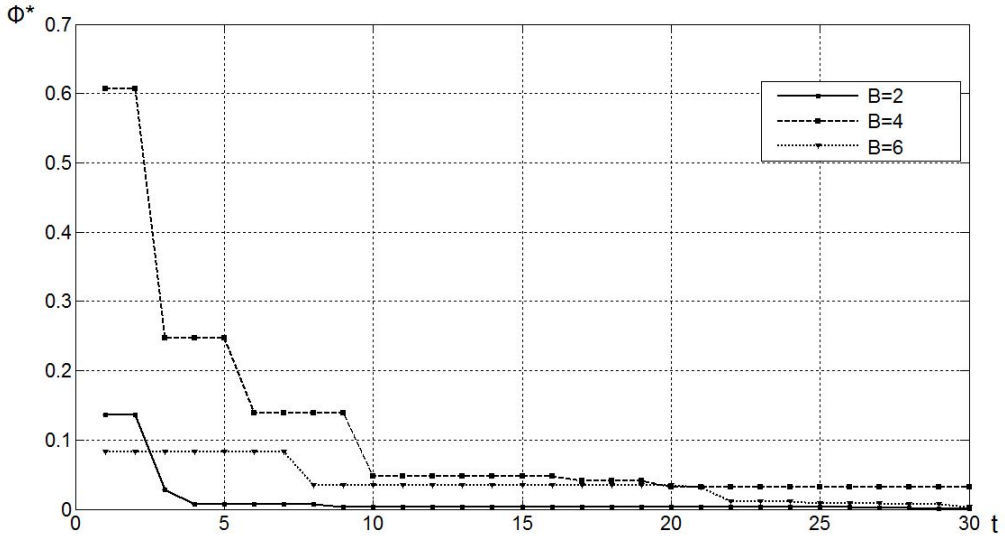


Рисунок 1 – Сходимость алгоритма в функции параметра B: функция Розенброка; N=2

Рисунок 1 показывает, что максимальную скорость сходимости обеспечивает значение коэффициента клонирования B, равное двум. Таким образом, для функции Розенброка оказывается нецелесообразной большая мощность временного множества клонов C.

Из рисунка 2 следует, что, как и следовало ожидать, с увеличением размера популяции для функции Химмельблау скорость сходимости алгоритма повышается. Однако при этом может возрастать суммарное число испытаний (вычислений значений целевой функции).

Рисунок 3 иллюстрирует тот ожидаемый факт, что для функции Растригина увеличение числа лучших антител n повышает скорость сходимости алгоритма. Как и в предыдущем случае при этом может возрастать число испытаний.

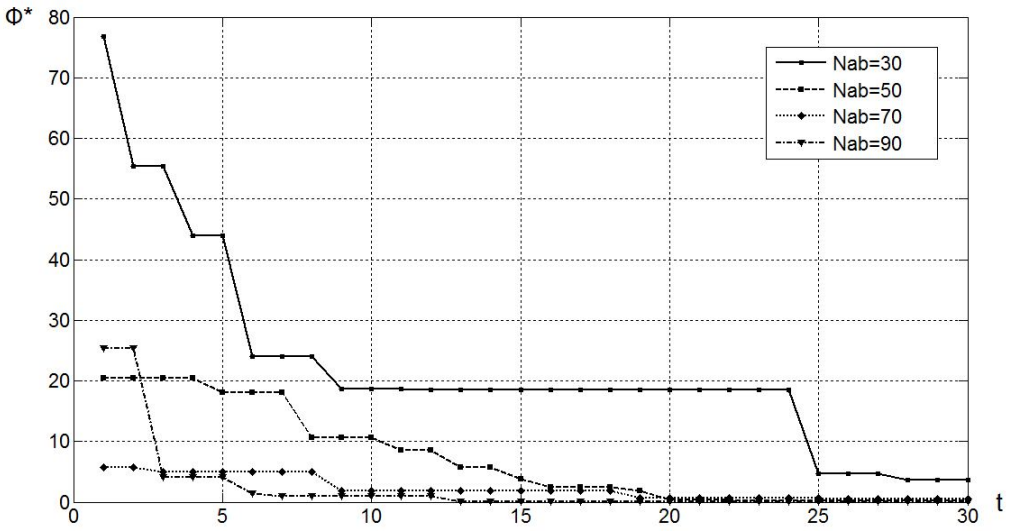


Рисунок 2 – Сходимость алгоритма в функции параметра N_{ab} : функция Химмельблау; N=2

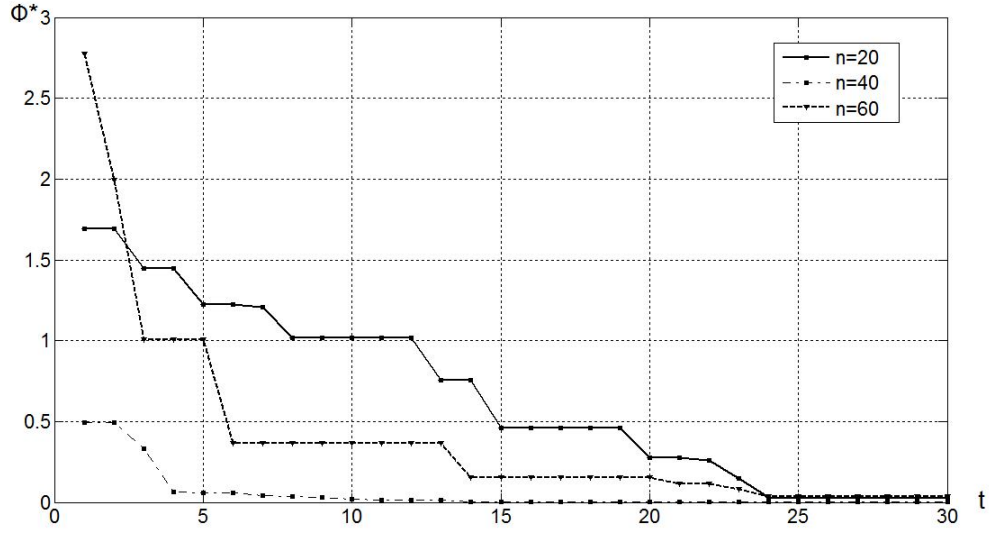


Рисунок 3 – Сходимость алгоритма в функции параметра n: функция Растригина; N=2

Аналогичные выводы относительно значений величин B, N_{ab}, n сделаны относительно других указанных выше тестовых функций.

Результаты исследования сходимости алгоритма выявили необходимость его модификации, направленной на повышение скорости сходимости. В работе предложена такая модификация, заключающаяся в использовании на различных стадиях поиска различной степени мутации.

Некоторые результаты сравнения сходимости исходного и модифицированного алгоритмов представлены на рисунке 4 (функция Химмельблау). Рисунок показывает, что модифицированный алгоритм имеет существенно более высокую скорость сходимости. Остается открытым вопрос об оптимальной стратегии изменения степени мутации в процессе итераций.

В целом, результаты исследования показывают сильную зависимость эффективности алгоритма от значений его свободных параметров (что является характерным для всех популяционных алгоритмов). Поэтому актуальной является задача поиска оптимальных значений этих параметров, т.е. задача метаоптимизации алгоритма ИИС.

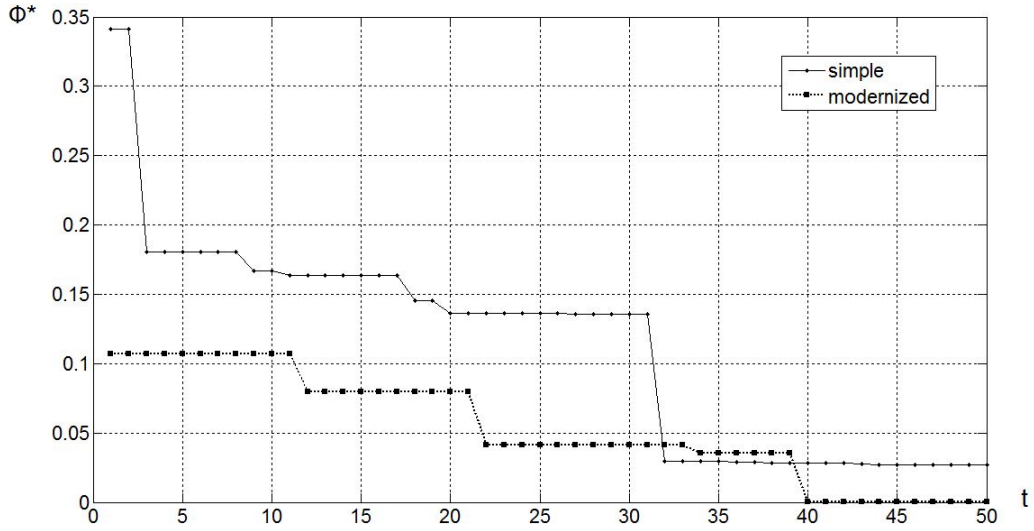


Рисунок 4 - Сравнение базового и модернизированного алгоритмов

В связи с высокой вычислительной сложностью целевых функций во многих практически значимых задачах, актуальными являются разработка и исследование параллельных методов и алгоритмов ИИС.

Заключение

В работе выполнена реализация и исследование эффективности алгоритма глобальной безусловной оптимизации на основе ИИС. Результаты исследования показали высокую эффективность алгоритм при оптимизации сложных овражных и многоэкстремальных функций. Вместе с тем, эти результаты выявили недостаток алгоритма, заключающийся в большом числе требуемых оценок целевой функции.

На этой основе в работе предложена модификация алгоритма, предполагающая использование различных операторов мутации на различных стадиях функционирования алгоритма. Выполнена реализация модифицированного алгоритма и исследование его эффективности. Показано, что модифицированный метод обеспечивает более высокую эффективность при оптимизации, как унимодальных, так и многоэкстремальных тестовых функций, по сравнению с исходным алгоритмом. При этом число оценок целевой функции оказывается приемлемым.

Кроме того, в работе выполнен поиск оптимальных значений свободных параметров алгоритма - общего числа антител в популяции, числа антител подлежащих клонированию, а также коэффициента клонирования.

Литература

1. Gelbukh A., Sidorov G., Lara-Reyes D., Chanona-Hernandez L. Division of Spanish Words into Morphemes with a Genetic Algorithm // Lecture Notes in Computer Science, 2008, V. 5039, pp. 19-26.

2. Nunes de Castro L., Von Zuben F. J. Learning and optimization using the clonal selection principle // IEEE Transactions Evolution Computation, 2002, Vol. 6, No. 3, pp. 239-251.

3. Herrera-Lozada J. C., Calvo H., Taud H. A Micro Artificial Immune System // Polibits, 2011, Vol. 43, pp.107-111.

ОТСЛЕЖИВАНИЕ ПОЛОЖЕНИЯ КАМЕРЫ СИСТЕМЫ КОМПЬЮТЕРНОГО ЗРЕНИЯ В ТРЁХМЕРНОМ ПРОСТРАНСТВЕ В РЕЖИМЕ РЕАЛЬНОГО ВРЕМЕНИ

Афиногенов Е.И.
Научный руководитель: к.т.н, доцент Волосатова Т.М.
Кафедра «САПР» (РК6) МГТУ им. Н.Э.Баумана, Москва, Россия

REAL-TIME 3D CAMERA TRACKING FOR COMPUTER VISION SYSTEM

Afinogenov E.I.
Scientific chief: PhD, associate professor Volosatova T.M.
CAD/CAE Department (RK6) of BMSTU, Moscow, Russia

Аннотация
В данной работе рассматривается задача восстановления трехмерного положения камеры по последовательности видеоизображений. Для решения задачи предлагается метод, основанный на ускоренном поиске соответствий ключевых точек на изображении с последующим восстановлением 3D структуры сцены, используя устойчивую схему факторизации.

Abstract
This article deals with real-time 3D camera tracking. Method is proposed. It is assumed that projects of points are correspondent. Speeded-Up Feature-Accelerated Segment Test (FAST) detector and Singular Value Decomposition method is proposed for solving such problem.

Введение

Компьютерное зрение – одна из самых востребованных областей на современном этапе развития информационных технологий. Оно требуется на производстве, при управлении роботами, при автоматизации процессов, в медицинских и военных приложениях [1].

Одной из важнейших задач в системе компьютерного зрения является задача распознавания объектов. Особенно остро данный вопрос стоит в роботизированных системах, осуществляющих манипуляцию телами в пространстве, когда требуется определить не просто принадлежность объекта к какому-либо классу, а установить его позицию в трехмерной сцене по отношению к роботу и другим телам.

В связи с развитием доступных вычислительных мощностей актуальным становится вопрос создания систем распознавания трехмерных объектов, функционирующих в режиме реального времени. В данной работе рассматривается задача 3D трекинга или, другими словами, отслеживание положения камеры в трехмерном пространстве в режиме реального времени как важнейшая составляющая системы распознавания 3D объектов в динамике.

Постановка задачи

Пусть имеется некоторая мобильная роботизированная установка, снабженная видеокамерой, обеспечивающей зрение системы. Если не учитывать ограничения, связанные с конкретной системой, то камера обладает 6-ю степенями свободы. Необходимо определить положение камеры при её движении в пространстве относительно стационарных объектов сцены (в системе координат, связанной с неподвижными объектами сцены). Система не предполагает использование каких-либо маркеров (зафиксированных датчиков положения), опираясь главным образом на пассивное зрение [1]. В данной работе не рассматривается случай наличия в сцене отдельных движущихся объектов.

На следующем рисунке представлена упрощенная схема работы предлагаемого 3D трекера, отслеживающего положение камеры в пространстве на основе видеопоследовательности, поступающей на вход системы.

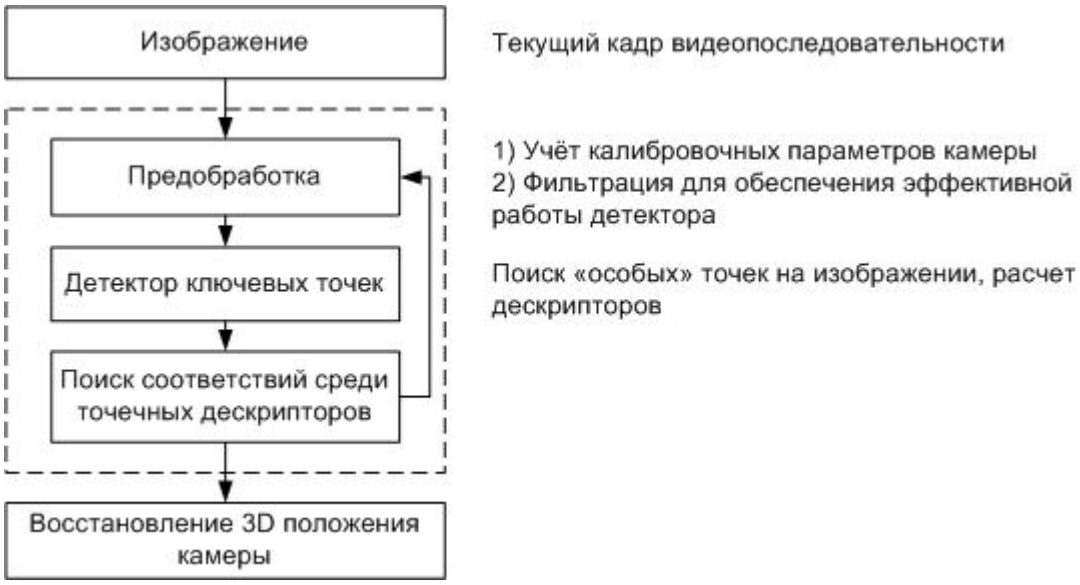


Рисунок 1 – Обобщенный алгоритм «компьютерного зрения»

Поступающее на вход системы изображение предварительно обрабатывается: переводится в градацию серого, нормализуются интенсивности пикселей. Это необходимо для того, чтобы полученные далее дескрипторы соответствующих точек имели высокую степень схожести независимо от изменения освещенности сцены.

Для обеспечения устойчивой работы системы при наличии смаза, появляющегося в результате движения камеры, используются пирамиды изображений с масштабным коэффициентом 2. Поиск особенностей осуществляется на каждом уровне пирамиды.

Далее изображение поступает на вход детектора ключевых точек. В настоящее время разработано огромное количество методов поиска особенностей. Наиболее известными являются уголковый детектор Харриса [4], SIFT (Scale-Invariant Feature Transform) [5], SURF (Speed-Up Robust Features) [6], Ferns [7]. Однако данные детекторы имеют высокую вычислительную сложность и не могут быть эффективно использованы для решения поставленной задачи.

Наиболее применимым в данном случае видится детектор FAST (Feature-Accelerated Segment Test) [8], осуществляющий поиск ключевых точек изображения. Он классифицирует точку p (рис. 2) как «уголок», если существует ряд смежных пикселей на окружности, интенсивности которых больше $I_p + t$ или интенсивности всех меньше $I_p - t$, где I_p – интенсивность точки p, а t – пороговая величина.

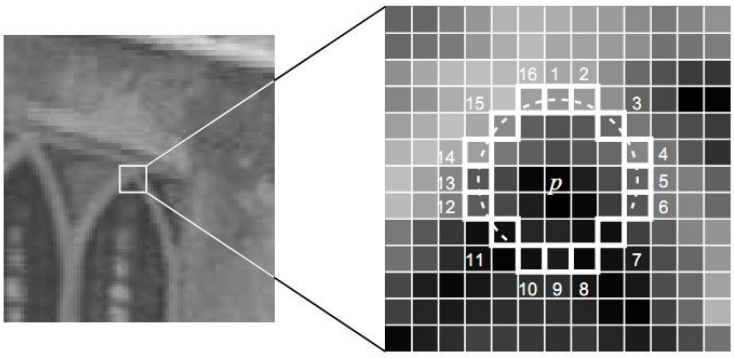


Рисунок 2 – Поиск ключевых точек

Далее для найденных на текущем кадре видеопоследовательности «особых» точек осуществляется поиск соответствующих точек на предыдущем кадре путем сравнения их дескрипторов. FAST-дескриптор представляет собой нормализованную гистограмму интенсивностей для 64 пикселей, взятых разреженно вокруг целевой точки (рис. 3а).

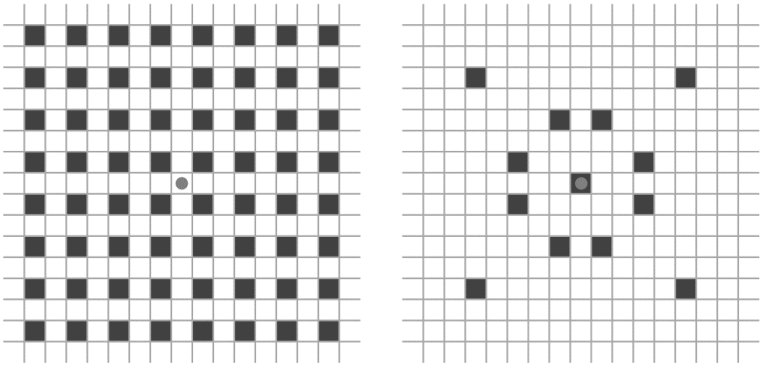


Рисунок 3 - Пиксели, используемые в дескрипторе FAST (а);
Пиксели, используемые для формирования индекса поиска (б).

Дескриптор ключевой точки можно записать следующим образом:

$$\begin{matrix} D_{0,0} & D_{0,1} & D_{0,2} & D_{0,3} & D_{0,4} \\ D_{1,0} & D_{1,1} & D_{1,2} & D_{1,3} & D_{1,4} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ D_{63,0} & D_{63,1} & D_{63,2} & D_{63,3} & D_{63,4} \end{matrix}$$

Здесь D_i – нормализованная гистограмма для точки i (i = 0…63), занимающая всего 5 бит ($D_{i,0}...D_{i,4}$). Таким образом, чтобы определить степень схожести двух ключевых точек, необходимо вычислить количество выставленных бит в 64-разрядном числе, получающемся путем побитового умножения соответствующих столбцов дескрипторов с последующим побитовым сложением.

Для того, чтобы ускорить поиск соответствий, применяется механизм индексации, при котором для каждой точки рассчитывается индекс поиска на основе интенсивностей 13 пикселей (рис. 3б), причем значение каждого пикселя кодируется одним битом. Таким образом, таблица индексов будет иметь размер 8192.

После определения достаточного количества точек в сцене, повторяющихся в нескольких последовательных кадрах, осуществляется восстановление трёхмерного положения камеры. Для решения данной задачи используется устойчивая схема факторизации на основе разложения по сингулярным значениям (SVD – Singular Value Decomposition) [2, 3].

Заключение

В настоящее время существует множество систем, позволяющих по видеоряду восстанавливать трехмерное положение камеры [9]. Данные системы получили широкое распространение в кинематографии для создания различных спецэффектов. Однако они не предусматривают использования в режиме реального времени, часто требуют наличия в сцене определенных маркеров.

В данной работе рассматривается 3D трекер, вычисляющий положение камеры в пространстве в режиме реального времени. Предлагаемая схема определения ключевых точек на изображении с ускоренным поиском соответствий занимает незначительное процессорное время, позволяя использовать совместно с системой распознавания трёхмерных объектов в динамике.

Литература

1. Форсайт Д., Понс Ж. Компьютерное зрение. Современный подход. : Пер. с англ. – М.: Издательский дом «Вильямс», 2004. – 928 с.
2. Tomasi C., Kanade T. Shape and Motion from Image Streams: a Factorization Method Full Report on the Orthographic Case, 1992
3. Афиногенов Е.И. Восстановление аффинной структуры сцены по движению.-М.: МГТУ им. Н.Э.Баумана. Тезисы доклада 13-й ММ НТК «Наукоемкие технологии и интеллектуальные системы», 2011, с.125-128.
4. C. Harris and M.J. Stephens. A combined corner and edge detector. In Alvey Vision Conference, pages 147-152, 1988.
5. David G. Lowe. Object recognition from local scale-invariant features. International Conference on Computer Vision, pages 1150-1157, 1999.
6. H. Bay, A. Ess, T. Tuytelaars, Luc Van Gool “SURF: Speeded Up Robust Features”, Computer Vision and Image Understanding, Vol. 110, No. 3, pp. 346-359, 2008.
7. V. Lepetit and P. Fua. Keypoint recognition using randomized trees. IEEE Transactions on Pattern Analysis and Machine Intelligence, 28(9): 1465-1479, Sept. 2006.
8. E. Rosten, R. Porter, T. Drummond. Faster and better: a machine learning approach to corner detection. 2008.
9. S. Mirpour. A Comparison of 3D Camera Tracking Software. 2008.

РАЗРАБОТКА ПО АРМ ОБРАБОТКИ 3D ИЗОБРАЖЕНИЙ В СИСТЕМАХ ТЕХНИЧЕСКОГО ЗРЕНИЯ: РАЗРАБОТКА ПРОГРАММНОГО СРЕДСТВА ДЛЯ РЕАЛИЗАЦИИ ГОЛОГРАФИЧЕСКОГО МЕТОДА ВИЗУАЛИЗАЦИИ 3D ДИНАМИЧЕСКОЙ СЦЕНЫ В СРЕДЕ MATHLAB

Криволапова А.С.
К.т.н., доцент РК6 Волосатова Т.М.
Кафедра САПР (РК6), МГТУ им. Н.Э. Баумана, г.Москва, Россия

THE SOFTWARE DEVELOPMENT IN TECHNICAL VISION'S SYSTEMS FOR THE AUTOMATED WORKPLACE: THE SOFTWARE DEVELOPMENT FOR REALIZATION OF HOLOGRAPHIC VISUALISATION METHOD FOR 3D DYNAMIC SCENE IN MATHLAB

Krivolapova A.S.
Ph.D., assistant professor RK6 Volosatova T.M.
CAD/CAE/PLM/PDM Department (RK6), Bauman Moscow State Technical University, Moscow, Russia

Аннотация
В работе рассмотрена задача синтеза Фурье-голограммы изображения и определения его графика фаз с целью последующего их анализа в рамках других проектов. Проведено описание программного средства, разработанного на основе математических функций пакета MathLab, разработан понятный пользователю интерфейс продукта, а также предоставлен наглядный пример работы данного программного модуля.

Abstract
This article deals with a task of creating a Furier-hologram of the image and finding its plot of phases for future analyses in other projects. The software, developed on the base of mathematical bases of MathLab, was described and user friendly interface of the product was created. The software working results was also represented.

Введение

На сегодняшний день построение голограмм не потеряло своей актуальности, тем более что голограммы, синтезированные на компьютере, могут выполнять функции, аналогичные оптически построенным голограммам, но спектр применения цифровой голографии намного шире. С их помощью можно распознавать текст и изображение, исправлять их качество и проводить интерферометрические испытания оптических поверхностей.

Математическое представление

Голограмма строится на основе математического представления объекта как некого облака точек, информация о которых известна. Таким образом, для синтеза Фурье-голограммы необходимо произвести большое количество отсчетов, каждый из которых состоит из амплитуды и фазы.

Представим функцию отсчетов $a_s(x)$: $a_s(x) = a(x)\sum_{m=-\infty}^{\infty}\delta(x - m\Delta x)$, где каждый отсчет функции a(x) будет определяться $\delta - функцией$, сила которой зависит от величины функции a(x) в соответствующей точке. Графически это представлено на рис.1 (длины стрелок пропорциональны силам).

Если отсчеты были взяты с соблюдением теоремы отсчетов, то точный Фурье-образ непрерывной функции a(x) может быть получен из Фурье-образа отсчетной функции: $A_s(\xi) = A(\xi) * comb(\Delta x\xi)$ Используя определение оператора свертки и свойства

фильтрации и симметрии $\delta - функции$, получаем: $A_s(\xi) = \frac{1}{\Delta x}\sum_{m=-\infty}^{\infty} A(\xi - \frac{m}{\Delta x})$. Перекрытия сдвинутых образов не происходит, если $\Delta x \leq \frac{1}{\xi_{max}}$.

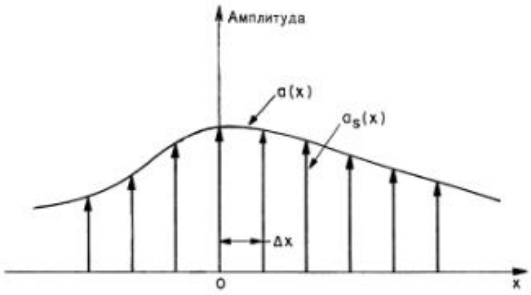


Рисунок 1 - Представление функции отсчетов

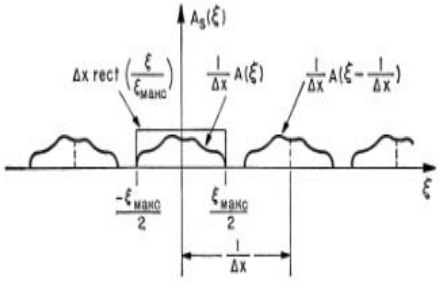


Рисунок 2 - Фурье-образ функции отсчетов

Таким образом, $A(\xi) = A_S(\xi)\Delta x\, rect(\frac{\xi}{\xi_{max}})$. Фурье-образ функции отсчетов, состоящей из бесконечного ряда функций, пропорциональных исходному Фурье-образу и отстающих на расстояние $\frac{1}{\Delta x}$, представлен на рис.2.

Для построение обратного Фурье-образа и, следовательно, восстановление a(x) чередой аналогичных преобразований получаем выражение: $a(x) = \sum_{m=-\infty}^{\infty} a(\frac{m}{\xi_{max}})\frac{\sin(\pi\xi_{max}x - \pi m)}{\pi\xi_{max}x - \pi m}$

Чтобы распространить теорему отсчетов на двумерное пространство, следует использовать двумерную матрицу $\delta - функций$.

Для построения Фурье-образа в работе используется Быстрое Преобразование Фурье, как алгоритм, существенно сокращающий время вычисления. Он реализован с помощью математичеких средств программы MathLab.

Физическое представление

Прежде чем обратиться непосредственно к программному продукту, стоит коротко объяснить существование голограмм с точки зрения физики: Имеющийся объект освещается основным пучком света и опорным пучком V_{on} когеретного света, идущего от лазера. Опорный пучок направлен таким образом, чтобы он пересекся со светом, прошедшим через объект или отраженным от него. Так как пучки света когерентны, то интерференционная картина образуется во всем объеме, в котором перекрываются пучки. Светочувствительная среда, находящаяся в области перекрытия подверглась определенным изменениям, и после окончания экспозиции и соответствующей обработки этих изменений в вариации оптического пропускания, позволяет получить голограмму. Наглядно это продемонстрировано на рис.3.

Прямое и мнимое изображение объекта может быть получено в случае повторного освещения изображения исходным опорным пучком, часть дифрагированного на ней света вновь воссоздает волновой фронт, который при регистрации голограммы шел от объекта. Восстановленная волна исходит из голограммы точно так же, как и первоначальная предметная волна. Наблюдатель воспринимает ее как бы исходящей от мнимого изображения, которое находится там, где ранее находился предмет. Наглядно эта часть эксперимента представлена на рис.4

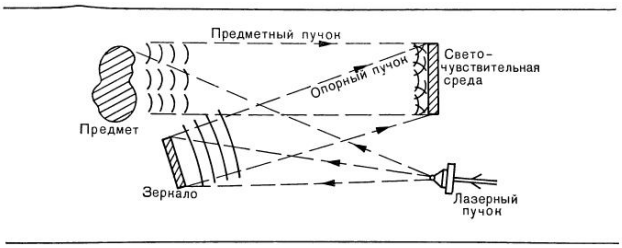


Рисунок 3 - Получение голограммы

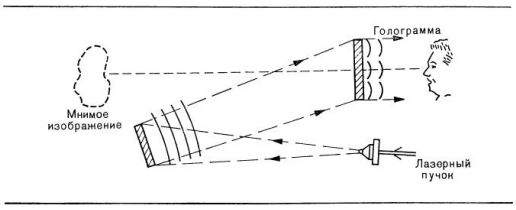


Рисунок 4 - Получение прямого и мнимого изображения

Если обратить опорный пучок так, что все лучи обращенного пучка будут направлены противоположно лучам исходного опорного пучка, то такой сопряженный пучок, освещаюший обратную сторону голограммы, создаст действительное изображение в месте первоначального размещения предмета. Восстановление изображения отражено на рис.5.

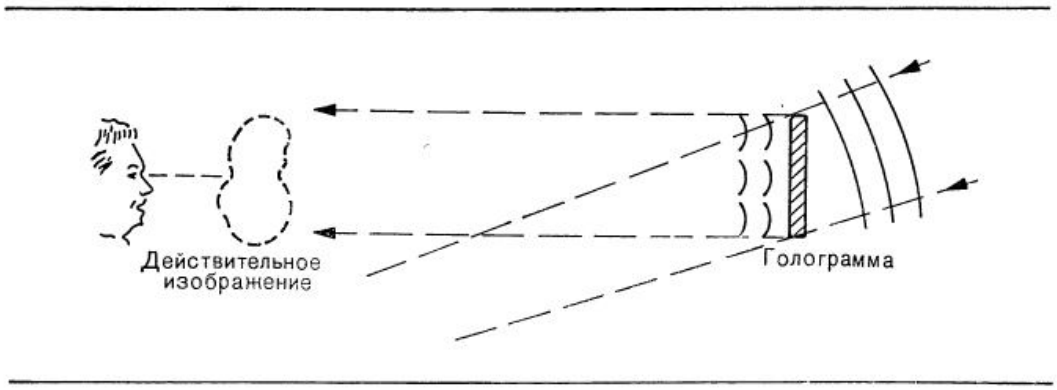


Рисунок 5 - Восстановление изображения

Программная реализация

Программный модуль основан на базе пакета MathLab, который предоставляет широкий спектр функциональных возможностей для обработки изображений. Первым этапом работы является загрузка и подготовка изображения или видео для последующей обработки. Затем производится прямое быстрое преобразование Фурье для матрицы объекта с помощью функции fft2(x) и с помощью освещения опорным пучком света получается голографическое изображение. Для полученной голограммы определяется интенсивность, а затем она повторно освещается опорным пучком, вследствие чего появляется возможность восстановить изображение, используя функцию обратного БПФ ifft2(x). В работе также важную роль играет функция ifftshift(x), позволяющая раздвигать голограммы. Конечным этапом является определение с помощью стандартных функций действительной и мнимой часть изображения для нахождения фазы $\varphi = arctg(\frac{Image}{Real})$

Программный модуль позволяет работать с изображениями типов bmp и jpg, а также с видео формата avi. Реализация оснащена понятным пользователю интерфейсом и реакцией на ошибочные действия. Также разработано руководство пользователя.

1.Изображения типов bmp и jpg

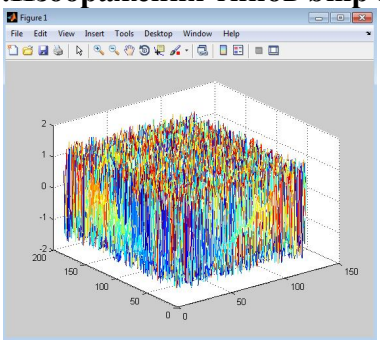


Рисунок 6 - График фаз

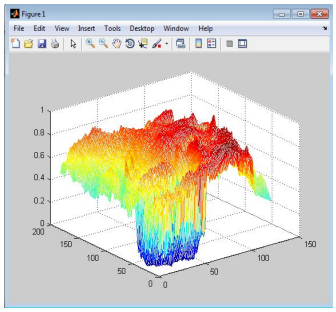


Рисунок 8 - Фурье-голограмма вид сбоку

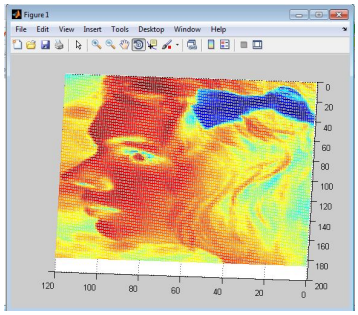


Рисунок 9 - Фурье-голограмма вид сверху

2.Кадры, полученные из видео формата avi

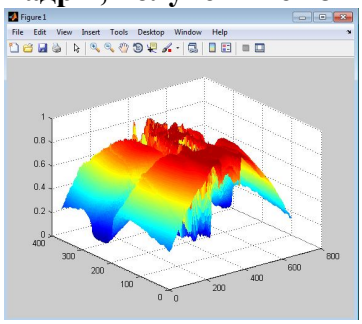


Рисунок 10 - Фурье-голограмма для 30-ого кадра

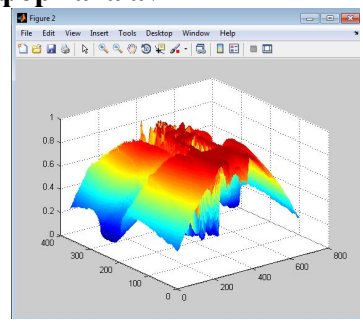


Рисунок 11 - Фурье-голограмма для 31-ого кадра

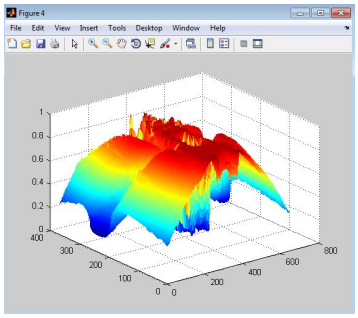


Рисунок 12 - Фурье-голограмма для 32-ого кадра

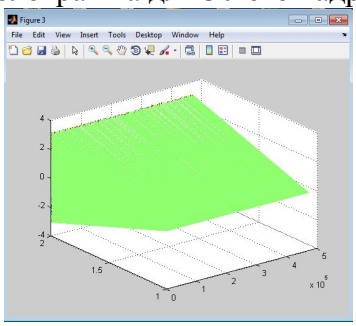


Рисунок 13 - График разности фаз для кадров 30-го кадра

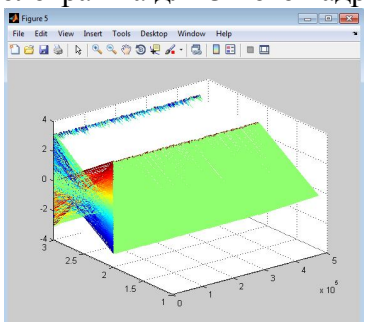


Рисунок 14 - График разности фаз для кадров 31-го кадра

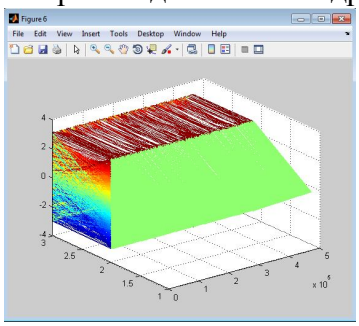


Рисунок 15 - График разности фаз для кадров 32-го кадра

Литература

1. Р.Кольер, К.Беркхарт, Л.Лин «Оптическая голография», под ред.Островского, «Мир», М., 1973г.
2. П.И. Рудаков, И.В. Сафонов «Обработка сигналов и изображений.MathLab 5.x», под ред. Потемкина, «Диалог-МИФИ»,М.,2000г.
3. А.Б. Сергиенко «Цифровая обработка сигналов», «Питер», СпБ, 2002г.
4. http://matlab.exponenta.ru/

МОДУЛЬНАЯ ИЕРАРХИЧЕСКАЯ СТРУКТУРА ОРГАНИЗАЦИИ ОТДЕЛА РАЗРАБОТКИ ПО

Новиков П.В.
Научный руководитель: к.т.н., доцент , Власов А.И.
МГТУ им. Н.Э. Баумана, Москва, Россия

PRINCIPLES OF CONSTRUCTION INTELLECTUAL SYSTEM ЩА ANALYSIS AND OPTIMIZATION LOAD OF SECOR AIRSPACE

Novikov P.V.
Supervisor: Ph.D., associate professor, Vlasov A.I.
MSTU, Moscow, Russia

Аннотация
В статье рассматриваются основные обобщенная структура построения отдела разработки программного обеспечения. При помощи данной структуры возможно организовать разработку ПО для любой цели разработки. Основными преимуществами данной структуры являются модульность представленной структуры, а так же взаимозаменяемость и расширяемость любого компонента системы.
Abstract
The article discusses the basic construction of the generalized structure of software development. With this structure can be organized to develop software for any purpose of development. The main advantages of this structure are presented modular structure, as well as interoperability and scalability of any system component.

Введение

В нынешнее время все компании – от малого до крупного размера нуждаются в информационных технологиях для того, чтобы выдерживать конкуренцию в условиях повсеместного использования высокотехнологичных решений. Однако специфика средств IT и методов их внедрения, эксплуатации и сопровождения в зависимости от масштаба предприятия может существенно отличаться. Если необходимый набор средств IT в случае с фирмой с численностью штата до 50 человек может исчерпываться только одной автоматизированной системой учета и одним системным администратором средней квалификации, то крупной организации необходим комплексный подход к созданию IT инфраструктуры, который включает в себя создание IT отдела [1].

Основные понятия создания отдела разработки ПО

Рассмотрение вопроса создания отдела разработки ПО в крупной фирме с несколькими проектами приводит к выводу, что отдел IT должен быть системой, а не просто набором средств и компанией дружно работающих специалистов из разных областей. Таким образом, отдел ИТ компании – это совокупность взаимодействующих средств IT и специалистов в области IT, целями которой являются [1]:

- Обеспечение информационными технологиями;
- Повышение эффективности деятельности компании посредством оптимизации информационных потоков.
- IT отдел компании выполняет задачи, такие, как:
- Реализация IT проектов;
- Обеспечение работоспособности информационных систем;
- Предоставление руководящему звену компании сведений о новых возможностях IT и технологиях управления ими;
- Делопроизводство отдела, ведение бюджета IT, учет IT активов, обеспечение кадрового состава IT.

Перечислим вкратце четыре основных категории симптомов, после чего перейдем к подробному их рассмотрению [2]:

1. Удовлетворенность. Речь пойдет об удовлетворенности от работы ИТ-отделов и уверенности в качестве их услуг.
2. Финансирование и затраты. Стоимость ИТ-ресурсов, услуг и капитальные затраты.
3. Проекты. Информационная поддержка бизнес-процессов компании, внедрение программного и аппаратного обеспечения для работы компании.
4. Штат. Управление сотрудниками, дух коллектива и организация ИТ-службы.

Основные принципы построения отдела разработки ПО

Основной принцип, заложенный в структуру – у каждого начальника не более 5–7 подчиненных. Каждый начальник в первую очередь занимается организационной работой непосредственно подчиненного персонала и во вторую выполнением своих остальных обязанностей.

Каждый подчиненный выполняет только задачи, поставленные ему начальником, и занимается только работой в своей области, тем самым углубляя свою квалификацию в данной области. Основное следствие из этого, заключается в том, что высококвалифицированный персонал справляется с задачей в короткие сроки и опускает меньшее число ошибок в сравнении с малоопытными сотрудниками, что в свою очередь сказывается на качестве выходного программного обеспечения.

Так же немаловажным аспектом является **модульность, расширяемость и древовидность** структуры. Что позволит ускорить реорганизацию данной структуры для различных задач, повысить надежность разрабатываемых решений, ускорить и распараллелить разработку ПО, обеспечить возможность смены кадрового состава с наименьшими потерями. Согласно схеме представленной на рисунке 1, каждый приведенный сегмент может являться отдельным модулем и может быть взаимозаменяем аналогичным. Расширяемость предлагаемой системы достигается путем Создания древовидной структуры подчинения и основополагающий заложенный принцип, описанный выше - у каждого начальника не более 5–7 подчиненных. Древовидность структуры заключается в подчинении одному начальнику 5 подчиненных, в то же время каждый подчиненный может являться начальником для еще 5 подчиненных.

Таблица 1 – Описание рабочих и их обязанностей структуры ИТ отдела

Графическое обозначение	Обязанности и Требования
Руководитель проектов	**Обязанности**: Курирует работу всех начальников, проверяет отчетность каждого начальника, отвечает за набор персонала и его распределением. Так же одной из основных задач является **обучение персонала**, коммуникация с заказчиком. **Требование**: глубокое знание всей технологической цепочки разработки ПО, глубокое знание предметной области разрабатываемых проектов, возможность управления персоналом и постановки четких задач.
Старший аналитик	**Обязанности**: Распределение задач подчиненным аналитикам, изучение предметной области задачи, проработка логики работы программы, построение логических диаграмм работы ПО и логических моделей структуры БД **Требования**: глубокое знание предметной области, опыт работы с документацией, умение разработки логических моделей UML
Аналитик	**Обязанности**: разбор ТЗ и его корректировка, построение логических диаграмм работы ПО и логических моделей структуры БД, на основе которых в дальнейшем будут построены структурные диаграммы работы ПО и физические модели БД **Требования**: знание предметной области, умение разработки логических моделей UML
Старший архитектор	**Обязанности**: распределение задач между подчиненными, принятие решение об архитектуре ПО, аппаратной архитектуре. Построение структурных диаграмм работы ПО и физической структуры БД **Требования**: глубокое знание платформы для разработки, большой опыт разработки ПО
Архитектор ПО	**Обязанности**: построение структурных диаграмм работы ПО, приближенных к реализации программного обеспечения, и физической структуры БД. **Требования**: опыт разработки ПО, умение разработки и чтения UML диаграмм

Старший тестировщик	**Обязанности**: распределение задач между подчиненными, разработка методик тестирования, руководства оператора, проверка работы подчиненных. **Требования**: знание технологического процесса работы готового проекта, умение разбирать логические UML модели.
Тестировщик	**Обязанности**: разработка методик тестирования, руководства оператора, проверка работы подчиненных. **Требования**: знание технологического процесса работы готового проекта, умение разбирать логические UML модели, внимательность, отсутствие удачи.
Старший разработчик	**Обязанности**: распределение и постановка четких задач подчиненным, разработка ПО, поверхностная проверка разработанного ПО. **Требования**: глубокое знание платформы для разработки, большой опыт разработки ПО, понимание UML моделей, знание предметной области проекта
Программист	**Обязанности**: разработка ПО, поверхностная проверка разработанного ПО, оформление и комментирование исходного кода. **Требования**: знание платформы для разработки, понимание UML моделей
Совещание по разбору результатов тестирования	**Обязанности**: принятие решений о выпуске, либо о доработке ПО. Распределение дефектов по степени сложности между разработчиками и аналитиками. **Требования к составу**: совещание в составе старших аналитика, разработчика, тестировщика, архитектора и руководителя проекта.
Системный администратор	**Обязанности**: развертывание ПО для разработки и работы разработанного ПО, оптимизация работы БД, поддержание в состоянии готовности аппаратного обеспечения и его развертывание, оформление руководства администратора **Требования**: большой опыт администрирования, оформление документации
Заказчик	**Обязанности**: постановка задачи, уточнение постановки задачи, прием готового ПО. **Требования**: –

Таким образом, выше описан персонал, входящий в одну ячейку разработки. В таблице описаны объекты, являющиеся результатом деятельности или поступающие на вход персоналу. Так же одной из главных составляющих данной системы является полная отчетность каждого перед руководящим органом, а так же ведение полного документирования своей работы.

Таблица 2 – Описание объектов структуры ИТ отдела

Графическое обозначение	Описание
Проект (ТЗ на разработку)	Техническое задание для разработки программного обеспечения
Набор логических схем описание функционала	Логические модели в составе: ● Диаграмма деятельности; ●Диаграммы прецедентов; ● Диаграммы состояний; ● Диаграммы взаимодействия; ● Логическая модель БД
Набор документов, описывающих архитектуру системы	Структурные модели в составе: ● Диаграмма классов ● Диаграмма компонентов ● Диаграмма кооперации ● Диаграмма развёртывания ● Диаграмма объектов ● Диаграмма пакетов ● Диаграмма профилей
Методики тестирования	Описание последовательности действий по проведению тестирования. А так же руководство оператора, представляющее собой ограниченную версию методик тестирование.
Результаты тестирования	Описание дефектов выявленных в ходе тестирование и занесенных в общую систему учета.
Разработанное ПО	Готовый программный продукт не прошедший тестирования на данном этапе
Готовый программный продукт	Готовый программный продукт, одобренный к выпуску на совещании

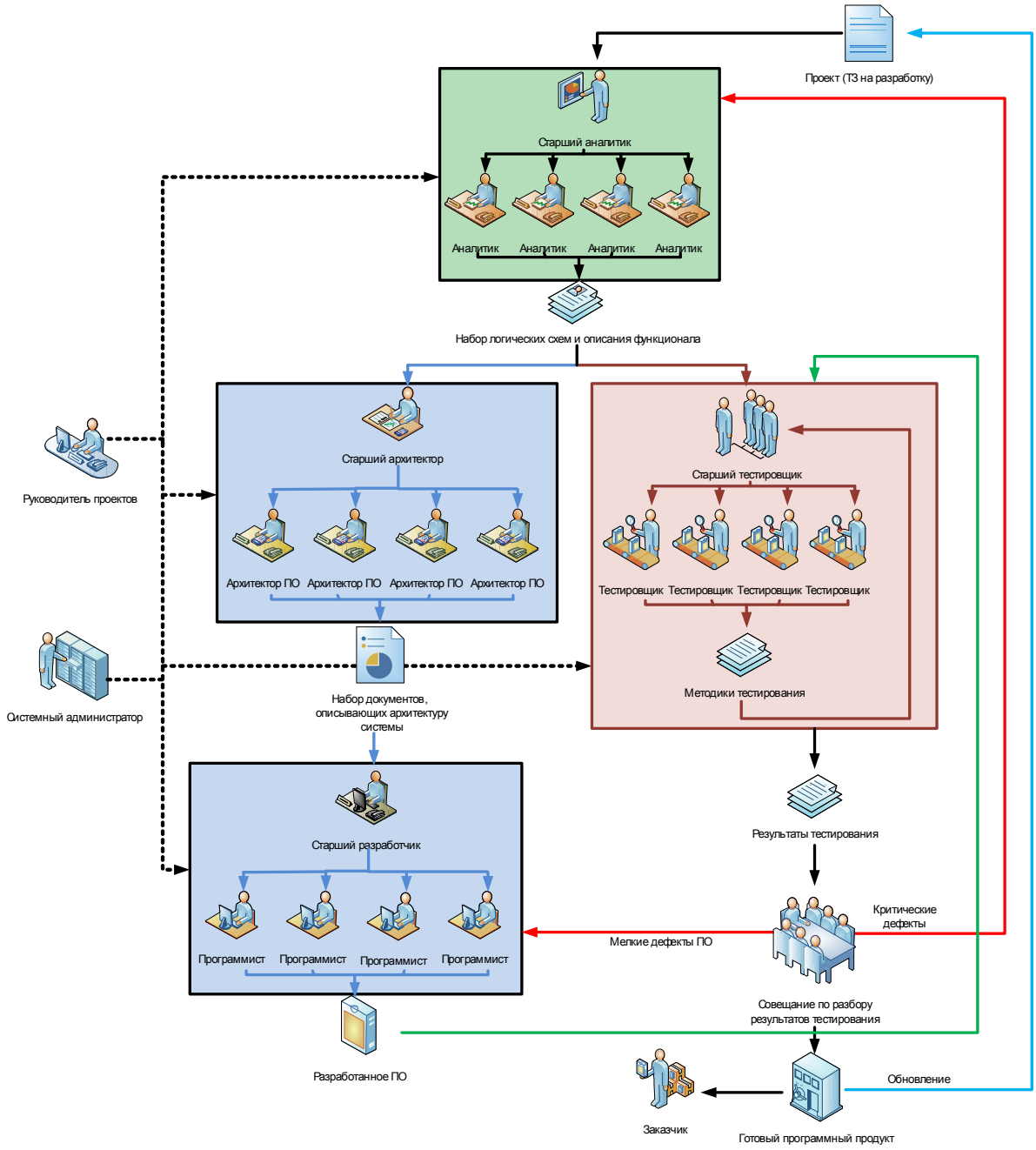


Рисунок 1 – Предлагаемая структура отдела разработки ПО

Заключение

Внедрение представленной в данной работе структуры отдела разработки ПО позволит повысить эффективность работы отдела, а так же позолит реализоавать взаимозаменяемость кадрового состава с наименьшими потерями. Так же к достоинствам данной системы следует отнести высокую формализованность большинства процессов. Недостатками данной структуры являются невыгодность применения к небольшим проектам и маленькому штату персонала.

Литература

4. «IT компании России», IT отдел и его структура, – Электронный ресурс. Режим доступа: http://www.itcompanies.ru/03itotdel.html - Проверено 29.02.2012.
5. Галкин Г., Эффективный ИТ-отдел, – Электронный ресурс. Режим доступа: http://www.iemag.ru/master-class/detail.php?ID=15710 - Проверено 29.02.2012..

АНАЛИЗ БЕЗОПАСНОСТИ PHP

Подорин А.А
Научный руководитель: асс. Аверьянихин А.Е.
МГТУ им. Н.Э. Баумана, кафедра ИУ4, Москва, Россия

THE ANALYSIS OF SAFETY PHP

Podorin A.A
Research advisor: ass. Averjanihin A.E.
Bauman University, IU4 department, Moscow, Russia

Аннотация
Одним популярных языков для программирования является PHP. Его популярность обусловлена рядом достоинств: простота языка, расширяемость, кроссплатформенность и многие другие. Но почти перед каждым программистом встает проблема безопасности в PHP. Безопасность – ключ к стабильной, длительной и надежной работе.

Abstract
One of popular languages for web programming is PHP. Its popularity is caused by a number of advantages: simplicity of language, expansibility, cross-platform and many other things. But before almost each programmer there is a safety problem in PHP. Safety - a key to stable, long and reliable work.

Введение

Одной из основных проблем современных портальных решений становятся проблемы с безопасностью. На фоне стремительного расширения онлайн-платежей и онлайн-магазинов это становится значительной проблемой, потому что большинство начинающих интернет-магазинов (да и обычных сайтов) обычно не озадачиваются обеспечением безопасности: «Работает – слава богу!». Расплата приходит очень быстро, особенно для популярных сайтов. Скачанная база данных, удаленные таблицы, исходные коды сайтов, даже обрушенный веб-сервер – лишь малый список того, что может сделать хакер средней руки, если программист пренебрежет элементарными правилами безопасности. А ведь в большинстве случаев достаточно соблюсти всего несколько элементарных правил, которые заставят хакера отложить готовые программы для взлома и подключиться самому к взлому, а может быть и совсем отказаться от попыток взлома.

Приведенные ниже правила не абсолютны, потому что каждый день находятся новые уязвимости в программном коде, разрабатываются новые способы взлома, но при этом остаются и старые способы взлома, которые невозможно устранить без содействия программиста. Одни из таких способов является SQL-injection. Если в сайте оставлена уязвимость такого рода, то он взламывается в полностью автоматическом режиме и вся информация базы данных становится доступна хакеру примерно через 2-3 часа, как максимум 2-3 дня. Среди этой информации могут быть номера банковских карт, пароли и другая секретная информация.

SQL-injection

Термин SQL-injection(SQL-инъекция) подразумевает под собой внедрение постороннего (взламывающего) SQL-запроса в исходный (взламываемый) SQL-запрос. Основными инструментами SQL-инъекций являются экранирование (“\”) и комментарии (“—“ или “/* */”). Большинство не поверит, что с помощью двух этих безобидных операторов можно уничтожить их базу данных, поэтому приведу простейший пример.

Скрипт сайта осуществляете запрос на получение информации по какому-то пользователю. В PHP это выглядит примерно так:

```
mysql_query(“SELECT * FROM `users` WHERE `username`='$username';”);
```

На первый взгляд ничего страшного, но какова будет расплата, если в переменную $username передать значение “test’;DROP TABLE `users`;--”.

```
$username= “test’;DROP TABLE `users`;--”;
```

В итоге база данных получает такой запрос:

```
SELECT * FROM `users` WHERE `username`='test';DROP TABLE `users`;--';
```

Как последствия полная потеря данных о пользователях. И это самое безобидное, что можно было сделать.

Решение данного случая абсолютно банально. Необходимо всего лишь проверить, чтобы имя пользователя содержало только допустимые символы (например: латиница, цифры, подчеркивание).

Комплексные меры защиты

Для решения любых проблем безопасности нужно придерживаться двух простых правил:

1. Запрещено все, что не разрешено
2. Никакая защита не является абсолютной

Что же эти правила под собой подразумевают? Первое в нашем случае можно расшифровать так: программе (веб-скрипту) должно быть выдано столько прав, которые нужны для того, чтобы он нормально работал. Вернемся к предыдущему примеру.

Мы получили итоговый запрос:

```
SELECT * FROM `users` WHERE `username`='test';DROP TABLE `users`;--;
```

Однако логика PHP подразумевает изначальное подключение к базе данных:

```
mysql_connect('127.0.0.1','test','test');
```

Таким образом, на сервер базы данных (в нашем случае MySQL) мы будем находиться от имени пользователя test и для предотвращения предыдущей атаки достаточно пользователю test запретить удалять таблицы (MySQL, как и большинство крупных баз данных, предоставляет такую возможность).

Теперь обратимся ко второму правилу. Оно подразумевает под собой ровно то, что написано. Каждый метод защиты – это кирпичик в стене на пути хакера и задача любого программиста уложить как можно больше таких кирпичиков, не ухудшая своей программы.

Способы защиты от SQL-инъекций

Руководствуясь предыдущими правилами, я могу выделить два основных способа защиты от SQL-инъекций.

1. Полный контроль входных параметров
2. Разрешение только необходимых SQL-операторов на сервере базы данных
3. Передача шифрованных данных
4. Необратимое шифрование важных данных
5. Сохранение данных об активности системы

Контроль входных параметров

Сразу же встает правомерный вопрос: Почему **полный** контроль входных параметров? Разве не достаточно проверить только те, которые участвуют в формировании запроса для базы данных? Ответ достаточно прост и является продолжением сразу обоих

правил. Зачем давать хакеру лишнюю свободу? Кроме атаки SQL-injection существует множество других, которые могут даже модифицировать код PHP-скрипта, поэтому лучше проверить все. Способов реализации данного способа существует огромное множество от простой поэлементной проверки до проверки четкого соответствия пользовательским типам данных. Например, если входные данные должны быть только строками содержащие латинские буквы и цифры:

```
foreach ($_POST as $key=>$value){
  if(!preg_match("/^[a-zA-Z0-9]+$/i",$value){//Проверка содержаться ли в строке иные символы кроме допустимых
    unset($_POST[$key]);//Удаление значения содержащего неформатные данные
  }
}
foreach ($_GET as $key=>$value){
  if(!preg_match("/^[a-zA-Z0-9]+$/i",$value){
    unset($_GET[$key]);
  } }
```

Данную систему можно свободно расширять. Например, можно сделать отдельные проверки для поля e-mail, цифр хэшей и т.п.

Настройка привилегий в базах данных

Разрешение только необходимых SQL-операторов поднимает уровень безопасности. Согласитесь – ни один сайт не будет удалять таблицы, базы данных, создавать новых пользователей и т.п. Следуя правилу «Запрещено все, что не разрешено» отключим все лишние функции. Если ваш сайт использует только SELECT, INSERT, UPDATE, DELETE, то оставьте только их и вы обезопасите свою базу данных.

Например, мы хотим дать пользователю test права на операторы SELECT, INSERT UPDATE, DELETE для базы данных db. Для этого в консоли MySQL от имени администратора необходимо прописать:

```
REVOKE ALL ON db.* FROM test;--Удаление всех прав для пользователя test
GRANT SELECT ON db.* TO test;--Разрешение на оператор SELECT
GRANT UPDATE ON db.* TO test;--Разрешение на оператор UPDATE
GRANT DELETE ON db.* TO test;--Разрешение на оператор DELETE
GRANT INSERT ON db.* TO test;--Разрешение на оператор INSERT
```

Передача шифрованных данных (симметричное шифрование)

Следующий метод защиты заключается в передаче серверу базы данных зашифрованной строки и последующая её расшифровка сервером базы данных. Этот способ разработал Ден Каминский (Dan Kaminsky — известен благодаря обнаружению им фундаментальной уязвимости в DNS). Суть заключается в передаче базе данных информации в base64 представлении (симметричное шифрование, которое не требует ключа для расшифровки). После кодирования информации на сайте в base64 из строки автоматически убираются все опасные символы (экранирование и комментарии), точнее сервер уже не будет рассматривать их как операторы. Для случая с базой данных MySQL это выглядит так:

```
$username;//Переменная, в которую передается опасное значение
mysql_connect('127.0.0.1','test','test');//Подключение к MySQL от имени пользователя test с паролем test
$username=base64_encode($username);//Кодируем строку в base64
mysql_query("SELECT * FROM `users` WHERE `user`=FROM_BASE64('$username'); "); //Выполнение запроса
```

Данный пример верен только для базы данных MySQL версии 5.6 и выше.

Одними из главных преимуществ данного метода является простота реализации и почти абсолютная защита от взлома, но есть и обратная сторона детали. Кодировка и раскодировка base64-строк занимают определенное процессорное время, если для PHP это время минимально и практически не влияет на скорость работы, то нагрузка на базу данных

может существенно увеличится (особенно в случае высоконагружаемых проектов). Также плюсом этого метода является защита от другой атаки XSS (Cross Site Scripting).

Использование необратимого шифрования

Предположим, хакер смог обойти всю вашу защиту и получил доступ к вашей базе данных. Теперь встает другая задача – как сделать, чтобы хакер смог извлечь минимум полезной информации из вашей базы данных. Для этого используют необратимое шифрование. Примерами такого шифрования являются алгоритмы MD5 и SHA1. Необратимое шифрование используется для данных, которые используются для сравнения т.е. нужно знать не конкретное значение, а необходимо лишь сравнить с другими значениями. Примером таких данных является пароль – нам совершенно нет нужды знать его значение т.к. оно нигде не используется – мы лишь сравниваем с паролем, который предлагает нам пользователь. Например:

```
$username;//переменная, в которую заносится имя пользователя
$pass;//переменная, в которую заносится пароль
$md5_pass=md5($pass);
mysql_connect('127.0.01','test','test');
mysql_query("INSERT INTO `users` (`login`,`pass`) VALUES ($username,$md5_pass);");
```

Внесение шифрованных данных в таблицу

```
$username;//переменная, в которую заносится имя пользователя
$pass;//переменная, в которую заносится пароль
$md5_pass=md5($pass);
mysql_connect('127.0.01','test','test');
mysql_query("SELECT * FROM `users` WHERE `login`=$username AND `pass`=$md5_pass;");
```

Проверка на соответствие данных

Теперь даже, если хакер получит доступ к вашей базе данных, то ему будет намного сложнее получить пароли пользователей, потому что на расшифровку данных, которые прошли через MD5 обработку, может уйти несколько лет.

Сохранение данных об активности системы (логирование)

Чаще всего от взлома может защитить только своевременное обнаружение атаки. Для этого необходимо сохранять *критические* данные, которые передаются в скрипт. Существует огромное количество способов реализации данного метода: сохранение во внешний файл, запись в базу данных и другие. Данный метод требует постоянного наблюдения за динамикой информации, однако позволяет вовремя обнаруживать атаку, блокировать её и исправлять ошибки безопасности.

Выводы

Любая система, которая существовала, существует или будет существовать, может быть взломана. В этом лежит суть противоборства хакеров и разработчиков, администраторов. С нахождением новых способов взлома находятся и новые способы защиты, но это противоборство ведется чаще всего на уровне соревнований или на заказ.

Литература

1. Д.Колисниченко. Самоучитель PHP5.
2. С.Хольцнер. PHP в примерах.
3. Л.Зерваас. Web2.0 и приложения на PHP.

КОНЦЕПЦИЯ ОРГАНИЗАЦИИ БЕСПРОВОДНЫХ СЕНСОРНЫХ СЕТЕЙ И ИХ ПРИМЕНЕНИЕ

Смурыгин И.М.
Научный руководитель к.т.н., доц. Власов А.И.
МГТУ им. Н.Э.Баумана, Москва, Россия

CONCEPT OF ORGANIZATION WIRELESS SENSOR NETWORKS AND THEY APLIANCE

Smurygin I.M.
Supervisor: Vlasov A.I.
MSTU, Moscow, Russia

Аннотация
В статье рассматривается основная информация об организации и применении сенсорных беспроводных сетей. Подробно представлена информация об основных проблемах при разработке сенсорных сетей и о преимуществах их применения. Кратко представлена информация о принципах передачи информации в сенсорных беспроводных сетях.

Annotation
In this article base information about organization and appliance of wireless sensor networks is considered. Information about main challenges in development of Wireless Sensor Networks and about main advantages of appliance is presented in details. Main principles of transferring information in wireless sensor networks are given briefly.

Введение

В настоящее время большую популярность приобретают сенсорные беспроводные системы, так как они предоставляют ряд преимуществ потребителю. Но из-за новизны данного вида сетей возникают некоторые сложности при их проектировании.

В данной статье приведены основные принципы функционирования сенсорных беспроводных сетей, а также методы передачи информации при помощи протокола Zigbee.

Анализ сенсорных систем и их применение

Сенсорные сети необходимы для преобразования информации, полученной в результате наблюдения за физическим объектом, в форму, которая может быть использована для хранения информации и ее последующим преобразованием. По результатам обработки информации может быть выработано управляющее воздействие.

Сенсор – объект, который используется для сбора информации о физическом объекте или процессе, включая появление определенных событий (таких как, изменение состояния – изменение температуры или давления) [1]. У человека функцию сенсоров выполняют глаза (восприятие оптической информации), уши (получение акустической информации) и нос (получение информации о запахе).

Также часто используется термин датчик (transducer). *Датчик* – устройство, которое используется для преобразования одного вида энергии в другой [1]. Следовательно, сенсор также является датчиком, который преобразует физическую информацию в электрическую. Далее она может быть передана вычислительной системе или контроллеру для вычислений. Основные шаги, выполняемые при работе сенсорных сетей, представлены на рисунке 1 [1].

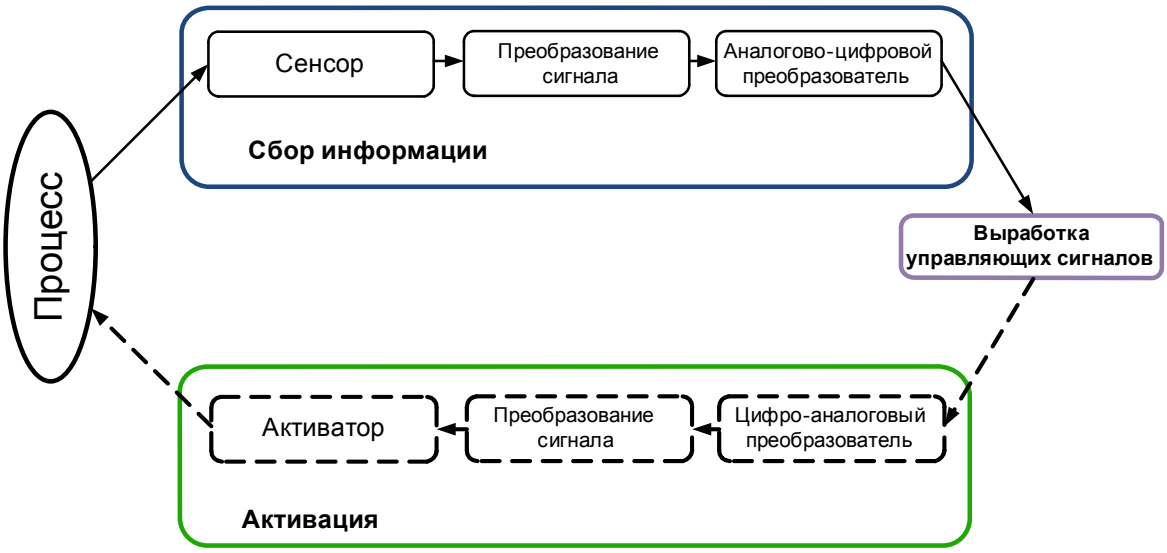


Рисунок 1 – Процесс сбора данных информации и активации процесса в сенсорных сетях

Какое-либо явление в физическом мире наблюдается при помощи сенсора. Полученные электрические сигналы от сенсора часто не готовы для вычислений, поэтому они проходят через стадию преобразования сигнала. На этой стадии может быть осуществлен ряж преобразований, необходимых для дальнейшего использования сигнала. Например, сигнал часто требует усиления, для увеличения амплитуды, затем применяются фильтры для устранения нежелательного шума в определенных промежутках частот. Преобразованный сигнал трансформируется при помощи АЦП в цифровой сигнал. Таким образом, сигнал доступен в цифровой форме и готов к дальнейшему вычислению, хранению и визуализации.

Важное значение при разработке структуры беспроводных сенсорных сетей играет метод построения сети. Классификация методов построения представлена в CMAP-карте на рисунке 2.

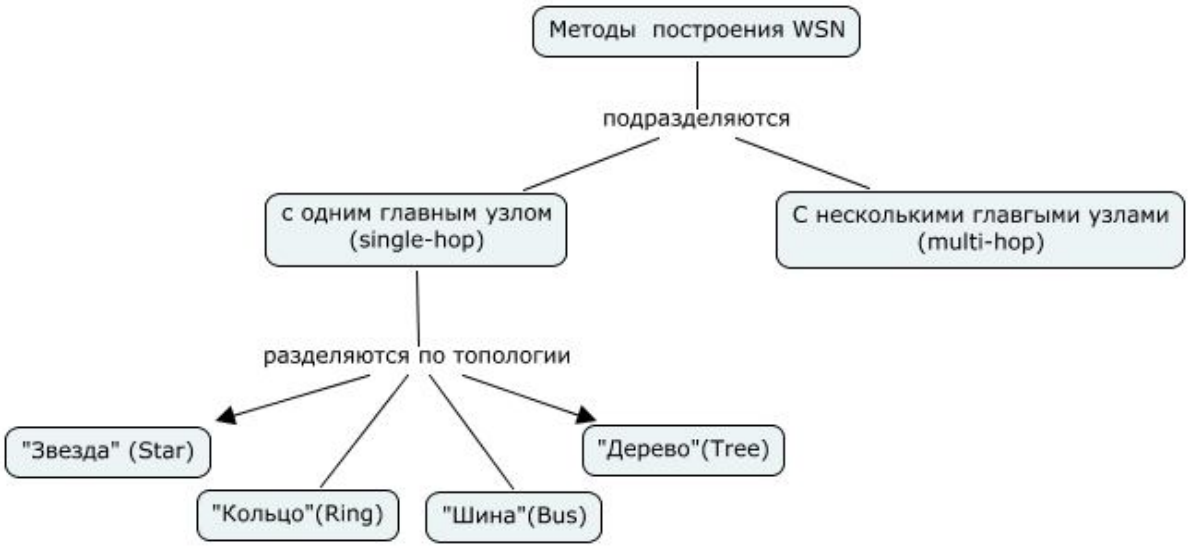


Рисунок 2 – CMAP-карта классификации методов построения сенсорных интеллектуальных сетей

Все методы построения можно объединить в две группы:

- *с одним главным узлом* (*single-hop*). Используется, когда мощность передатчика сенсора достаточна для передачи сигнала к базовой станции.

- *с несколькими главными узлами* (*multi-hop*). В данной топологии некоторые узлы не только собирают информацию о наблюдаемом процессе, но и сбирают информацию от других узлов.

Подробная графическая интерпретация методов построения представлена на рисунке 3.

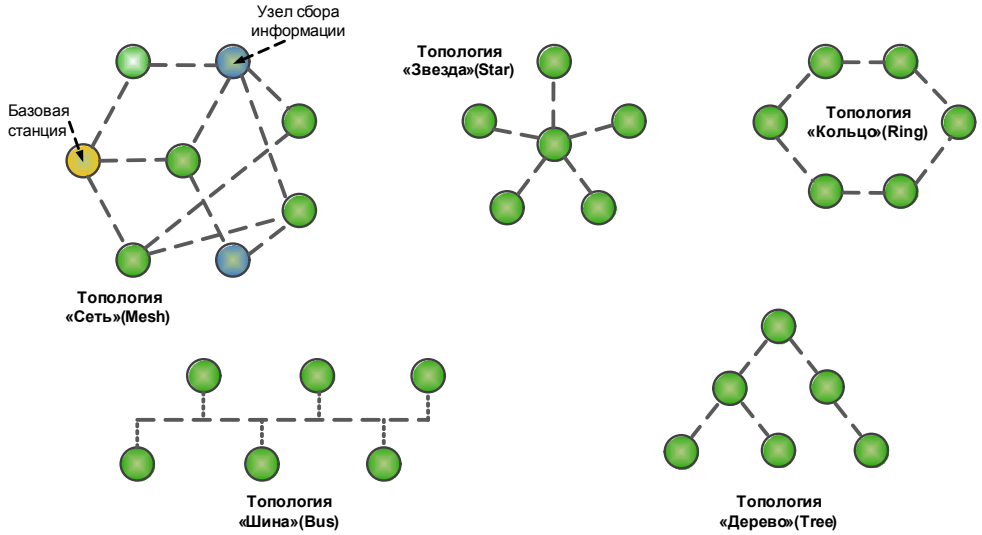


Рисунок 3 – Графическая интерпретация методов построения сенсорных интеллектуальных сетей

Из сравнения двух видов топологий сетей отчетливо видны их достоинства и недостатки. Достоинства и недостатки сенсорных сетей в зависимости от способа их построения приведены в таблице 1.

Таблица 1 – Достоинства и недостатки сенсорных сетей в зависимости от способа их построения

Критерий сравнения	**Сетевая топология с одним главным узлом**	**Сетевая топология с несколькими главными узлами**
Надежность	Выход из строя главного узла приведет к неработоспособности сети	Выход из строя одного из главных узлов не приведет к неработоспособности сети. Главные узлы настраиваются на новый маршрут передачи данных и сеть выполняет необходимые операци
Стоимость	Сложное мощное вычислительное устройство в одном узле – базовой станции. Высокая стоимость	Более простые вычислительные устройства в нескольких узлах.
Энергопотребление	Установка мощных приемо-передающих устройств на базовой станции. Большой расход энергии	Установка нескольких слабых приемо-передающих устройств. Работа узлов только в некоторый промежуток времени, необходимый для передачи информации (*wakeup on demand*[3]). Экономия энергии до 70%[2]

Таблица 1 – продолжение

Критерий сравнения	**Сетевая топология с одним главным узлом**	**Сетевая топология с несколькими главными узлами**
Масштабируемость	Усложнена из-за лимита приема-передачи информации по расстоянию главным узлом	Может поддерживать тысячи индивидуальных устройств. Добавление нового устройства в сеть происходит просто его включением без определенного местоположения.

Сети с несколькими главными узлами являются более эффективными по ряду показателей

Данные виды сетей называют *интеллектуальными*. Т.е. в них заложены следующие функциональная возможность самоуправления. Устройство собирает информацию о своих соседях, адаптируется к изменениям в окружающей среде и формирует соединения между узлами топологии, в зависимости от состояния соседних узлов.

Основные принципы передачи информации при помощи технологии Zigbee

Имеются различные стандарты для передачи данных по сети, видеоданных и другие, однако, до сегодняшних дней не было стандарта, который бы удовлетворял требованиям сенсорных беспроводным сетям. Они не требуют высоких скоростей передачи, но при их работе необходимы низкие задержки при передаче информации, низкое энергопотребление и невысокая стоимость. Так как ни один стандарт не удовлетворял данным требованиям, то был разработан комплект протоколов и расширений для передачи данных в сенсорных беспроводных сетях, которые вошли в стандарт IEEE 802.15.4. Если быть точным, то консорциум компаний ZigBee Alliance разработал комплект протоколов и расширений, дополняющих международный стандарт IEEE 802.15.4.

Согласно определению IEEE 802.15.4 — это стандарт для низкоскоростных частных радиосетей — *Low Rate Wireless Personal Area Network* (LR-WPAN), которые обеспечивают беспроводную связь в различных приложениях с ограниченным энергопотреблением, ограниченной пропускной способностью, при передаче информации на небольшие расстояния от нескольких метров до сантиметров.

Основными преимуществами LR-WPAN являются:

- простота установки;
- надежная передача данных;
- небольшая дальность действия;
- низкая по сравнению с аналогами цена;
- продолжительное время работы от батарейки.

Согласно стандарту[x] сенсорная беспроводная система состоит из нескольких компонентов. Основными компонентами являются два типа устройств:

FFD-устройства (*full function device*). Может выступать в качестве:

- PAN-координатор (*personal area network coordinator*). Наиболее дорогой тип устройств, так как он должен содержать всю информацию о сетевых соединениях, иметь большой объем памяти и высокую производительность;
- координатор. Координатор может реализовывать функции ретранслятора, моста, маршрутизатора и шлюза;
- конечное устройство.

RFD-устройства (*reduced function device*). Могут выступать только в роли конечного устройства. Могут сообщаться только с FFD-устройствами. Применяются в простых приложениях, таких как управление светом. RFD-устройствам не нужно передавать большой объем информации и они могут сообщаться только с одним FFD устройством в определенный момент времени.

Минимальное количество устройств в сети LR-WPAN — 2. Причем в любой сети должно быть по крайней мере одно FFD-устройство, работающее в качестве PAN-координатора. FFD-устройства могут сообщаться с RFD-устройствами или другими FFD-устройствами.

Выбираемый вид топологии зависит от требований приложения. Стандарт IEEE 802.15.4 в LR-WPAN сетях предусматривает применение двух видов топологий: топология «Звезда» (*star topology*) и топология с децентрализованным управлением (*peer-to-peer topology*). Примеры реализаций данных видов топологий представлены на рисунке 4.

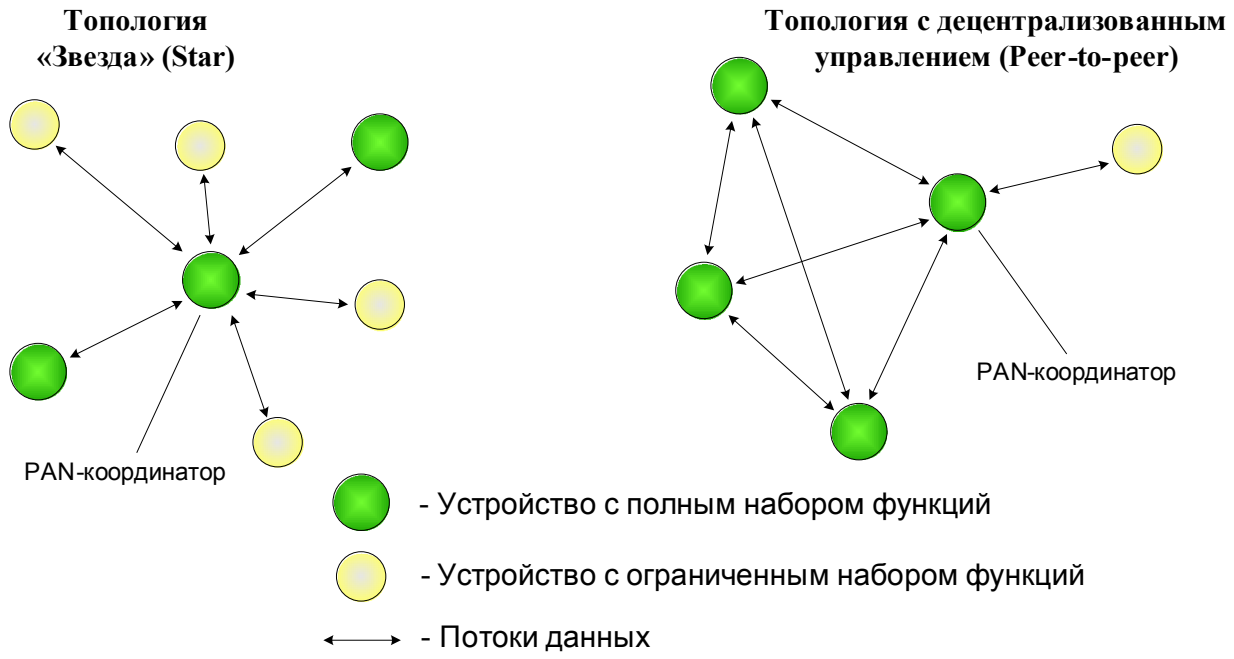


Рисунок 4 – Пример реализации топологии вида «Звезда» и топологии с децентрализованным управлением

В топологии «Звезда» связь осуществляется между несколькими устройствами и единственным центральным контроллером, который называется PAN-координатор. Часто для питания PAN-координатора используется постоянный источник питания, когда остальные узлы питаются от батареек. Топология «Звезда» используется для построения домашних систем автоматического контроля, управления периферией домашних персональных компьютеров и игр.

Топология с децентрализованным управлением отличается тем, что каждое устройство может взаимодействовать с другим устройством на расстоянии, на котором возможно произвести обмен при помощи приемопередающего устройства. Данный вид топологии позволяет построить более сложные конфигурации сетей. Например, данный вид топологии может быть применен для построения систем производственного контроля и мониторинга, **беспроводных сенсорных сетей**, перевозки хрупких товаров и систем умного сельского хозяйства. Топология с децентрализованным управлением может быть самоорганизующейся и иметь возможность автоматического ремонта. Примером применения данного вида топологии может служить структура сети в виде кластерного дерева. Вид данной сети представлен на рисунке 5[2].

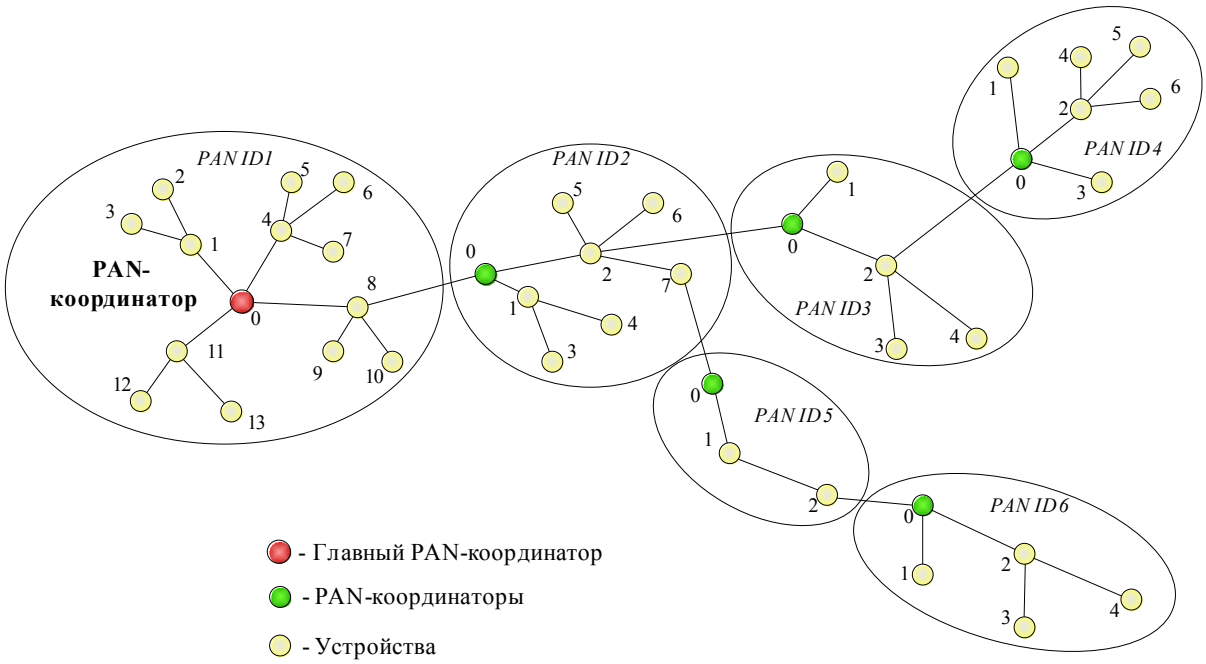


Рисунок 5 – Структура сети в виде кластерного дерева

В данном случае большинство устройств являются устройства с полным набором функций(FFD), RFD присоединяются к сети как конечные устройства, т.к. они не позволяют связывать с ними другие RFD устройства. Любое же FFD устройство может функционировать как координатор и выполнять функцию синхронизатора для других устройств. Только одно устройство может быть главным PAN-координатором. Оно должно содержать большие вычислительные ресурсы, чем все другие устройства в сети. Главный PAN-координатор рассылает сигналы запроса соседним устройствам, формируя кластеры.

Проблемы и ограничения

Так как сенсорные беспроводные сети имеют отличия от обычных сетей, то они также имеют собственные ограничения и проблемы [1]. Основные проблемы сенсорных беспроводных сетей представлены в CMAP-карте на рисунке 6.

Рисунок 6 – CMAP-карта проблем и ограничений в сенсорных беспроводных сетях

Рассмотрим основные проблемы и ограничения более подробно.

Энергопотребление. Ограничение по энергопотреблению связанно с тем, что сенсоры работают от источника питания с ограниченным лимитом энергии(обычно батарейка). Чем реже они будут заменяться или заряжаться, тем более низкую стоимость будет иметь их обслуживание.

Также энергопотребление является важным ограничением при использовании сенсоров, доступ к которым осложнен, следовательно, источник питания не может быть заменен или подзаряжен. Например, научное наблюдение за сдвигом ледников, где требуется работа сенсоров в течение нескольких лет.

Требование на ограничение энергопотребления распространяется на сенсорный узел и на конструкцию сети. Например, выбор сделанный на физическом слое сенсорного узла влияет на энергопотребление всего устройства и конструкцию уровней, расположенных выше [3]. Потребление энергии процессоров, произведенных по CMOS технологии, главным образом зависит от энергии переключения и энергии утечки [4]:

$$E_{CPU} = E_{switch} + E_{leakage} = C_{total} V_{dd}^2 + V_{dd} I_{leak} \Delta t$$

где -емкость переключений при вычислительных операциях; - напряжение питания; - ток утечки, - время операции.

Энергия переключения ограничивает работу процессора по частоте. В наше время энергия переключения составляет более половины энергии работы процессора [5].

Также для уменьшения энергопотребления предусмотрено отключение передатчиков сенсорных узлов, когда нет необходимости передачи информации. На сетевом уровне обнаруживаются оптимальные пути передачи информации от сенсорного узла до базовой станции, учитывая число узлов, требуемую энергию и доступную энергию.

Кроме сетевого протокола на потребление энергии влияет конструкция узлов (например, маленький размер памяти, эффективность переключений между заданиями), программное обеспечение, механизмы защиты и даже рабочие приложения.

Самоуправление. Сенсорные сети часто должны работать в удаленных областях и в жестких условиях, без возможности их обслуживания и починки. Поэтому, сенсорные узлы должны конфигурироваться самостоятельно, взаимодействовать с другими узлами, адаптироваться к поломкам изменениям окружающей среды без вмешательства человека.

Самоуправление может быть представлено в нескольких формах:

1) Самоорганизация. Способность сети конфигурировать параметры системы. Например, сенсорное устройство может выбрать силу передачи данных, чтобы получить доступ к устройствам на бо́льших расстояниях;
2) Автоматическая оптимизация. Возможность устройств наблюдать имеющиеся ресурсы и оптимизировать их использование;
3) Самозащита. Позволяет устройству распознавать и защищать себя от сторонних вторжений и атак;
4) Возможность автоматического ремонта. Позволяет сенсорным узлам реагировать на изменения в сети и вносить изменения для их правильного функционирования

Беспроводное соединение. Выбор беспроводного соединения налагает ряд ограничений на конструкцию сенсорных сетей. Например, затухание сигнала ограничивает расстояние передачи информации. Связь между переданной и принятой информацией описывается законом обратных квадратов:

$$P_{пр} \sim \frac{P_{пер}}{d^2},$$

где - мощность принятого сигнала, - мощность переданного сигнала, – расстояние между приемником и передатчиком.

Следовательно, увеличение расстояния между сенсорным узлом и базовой станции приводит к увеличению мощности передаваемого сигнала. Поэтому более эффективно, с точки зрения затрат энергии, разделить большие расстояния на несколько маленьких.

Для меньшего энергопотребления можно использовать различные подходы. Например, устройства могут содержать два приемо-передающих устройств. Приемопередающее устройство с низким потреблением энергии используется для формирования сигналов включения узла. Или адаптивное управление данными[6], когда часть узлов остается включенными в определенное время.

Децентрализованное управление. Алгоритмы построения многих сенсорных сетей строятся по централизованному принципу. При децентрализованном управлении сенсорные узлы должны обмениваться информацией с соседними узлами, чтобы сгенерировать решении о коммутации узлов, без глобальной информации о всей сети. В следствии, децентрализованные алгоритмы могут быть не оптимальными, но более эффективные в отношении энергии, чем централизованные.

Например, при централизованном управлении базовая станция может опрашивать все сенсорные узлы, принимать от них информацию, сообщать каждому узлу свой маршрут передачи информации. При частом изменении сети потери будут значительны. Децентрализованный подход позволяет каждому узлу делать собственное решение при наличии небольшой информации (список соседних устройств, включающий информацию о расстоянии до базовой станции). В данном случае потери на управление будут уменьшены значительно.

Конструкция. В данное время возможности компьютерных систем продолжают быстро увеличиваться, однако главной целью беспроводных сенсорных сетей является создание маленьких, дешевых и эффективных устройств. Из-за требования к низкому потреблению энергии типичный сенсорный узел имеет небольшие скорости выполнения операций и объемы хранимой информации. Также из-за этого нежелательно использование некоторых устройств, таких как GPS приемники. Ограничения по размерам влияет на структуру протоколов и алгоритмов, реализованных в беспроводных сенсорных сетях. Например, таблица всех маршрутов в сети может быть слушком большой и не поместиться в памяти узла. Поэтому только небольшая часть информации (например, список соседних узлов) может храниться в памяти узла.

Таким образом, многие технические решения должны быть применены для работы на сильно ограниченном по параметрам оборудовании.

Безопасность. Удаленное расположение сенсоров и их автоматическая работа увеличивает их незащищенность к сторонним вторжениям и атакам. При беспроводном соединении достаточно легко для нарушителя перехватить пакеты, передаваемые сенсорным узлом. Например, наиболее большая угроза осуществления атаки «отказа в обслуживании» (*denial-of-service*), цель данной атаки нарушить корректное функционирование сенсорной сети. Это может быть достигнуто при помощи различных способов, например, при подаче мощного сигнала, который мешает сенсорным узлам обмениваться информацией («белый шум» или jamming attack) [1]. Есть различные варианты защиты систем от злоумышленников, но для многих из них необходимы высокие требования к аппаратным ресурсам, что труднодостижимо на жестко ограниченных по многим требованиям сенсорных узлах. Следовательно, сенсорные беспроводные сети требуют новых решений для создания ключей, их распространения, идентификации и защиты узлов.

Заключение

Разработка и введение сенсорных сетей во все сферы жизни предоставит огромное количество преимуществ человечеству. Тематика сенсорных беспроводных сетей еще не достаточно изучена, имеются на данный момент ряд нерешенных проблем и ограничений, но преимущества привлекают компании для разработки стандартов передачи информации в сенсорных сетях, например, стандарт передачи данных ZigBee.

Данный вид сетей является интеллектуальным, то есть имеет возможность автоматического определения устройств и маршрутов их сообщения.

Литература

1. Waltenegus Dargie, Christian Poelabauer «Fundamentals of Wireless Sensor Networks Theory and Practice» - Wiley Series on Wireless Communications and Mobile Computing, 311c, 2010 John Wiley & Sons Ltd.
2. IEEE Standard for Information technology—Telecommunications and information exchange between systems— Local and metropolitan area networks— Specific requirements. Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs). IEEE Std 802.15.4™-2006 (Revision of IEEE Std 802.15.4-2003)
3. Shih, E., Bahl, P., and Sinclair, M. (2002) Wake-on wireless: An event driven energy saving strategy for battery operated devices. Proc. of the ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom).
4. Sinha, A., and Chandrakasan, A.P. (2000) Energy aware software. Proc. of the 13th International Conference on VLSI Design.
5. De, V., and Borkar, S. (1999) Technology and design challenges for low power and high performance. Proc. Of the International Symposium on Low Power Electronics and Design (ISLPED).
6. Ye,W., Heidemann, J., and Estrin, D. (2004) Medium access control with coordinated adaptive sleeping for wireless sensor networks. *IEEE/ACM Transactions on Networking* 12 (3), 493–506.

Результаты работы получены при частичной поддержке проекта №7.6161.2011

МОДУЛЬ ОБРАБОТКИ ЧЕРТЕЖЕЙ САПР ЛИНЕЙНОГО ТРАНСПОРТА: 1. ЧЕРТЕЖИ ПЛАНА

Горин Я.А.
Научный руководитель к.т.н. Волосатова Т.М.
Кафедра РК6, МГТУ им. Н.Э. Баумана, Москва, Россия

MODULE OF DRAWING PROCESSING CAD OF THE LINEAR TRANSPORT: 1. DRAWINGS OF PLAN

Gorin J.A.
The scientific chief: Ph.D. senior lecturer Volosatova T.M.
Department RK6, BMSTU, Moscow, Russia

Аннотация
В работе приведен анализ задачи обработки изыскательских чертежей плана. Рассмотрены основные особенности, налагаемые методом получения исходных данных. Предложены алгоритмы, позволяющие решить поставленную задачу. Разработано соответствующее ПО и проиллюстрированы результаты его работы.
Abstract
An overview of the problem of processing survey drawings of the plan is given. The main features imposed by the method of obtaining the source data are discussed. Algorithms to solve the problem are suggested. Developed by appropriate software and illustrated the results of his work.

Цифровая модель рельефа (ЦМР) является одной из важных составных частей геоинформационных технологий. Традиции и практика представления этих моделей неразрывно связаны с технологиями формирования топографических карт и планов [5].

В настоящее время основным методом получения информации о рельефе в большинстве регионов РФ является оцифровка существующих картографических материалов. Рельеф на картах представлен наборами высотных отметок и структурных линий. Высотные отметки обычно определяют характерные точки рельефа, например, локальные экстремумы. Структурные линии задают дополнительные ограничения на форму рельефа. Это могут быть, прежде всего, горизонтали (изолинии, изогипсы) и области резкого изменения наклона поверхности (границы оврагов, береговые линии, линии насыпей, обрывов и т.п.) [4].

Процесс оцифровки достаточно трудоемкий и монотонный, выполняется многими операторами, поэтому неизбежно возникает много ошибок, как атрибутных, так и графических. В подавляющем большинстве это различные пики, петли на линиях, осцилляции или излишняя сглаженность, приводящая к неоправданному росту объемов данных. Применение известных методов аппроксимации для устранения данных ошибок может привести к недопустимым пересечениям изолиний разных уровней и другим топологическим нарушениям.

Современные подходы к решению задач этой направленности характеризуются применением программных средств для обработки данных. Следует отметить качественные различия между “ручными” и “автоматизированными” решениями в этой области по представлению ЦМР. В частности, основная часть затрат при производстве цифровых карт и планов определяется ручной коррекцией автоматизированных решений [5].

Качество цифровой модели рельефа (ЦМР) зависит, прежде всего, от качества исходных данных. Поэтому необходимым этапом в процессе построения ЦМР является предварительная обработка исходных данных о рельефе [4].

Существует ряд работ посвященных данной тематике [1-5].

Место модуля в САПР линейного транспорта

На рис. 1. представлена архитектура САПР линейного транспорта.

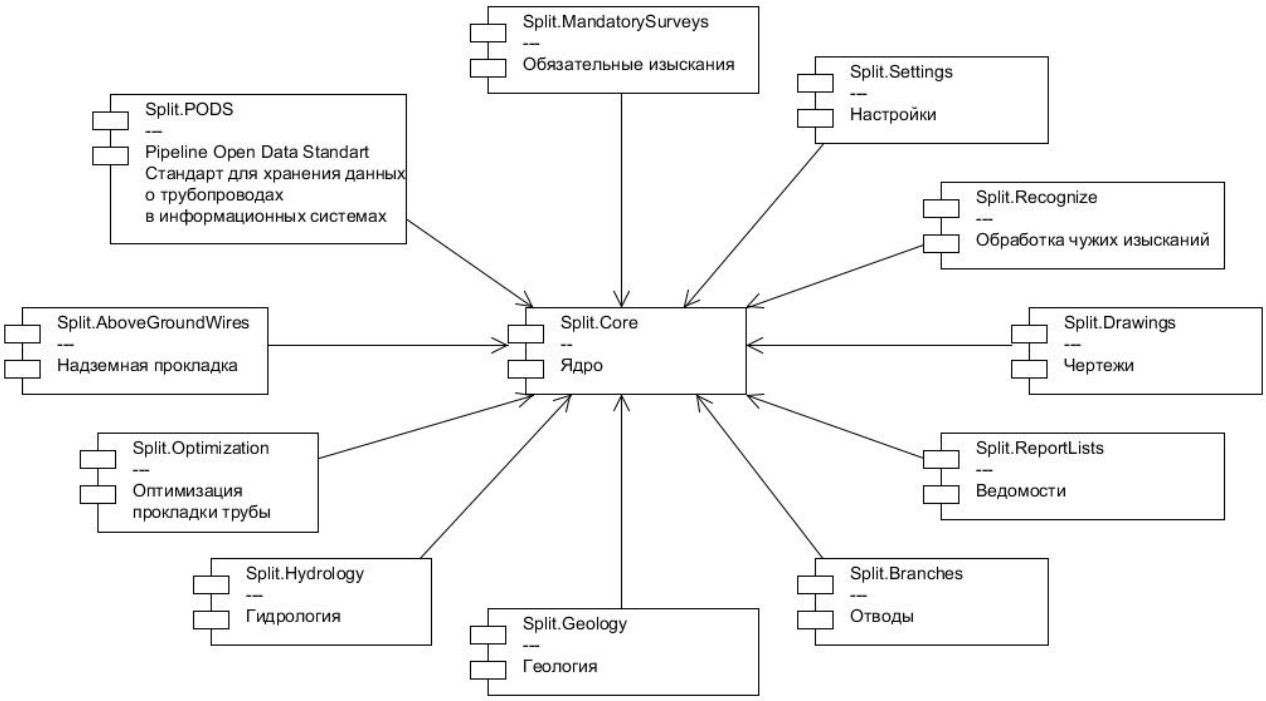


Рисунок 1 - Архитектура САПР линейного транспорта

Структура модуля обработки изыскательских чертежей. Модуль обработки изыскательских чертежей состоит из двух основных компонент – обработки чертежей профиля и обработки чертежей плана (см. рис. 2.).

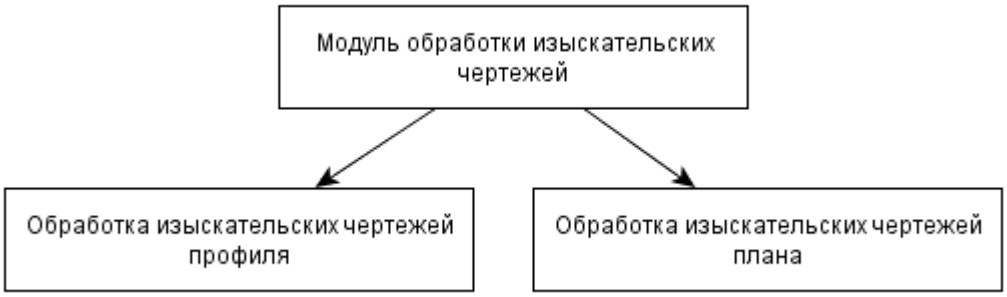


Рисунок 2 - Структура модуля обработки изыскательских чертежей

Место модуля в последовательности обработки данных. Модуль обработки изыскательских чертежей занимает начальное место в последовательности обработки данных. Существует два способа начать работу в САПР линейного транспорта:

1. Создать новый проектируемый участок,
2. Обработать изыскательские чертежи и на основании полученных данных сгенерировать проектируемый участок.

Второй вариант более предпочтительный (при наличии изыскательских чертежей), так как в нем возрастает доля автоматизации процесса получения проектируемого участка.

Постановка задачи обработки чертежей плана

Имеется топоплан (карта рельефа) в формате DWG.

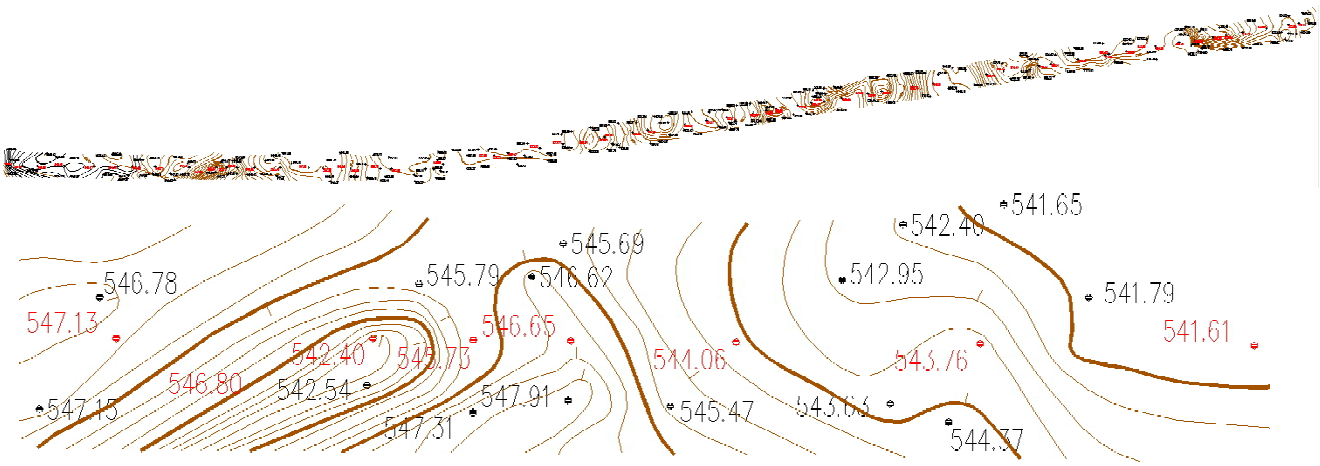


Рисунок 3 - Исходные данные - топоплан

Топоплан представлен наборами высотных отметок и структурных линий. Основной целью обработки чертежей плана является построение ЦМР. Необходимо гарантировать высокую точность обработки исходных данных и построения ЦМР при минимальной помощи со стороны пользователя.

Данную задачу можно “разбить” на ряд подзадач:

1. Предварительная обработка (фильтрация) исходных данных с целью избавления от выбросов. Выбросы – точки (или тексты), лежащие вне коридора проектирования трассы, но имеющие такое же графическое обозначение как и точки (или тексты), лежащие внутри коридора проектирования трассы.
2. Обработка точек рельефа (“привязывание” точек рельефа к их высотным отметкам)
3. Обработка горизонталей (определение высотных отметок для горизонталей)
4. Построение ЦМР (с использованием информации как о точках рельефа, так и о горизонталях).

Анализ задачи обработки чертежей плана

Задача обработки чертежей плана обладает следующими особенностями:

1. Неоднозначность, неполнота и противоречивость исходных данных.

Неоднозначность исходных данных связана с различными способами обозначениями точек и текстов на чертежах.

Неполнота исходных данных обусловлена методом получения исходных данных. Исходными данными при проектировании линейных объектов является **коридор проектирования трассы**. Коридор проектирования трассы получают путем “вырезания” из топоплана больших масштабов прилегающих к намеченной трассе участков (следовательно вместо того, чтобы обрабатывать весь топоплан больших масштабов, приходится обрабатывать лишь его часть).

Противоречивость исходных данных (наличие ошибок в исходных данных) связана с появлением точек, тексты которых остались вне коридора проектирования трассы, и текстов, точки которых остались за пределами коридора проектирования, в процессе “вырезания”.

2. Нехватка информации для однозначного принятия решения

Нет формального критерия принадлежности точки тексту. Относительное расположение точки и соответствующего ей текста различно в рамках одного чертежа.

Данные особенности позволяют отнести данную задачу к классу плохоформализуемых задач [6].

Анализ алгоритмов

Из постановки задачи и анализа исходных данных следует, что для решения поставленных задач в большинстве случаев придется принимать решение в условиях неопределенности (делать предположения о доминировании одного решения над другим).

Необходимость в предположениях возникает всякий раз, когда для выбора решения из множества возможных решений необходима дополнительная информация. В этом случае делается предположение о доминировании одного решения над другим. Если в ходе решения появится новая информация, противоречащая данным предположениям, тогда необходимо вернуться к точке принятия предположений и пересмотреть их, сделав новый выбор [6].

Поиску и анализу таких алгоритмов обработки чужих изысканий посвящена данная работа.

Алгоритмы обработки чертежей плана

Условные графические изображения точек рельефа и горизонталей могут отличаться в разных чертежах. Избавиться от данной неоднозначности исходных данных, можно введя настройки распознавания и/или, разработав инструмент, позволяющий пользователю определять графические объекты (шаблоны), являющиеся точками рельефа и горизонталями. По данным шаблонам будут найдены все аналогичные графические объекты, которые в дальнейшем будут интерпретироваться как точки рельефа и горизонтали.

Алгоритмы фильтрации исходных данных

Подходы к решению задачи фильтрации (отсеивания выбросов):

1. **Эвристический.**

Отсеивать точки, радиусы до ближайших текстов которых много больше среднего расстояния от точки до ближайшего к ней текста. Данный подход может быть применен при условии, что количество выбросов много меньше количества точек внутри коридора проектирования.

2. **Кластеризация.**

Необходима кластеризация, основанная на близости объектов (affinity-base clustering, spectral clustering). В качестве объектов кластеризации можно использовать либо точки рельефа, либо тексты высотных отметок, либо точки рельефа и тексты высотных отметок. В качестве меры близости можно использовать евклидово расстояние.

3. **Линейное программирование. Венгерский алгоритм.**

Данный подход основан на предположении, что на чертежи присутствуют выбросы только одного типа (либо “лишние” точки рельефа, либо “лишние” тексты). В этом случае сформулировав задачу о назначении текстов точкам в терминах линейного программирования, мы получаем не квадратную матрицу стоимостей. Следовательно, для того чтобы решать задачу методами линейного программирования следует ввести нужное количество строк или столбцов и их элементам присвоить значения, определяемые условиями решаемой задачи.

В случае если выбросами являются точки, следует дополнить матрицу “мнимыми” текстами, присвоив при этом каждой ячейке столбца (строки) стоимость равную 0. Тогда при минимизации суммы расстояний от точек до текстов, выбросам будут присвоены “мнимые” тексты. После чего точки соответствующие “мнимым” текстам могут быть исключены из дальнейшего рассмотрения. Аналогично следует поступить в случае, если выбросами являются тексты.

Алгоритмы обработки точек рельефа

На данном этапе предполагается, что исходные данные не содержат выбросов либо были отсеяны на этапе фильтрации.

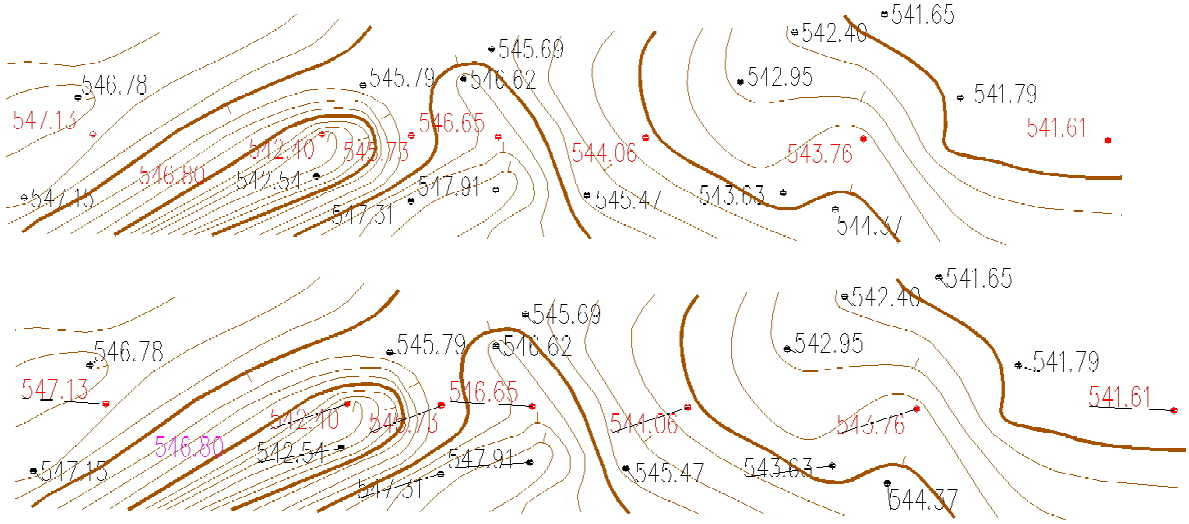


Рисунок 4 - Исходные данные и результат работы алгоритма обработки точек рельефа

Подходы к решению задачи “привязывания” точек рельефа к их высотным отметкам:

1. **Итерационный.**

Пока количество точек или текстов не равно 0:

Связать точку с ближайшим текстом, и исключить данный текст и данную точку из дальнейшего рассмотрения.

2. **Линейное программирование. Венгерский алгоритм.**

Задачу привязывания” точек рельефа к их высотным отметкам можно переформулировать в терминах линейного программирования как задачу о назначении. Основной целью при данном подходе является минимизация сумму расстояний от точек до текстов. Венгерский алгоритм — алгоритм оптимизации, решающий задачу о назначениях за O(n^3).

3. **Машинное обучение. Обучение нейросети с учителем**

В основе данного подхода лежит предположение о том, что плотность распределения точек и текстов на всех чертежах одинаковая. Данное предположение выглядит вполне разумно, так как коридор для проектирования трассы является вырезкой из топоплана (карты) высокой точности.

3.1. Для каждой точки определить кротчайшее расстояние до ближайшего текста.

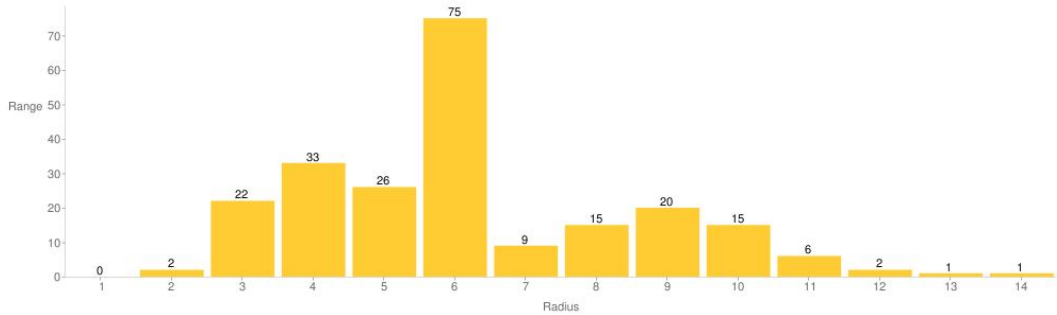


Рисунок 5 - Распределение, характеризующее текущий чертеж

3.2. Объединить точки в группы, на основании “схожести” расстояний, полученных в 3.1. Получить распределение количества точек данного радиуса от величины радиуса

P(N,r). Данное распределение в какой-то степени характеризует текущий чертеж (см. рис. 5.).

3.3. Попытаться "сжать" данные о распределении. Ввести параметры математического ожидания и дисперсии подобно нормальному распределению (либо параметр лямбда, аналогично пуассоновскому распределению).

3.4. Для текущего чертежа получить оптимальный радиус. Большой радиус приведет к сильному "задействованию" пользователя в процессе распознавания и тем самым сократит долю автоматического распознавания. Малый радиус приведет к неточному распознаванию (определенное количество точек вовсе будут не привязаны к текстам).

3.5. Найти отображение (функцию) из сжатых данных характеризующих распределение (см. пункт 3.2.) в оптимальный радиус для данного чертежа. С этой целью можно использовать **линейную регрессию.** При этом данные аппроксимируются прямой линией (по обучающим данным строится прямая, ошибка отклонения от которой минимальна).

4. **Искусственный интеллект. Моделирование логики человека при решении данной задачи.**

4.1. Генерируется **область поиска** вокруг каждой из точек рельефа.

4.2. В данной области ищутся тексты. Информация о найденных текстах сохраняется.

4.3. Запускается **алгоритм автоматического распознавания** для тривиальных случаев. Если не все точки распознаны goto 3., иначе goto 4.

4.4. Определение высоты точки при помощи пользователя в нетривиальном случае. Если не все точки распознаны goto 3., иначе end

Основную идею **алгоритм автоматического распознавания** для тривиальных случаев можно сформулировать следующим образом:

Если в **области поиска** данной точки есть только один свободный (не привязанный к какой-либо точке) текст, необходимо присвоить данный текст данной точке. Если в области поиска данной точки есть **n** текстов и среди них **(n-1)** текстов уже привязаны к каким-либо точкам, необходимо присвоить данный текст оставшейся точке.

В качестве **области поиска** используется окружность радиуса R. Для выбора радиуса R необходимо проанализировать весь чертеж целиком. От выбора радиуса R зависит как правильность автоматизированного распознания, так и количество точек распознанных автоматически:

- Если R слишком большой, то все случаи будут не тривиальными и пользователю придётся определять значения высот для каждой точки.
- Если R будет слишком мал, то точки, скорее всего, будут распознаны неправильно.

В реализованном в программе алгоритме радиус R выбирается следующим образом:

1. Для каждой точки определить кротчайшее расстояние до ближайшего текста.
2. Выбрать максимальное расстояние.
3. Увеличить максимальное расстояние на 1, что гарантирует пересечение текста (а не касание), и принять его за радиус поиска R.

Алгоритмы обработки горизонталей

На этапе распознавания горизонталей предполагается, что завершился этап распознавания точек рельефа и точки рельефа распознаны правильно.

Так как данная задача относится к классу плохоформализуемых задач, одним из возможных подходов к её решению является моделирование логики человека при решении данной задачи.

Реализованный в программе алгоритм автоматизированной обработки горизонталей:

1. Сгенерировать **область поиска** вокруг распознанных точек.
2. В данной области найти точки “соседние” с текущей

3. Текущую точку последовательно “соединить” секущей линией с “соседними” точками.

4. Определить количество горизонталей пересечённых данной линией.

5. Округлить значение высоты текущей и соседней точек рельефа с точностью до **шага горизонталей** (является настройкой). Получить их разность и разделить на **шаг горизонталей.** Получить количество линий в тривиальном случае (тривиальный случай заключается в том, что между 2-мя точками высота горизонталей либо возрастает, либо убывает; сложный случай – между 2-мя точками существуют различные участки, на которых значении горизонталей то возрастают, то убывают).

6. Если кол-во горизонталей вычисленное в пункте 4. совпадает с количеством горизонталей вычисленном в 5., присвоить соответствующие значения высот горизонталям и отметить их как распознанные. Иначе горизонтали считать нераспознанными.

Программный продукт

Целью данной программы является предварительная обработка исходных данных о рельефе необходимых для построения ЦМР. Программный продукт оформлен в виде динамический библиотеки. Команды, предназначенные для обработки чертежей плана, перечислены ниже.

1. Команда **recognizePoints** вызовет на выполнение алгоритм “привязывания” точек рельефа к их высотным отметкам. При возникновении неоднозначностей алгоритм будет переходить в интерактивный режим (запрашивать помощь у пользователя). В результате работы алгоритма на чертеже появятся линии соединяющие соответствующие тексты и точки. Тексты и точки для которых не нашлось соответствие, будут выделены соответствующим цветом (см. рис. 2.). Это позволит визуально оценить правильность распознавания.

2. Команда **recognizePointsStatistic** вызовет на выполнение алгоритм, позволяющий получить распределение количества точек данного радиуса от величины радиуса P(N,r) и отобразить его с помощью Google Chart Tools. Данная команда разработана в исследовательских целях для подтверждения или опровержения предположения принятого при рассмотрении подхода, основанного на технологии обучения нейросети с учителем.

3. Команда **recognizeHorizontals** вызовет на выполнение алгоритм обработки горизонталей. При возникновении неоднозначностей алгоритм будет переходить в интерактивный режим (запрашивать помощь у пользователя). В результате работы алгоритма необработанные горизонтали будут выделены соответствующим цветом. Это позволит визуально оценить правильность распознавания.

Литература

1. Костюк Ю.Л., Фукс А.Л. Предварительная обработка исходных данных для построения цифровой модели рельефа местности - Вестник ТГУ, 2003. №280. С. 281-285.

2. Костюк Ю.Л., Фукс А.Л. Построение цифровой модели рельефа местности на основе структурных линий и высотных отметок - Вестник ТГУ, 2003. №280. С. 286-289.

3. Фукс А.Л. Разработка и исследование алгоритмов интерполяции однозначных поверхностей и их использование при построении цифровых моделей рельефа: Диссер. на соискание уч. степ. к.т.н. Томск, ТГУ, 2001.

4. **http://www.inf.tsu.ru/library/DiplomaWorks/CompScience/2004/Kravchenko/diplom.pdf** - Предварительная обработка структурных линий для построения цифровой модели рельефа местности

5. Чернова Л.И. Обработка геопространственной информации при цифровом моделировании топографических карт – Автореферат на соискание уч. степ. к.т.н. Иркутск, ИрГТУ, 2006.

6. http://fedoruk.comcor.ru/AI/ai_lect.html - Искусственный интеллект в САПР

7. Горин Я.А. «Инженерные изыскания для строительства линейных сооружений». Отчет по научно-исследовательской практике. – МГТУ им. Н.Э.Баумана, 2011.

МОДУЛЬ ОБРАБОТКИ ЧЕРТЕЖЕЙ САПР ЛИНЕЙНОГО ТРАНСПОРТА: 2. ЗАДАЧА КЛАСТЕРИЗАЦИИ

Горин Я.А.
Научный руководитель к.т.н. Волосатова Т.М.
Кафедра РК6, МГТУ им. Н.Э. Баумана, Москва, Россия

MODULE OF DRAWING PROCESSING CAD OF THE LINEAR TRANSPORT: 2. CLUSTERING PROBLEM

Gorin J.A.
The scientific chief: Ph.D. senior lecturer Volosatova T.M.
Department RK6, BMSTU, Moscow, Russia

Аннотация
В работе рассматривается алгоритм кластеризации точек на плосткости. Алгоритм является комбинацией алгоритмов кластеризации на основе теории графов и агломеративной кластеризации. Рассмотрены основные особенности исходных данных. Разработано соответствующее ПО и проиллюстрированы результаты его работы.

Abstract
An overview of clustering 2D-points algorithm is given. The algorithm is a combination of clustering algorithms based on graph theory and agglomerative clustering. The main features of the initial data are discussed. Developed by appropriate software and illustrated the results of his work.

Введение

При обработке чертежей плана возникает задача фильтрации исходных данных с целью избавления от выбросов. Выбросы – точки (или тексты), лежащие вне коридора проектирования трассы, но имеющие такое же графическое обозначение, как и точки (или тексты), лежащие внутри коридора проектирования трассы [5].

Одним из подходов к решению данной задачи является кластеризация точек рельефа. Помимо избавления от выбросов кластеризация позволяет разбить задачу на подзадачи в случае размещения нескольких коридоров проектирования трассы на одном чертеже.

Основные обозначения

Пусть множество $I = \{I_1, I_2, \ldots, I_n\}$ обозначает n точек на плоскости. Каждая точка из I обладает некоторым множеством *наблюдаемых* показателей или характеристик $C = (C_1, C_2, \ldots, C_p)^T$. В общем случае наблюдаемые характеристики могут быть как *количественными,* так и *качественными.* Количественные данные иногда называют *измерениями*. Результат измерения *i*-ой характеристики I_j точки будем обозначать символом x_{ij}, а вектор $X_j = [x_{ij}]$ размерности $p \times 1$ будет отвечать каждому ряду измерений (для j-ой точки). Таким образом, для множества точек I исследователь располагает множеством векторов измерений $X = \{X_1, X_2, \ldots, X_n\}$, которые описывают множество I.

Выбор наблюдаемых характеристик (определение признаков оценки объектов). В качестве наблюдаемых характеристик выбираем координаты x и y геодезических точек. Координата z, во-первых, отсутствует в явном виде на изыскательском чертеже, во-вторых, должна быть исключена из рассмотрения по определению *коридора проектирования трассы* (см. [5]). Единицы измерение x и y совпадают. Следовательно, результат не будет зависеть от масштабов, от единиц измерения признаков. В нормировании наблюдаемых характеристик нет необходимости.

Задача кластерного анализа

Пусть m – целое число, меньшее, чем n. Задача кластерного анализа заключается в том, чтобы на основании данных, содержащихся в множестве X, разбить множество точек I на m кластеров (подмножеств) $\pi_1, \pi_2, \dots, \pi_m$ так, чтобы каждый объект I_i принадлежал одному и только одному подмножеству разбиения и чтобы объекты, принадлежащие одному и тому же кластеру, были *сходными*, в то время как объекты, принадлежащие разным кластерам, были разнородными (*несходными*) [1].

Решением задачи кластерного анализа является разбиение, удовлетворяющее некоторому критерию оптимальности. Этот критерий может представлять собой некоторый функционал, выражающий уровни желательности различных разбиений и группировок. Этот функционал называют *целевой функцией* [1].

Целями кластеризации приминительно к решаемой задаче являются:

1. Понять структуру множества объектов X^l, разбив его на группы схожих объектов. Упростить дальнейшую обработку данных и принятия решений, работая с каждым кластером по отдельности (стратегия “разделяй и властвуй”). В этом случае число кластеров стараются сделать как можно меньшим.

2. Выделить нетипичные объекты, которые не подходят ни к одному из кластеров. Эту задачу называют одноклассовой классификацией, обнаружением нетипичности или новизны (novelty detection).

Расстояния между объектами (метрика).

Для определения сходства между объектами используется понятие расстояния $d(X_i, X_j)$. Чем меньше расстояние, тем более похожими считаются объекты. Наиболее употребительные функции расстояний приведены в [1].

Учитывая специфику решаемой задачи, остановимся на использовании *евклидова расстояния* в качестве меры близости между точками на плоскости:

$$d_2(X_i, X_j) = \left[\sum_{k=1}^{p} (x_{ki} - x_{kj})^2\right]^{\frac{1}{2}}$$

В качестве **расстояний между кластерами** используются длины ребер, соединяющих кластеры.

Реализованный метод кластеризации

Реализованный метод кластеризации состоит из двух этапов:

1. Кластеризация на основе теории графов,
2. Иерархическая агломеративная кластеризация на основе кластеров полученных на первом этапе.

Таким образом, мы получаем более тонкую, двухуровневую, структуру кластеров: каждый кластер верхнего уровня распадается на более мелкие подкластеры нижнего уровня.

Определение числа кластеров. В разработанном алгоритме число кластеров зависит от параметров кластеризации на первом и втором этапе - α_1 и α_2.

Кластеризация на основе теории графов

Обширный класс алгоритмов кластеризации основан на представлении выборки в виде графа. Вершинам графа соответствуют объекты выборки, а рёбрам - попарные расстояния между объектами $d_{ij} = d(X_j, X_i)$.

На плоскости заданы n точек. Необходимо построить дерево, вершинами которого являются все заданные точки и суммарная длина всех ребер минимальна. Такое дерево называется *Евклидовым Минимальным Остовным Деревом* (*ЕМОД, EMST - Euclidean Minimum Spanning Tree*).

ЕМОД множества S из n точек на плоскости может быть построено на основании триангуляции Делоне множества S, за оптимальное время $O(n)$. В свою очередь

триангуляция Делоне множества S может быть построена за время $O(n \cdot log(n))$. Следовательно, ЕМОД множества S из n точек на плоскости может быть построено за оптимальное время $O(n \cdot log(n))$.

Детальное описание построение ЕМОД можно найти в [3].

Удаление любого из ребер ЕМОД разбивает дерево на лес. Существует множество способов выбрать ребро для удаления.

В реализованном алгоритме используется модификация идеи изложенной в [4]. При выборе ребра – кандидата на удаление мы сравниваем его длину с длинами других ребер, прилегающих к его вершинам. Назовем ребро r *несовместимым*, если его длина l значительно больше $\hat{l}$, где $\hat{l}$ - длина минимального ребра, смежного ребру r.

В реализованном алгоритме удаляются все ребра, чья длина $l > \alpha_1 \cdot \hat{l}$. На момент написания статьи оптимизация параметра α_1 не проводилась, $\alpha_1 = 5$.

В качестве дополнительного критерия для удаления ребра может выступать – длина отклонения от *диаметрального пути*. Диаметральный путь – самый длинный путь по дереву. Когда точки данных располагаются в длинные цепочки, минимальное покрывающее дерево образует естественный скелет для цепочки.

Иерархическая агломеративная кластеризация

Среди алгоритмов иерархической кластеризации различаются два основных типа. Дивизимные или нисходящие алгоритмы разбивают выборку на всё более и более мелкие кластеры. Более распространены агломеративные или восходящие алгоритмы, в которых объекты объединяются во всё более и более крупные кластеры.

На данном этапе работы алгоритма уже имеется определенное число кластеров, полученных с помощью кластеризации на основе теории графов. В качестве связей между кластерами будем рассматривать удаленные на первом этапе ребра. Будем возвращать ребро r между кластерами U и V, удаленное на предыдущем этапе, если его длина l значительно меньше как $\overline{l_U}$, так и $\overline{l_V}$, где $\overline{l_U}$ и $\overline{l_V}$ – средние длины ребер в соседних ребру r кластерах.

В реализованном алгоритме возвращаются все ребра, удаленные на предыдущем этапе, чья длина $l < \alpha_2 \cdot \overline{l_U}$ и $l < \alpha_2 \cdot \overline{l_V}$. На момент написания статьи оптимизация параметра α_2 не проводилась, $\alpha_2 = \alpha_1$.

Функционалы качества кластеризации

Для сравнения качества разбиения на классы используется ряд функционалов качества. Задачу кластеризации можно ставить как задачу дискретной оптимизации: необходимо так приписать номера кластеров Y_i объектам X_i, чтобы значение выбранного функционала качества приняло наилучшее значение. По сути дела, каждый метод кластеризации можно рассматривать как точный или приближённый алгоритм поиска оптимума некоторого функционала. Существует много разновидностей функционалов качества кластеризации, наиболее употребляемые из них могут быть найдены в [1,2].

Так как основой данного метода кластеризации является кластеризация на основе теории графов, можно воспользоваться какой-либо статистикой, получаемой из МОД. Одной из полезных статистик, получаемой из МОД, является *распределения длин ребер*. В качестве функционала предлагается использовать функцию от *распределения длин ребер* МОД, построенного на точках относящихся к одному кластеру.

Достоинства и недостатки.

Достоинствами графовых алгоритмов кластеризации являются:

1. наглядность,
2. относительная простота реализации,
3. возможность вносить различные усовершенствования, опираясь на простые геометрические соображения.

Результаты работы алгоритма

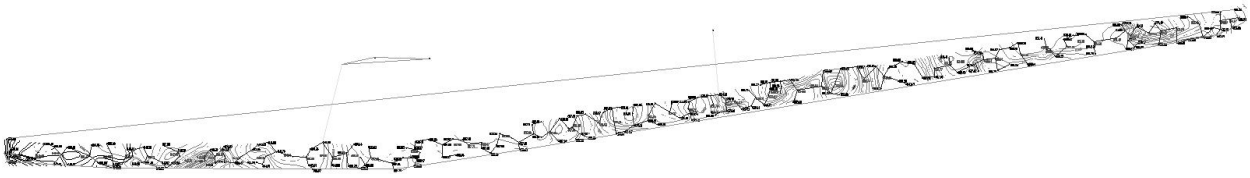


Рисунок 1 - Единственный кластер и наличие “выбросов”

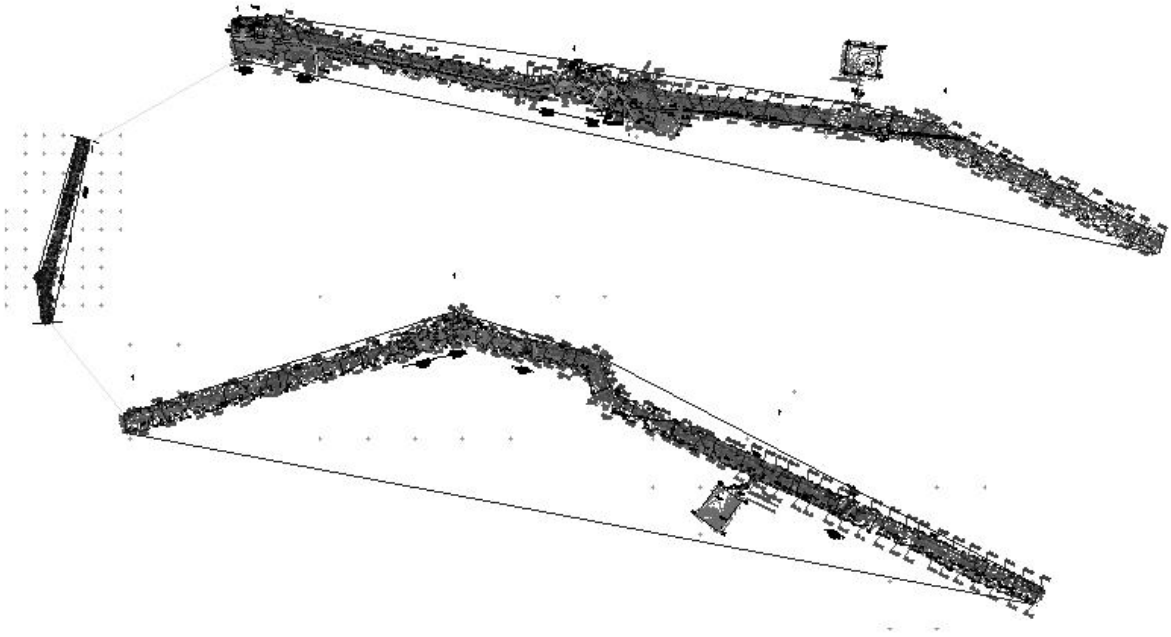


Рисунок 2 - Три кластера и отсутствие “выбросов”

Рисунок 3 - Четыре кластера и наличие “выбросов”

Достоинством реализованного графового алгоритма также является *устойчивость при масштабировании*. Под устойчивостью к масштабированию здесь понимается адекватность кластеризации в случае различных масштабов коридоров проектирования трассы, расположенных в одном пространстве.

Алгоритм обладает цепочечным эффектом, когда независимо от формы кластера к нему присоединяются ближайшие к границе объекты.

К н**едостаткам** алгоритма можно отнести неадекватность кластеризации при наличии разреженного фона или “узких перемычек” между кластерами.

Литература

1. Дюран Б., Оделл П. Кластерный анализ
2. Воронцов К. В. Лекции по алгоритмам кластеризации и многомерного шкалирования. 21 декабря 2007 г.
3. Препарата Ф., Шеймос М. Вычислительная геометрия: Введение. – М.: Мир, 1989.
4. Дуда Р., Харт П. Распознавание образов и анализ сцен. - М.: Мир, 1976.
5. Горин Я.А. Модуль обработки чертежей САПР линейного транспорта: чертежи плана. – Сборник статей 14-ой международной научно-технической конференции "Наукоемкие технологии и интеллектуальные системы", Москва, 2012
6. Загоруйко Н. Г. Прикладные методы анализа данных и знаний. – Новосибирск: ИМ СО РАН, 1999.
7. Загоруйко Н. Г., Ёлкина В. Н., Лбов Г. С. Алгоритмы обнаружения эмпирических закономерностей. - Новосибирск: Наука, 1985.
8. Кулаичев А. П. Методы и средства комплексного анализа данных. – М: ИНФРА- М, 2006.
9. Лагутин М. Б. Наглядная математическая статистика. - М.: П-центр, 2003.
10. Мандель И. Д. Кластерный анализ. - М.: Финансы и Статистика, 1988.
11. Уиллиамс У. Т., Ланс Д. Н. Методы иерархической классификации // Статистические методы для ЭВМ / Под ред. М. Б. Малютов. - М.: Наука, 1986. - с. 269–301.
12. Jain A., Murty M., Flynn P. Data clustering: A review // ACM Computing Surveys. - 1999.- Vol. 31, no. 3.- Pp. 264–323. http://citeseer.ifi.unizh.ch/jain99data.html.
13. Lance G. N., Willams W. T.A general theory of classification sorting strategies. 1. hierarchical systems //Comp. J. 1967. - no. 9. - Pp. 373–380.

МОДУЛЬ ОБРАБОТКИ ЧЕРТЕЖЕЙ САПР ЛИНЕЙНОГО ТРАНСПОРТА: 3. ЧЕРТЕЖИ ПРОФИЛЯ

Горин Я.А.
Научный руководитель к.т.н. Волосатова Т.М.
Кафедра РК6, МГТУ им. Н.Э. Баумана, Москва, Россия

MODULE OF DRAWING PROCESSING CAD OF THE LINEAR TRANSPORT: 3. DRAWINGS OF PROFILE

Gorin J.A.
The scientific chief: Ph.D. senior lecturer Volosatova T.M.
Department RK6, BMSTU, Moscow, Russia

Аннотация
В работе приведен анализ задачи обработки чертежей профиля. Рассмотрены основные особенности, налагаемые методом получения исходных данных. Предложены алгоритмы, позволяющие решить поставленную задачу. Разработано соответствующее ПО и проиллюстрированы результаты его работы.
Abstract
An overview of the problem of processing drawings of the profile is given. The main features imposed by the method of obtaining the source data are discussed. Algorithms to solve the problem are suggested. Developed by appropriate software and illustrated the results of his work.

Продольным профилем называется проекция развертки трассы на вертикальную плоскость [2]. Чертеж продольного профиля с нанесенным на него проетируемым линейным объектом является одним из основных документов, на основании которого осуществляется строительство.

Изыскательский чертеж продольного профиля является основой для проектирования линейного объекта. На стадии проектирования чертеж профиля с нанесенным на него инженерно-геологическим разрезом (ИГР) и уровнем подземных вод (УПВ) является основным источником информации о геологии и гидрологии. Структура САПР линейного транспорта приведена в [4].

Постановка задачи обработки чертежей профиля

Имеется чертеж профиля в формате DWG.

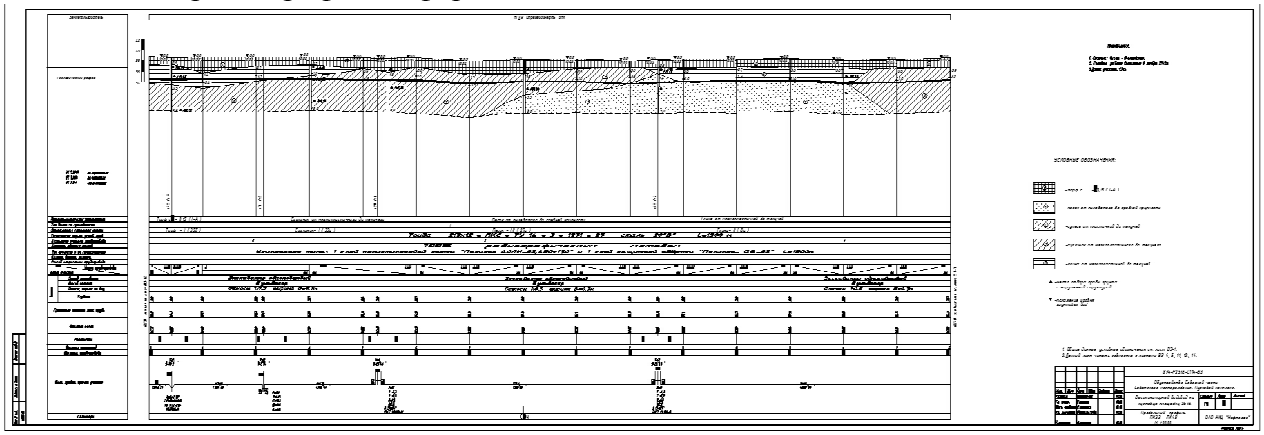


Рисунок 1 - Исходные данные – чертеж профиля

Необходимо обработать чертёж профиля с целью получения информации о:

1. линии профиля (строки “Высотные отметки”, “Расстояния”, “пикетаж” подвала, масштабы чертежа (горизонтальный, вертикальный, вертикальный геологический)),

2. характерных точках на линии профиля (ординаты),
3. углах поворота трассы (строка “Развернутый план” подвала),
4. геологии.

Необходимо гарантировать высокую точность обработки исходных данных, при минимальной помощи со стороны пользователя.

Анализ задачи обработки чертежей профиля

Чертеж продольного профиля по существу состоит из двух частей:

1. подвала - сетки с горизонтальными графами, в которых приведены цифровые данные полевых и проектных работ.
2. верхней графической части, которая изображает вертикальный разрез проектируемого линейного объекта вдоль его оси.

Подвал в свою очередь можно разделить на 2 части:

1. “Тело” подвала (табличные данные) – часть подвала, в которой приведены цифровые данные полевых и проектных работ.
2. Боковик (“заголовок”) подвала – часть подвала, содержащая названия горизонтальных граф, в которых приведены цифровые данные полевых и проектных работ.

Основной особенностью задачи обработки чертежей профиля является неоднозначность исходных данных, вызванная существованием как различных форм боковиков таблицы, так и компоновкой чертежа профиля в целом.

Анализ алгоритмов

Из постановки задачи и анализа исходных данных следует, что для решения поставленных задач в большинстве случаев придется принимать решение в условиях неопределенности (делать предположения о доминировании одного решения над другим).

Необходимость в предположениях возникает всякий раз, когда для выбора решения из множества возможных решений необходима дополнительная информация. В этом случае делается предположение о доминировании одного решения над другим. Если в ходе решения появится новая информация, противоречащая данным предположениям, тогда необходимо вернуться к точке принятия предположений и пересмотреть их, сделав новый выбор [1]. Поиску и анализу таких алгоритмов обработки чужих изысканий посвящена данная работа.

Алгоритмы обработки чертежей профиля

При обработке изыскательских чертежей профиля используются **метод предположений** и **метод поиска по маскам** текстов на чертеже с использованием регулярных выражений.

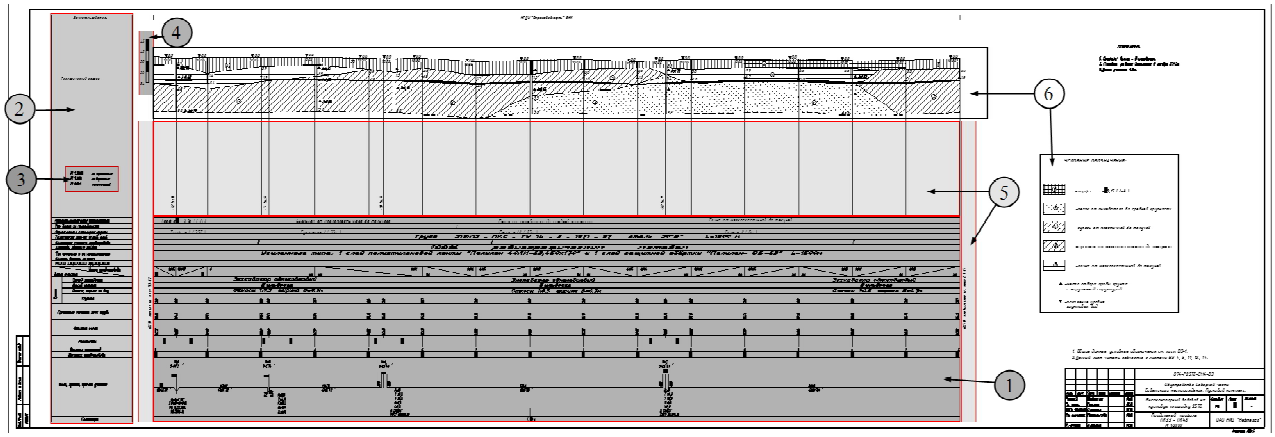


Рисунок 2 - Информация, содержащаяся на чертеже профиля, разбитая на логические группы: 1 – подвал; 2 – боковик подвала; 3 – масштабы; 4 – “линейка” высот; 5 – ординаты; 6 – ИГР

Реализованный в программе алгоритм автоматизированного распознавания изыскательских чертежей профиля:

1. Ищем тексты, соответствующие заголовку подвала, в соответствии с масками (маски указываются в настройках чертежа, если таковых настроек нет, то используются настройки по умолчанию)
2. Объединяем тексты в группы, на основании их геометрического расположения (тексты заголовка подвала находятся в одном столбце "подвала" (на одной вертикали))
3. Задаём приоритет группам (на основании количества текстов в группе и расположения на чертеже (приоритет имеют группы текстов находящиеся левее и имеющие больше всего текстов))
4. Заносим группы с приоритетами в дерево возможных решений
5. Выбираем очередную группу и пытаемся продолжить распознавание
6. Проверяем наличие ячеек таблицы вокруг текстов из группы и совпадение правых и левых границ ячеек.
   Если проверка вернула отрицательный результат goto 5.
   Иначе goto 7.
7. Ищем горизонтальные линии, такие что:
   7.1. находятся правее правой границы заголовка подвала,
   7.2. их координатами Y соответствуют координатам верхних и нижних границ ячеек
8. Объединяем полученные в 7. горизонтальные линии в группы на основании схожести (с определенной точностью (1мм)) их начальных и конечных значений координат X
9. Задаём приоритет группам (На основании количества линий в группе, длины линий, и расположения относительно правой границы подвала)
10. Заносим группы с приоритетами в дерево возможных решений
11. Выбираем очередную группу и пытаемся продолжить распознавание
12. Распознаем строки "Расстояния", "Высотные отметки", "Изыскательский пикетаж"
    12.1. Распознаем Масштаб
    12.2. Определяем координату Y начала линии профиля
13. Распознаем строку "развернутый план"
14. Определяем расположение ординат
15. Распознаем ординаты
16. Определяем расположение информации о геологии
17. Распознаем геологию

Распознавание строк подвала “Высотные отметки”, “Расстояния” и “Пикетаж”

Перед началом распознавания строк подвала “Высотные отметки”, “Расстояния” и “Пикетаж” необходимо определить масштаб и “линейку” высот (см. рис. 3.).

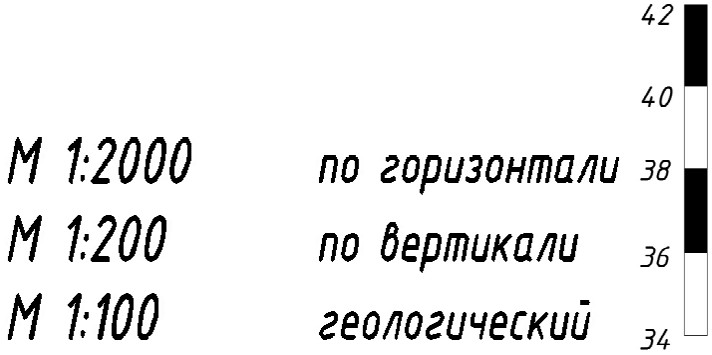


Рисунок 3 - Масштаб и “линейка” высот

Реализованный в программе алгоритм автоматизированной обработки строк подвала “Высотные отметки”, “Расстояния” и “Пикетаж” (см. рис.4.):

1. Получить строки “Высотные отметки”, “Расстояния” и “Пикетаж”.
2. Создать линию профиля по меткам Пикетажа (по длинным вертикальным линиям в графе "Расстояния")
3. Добавить начальную и конечную точки линии профиля
4. Восстановить значения пикетов (в том случае если пикеты нумеруются от 1 до 9 (после 9-ки следует пустое место))
5. Найти тексты в строке подвала "Расстояния" находящиеся между соответстующимися пикетами
6. Найти тексты в строке подвала "Высотные отметки" находящиеся между соответстующимися пикетами
7. Сопоставить тексты из строки “Расстояния” с текстами из строки “Высотные отметки”.

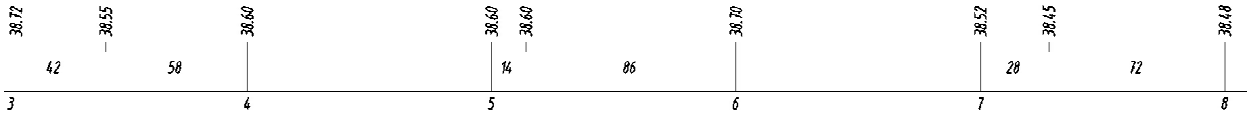


Рисунок 4 - Строки подвала “Высотные отметки”, “Расстояния” и “Пикетаж”

Распознавание строки подвала “Развёрнутый план”

Развернутый план трассы предназначен для удобства проектирования проектной линии и дальнейшего использования продольного профиля для разбивочных работ.
Развернутый план содержит информацию о:

1. вершинах углов поворота пути, их обозначениях и номерах, начале и конце кривых, их обозначениях и привязке к пикетам;
2. числовых значениях элементов кривых: углах поворота, радиусах, тангенсах, суммарных длинах круговых и переходных кривых, длинах переходных кривых;
3. пикетах и указателях километров;

При обработке строки подвала “Развернутый план” можно выделить 2 типа объектов:

1. Линейные участки
2. Углы поворота трассы (холодные и горячие)

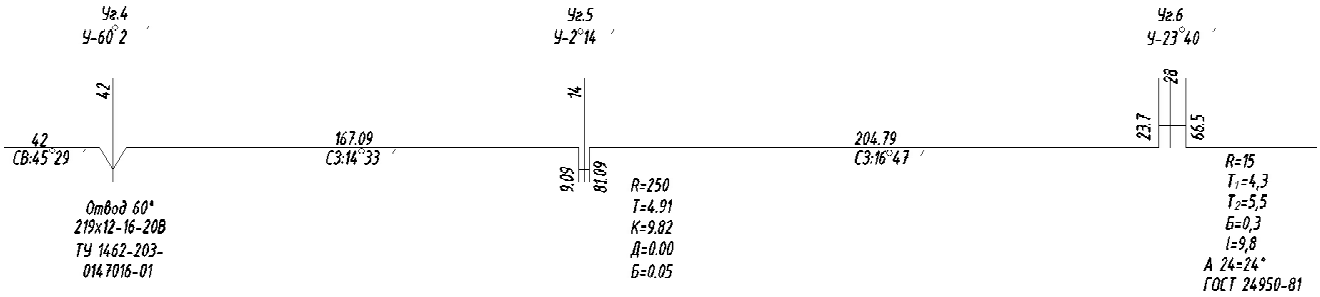


Рисунок 5 - Строка подвала “Развернутый план”

Реализованный в программе алгоритм автоматизированной обработки строки подвала “Развёрнутый план” (см. рис 5.):

1. Ищем линейные участки (всегда представлены в строке “развернутого плана” горизонтальными линиями)
2. Определяем холодные и горячие углы (используя информацию о линейных участках)
3. Ищем группы текстов, характеризующие прямые участки, и связываем графическую информацию прямых участков с соответствующими группами текстов
4. Ищем группы текстов, характеризующие холодные и горячие углы (используя информацию о линейных участках), и связываем графическую информацию холодных и горячих углов с соответствующими группами текстов

5. Обрабатываем группы текстов, используя регулярные выражения, с целью получения параметров, характеризующих прямые участки, холодные и горячие углы

Распознавание ординат

Положение ординат можно определить, зная координаты точек линии профиля. Под линией профиля находятся геология, ординаты и подвал. Тексты ординат всегда ориентированы вертикально.

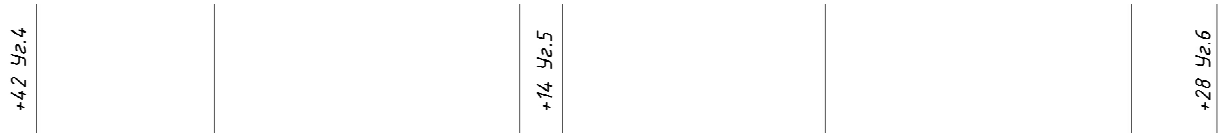


Рисунок 6 - Ординаты

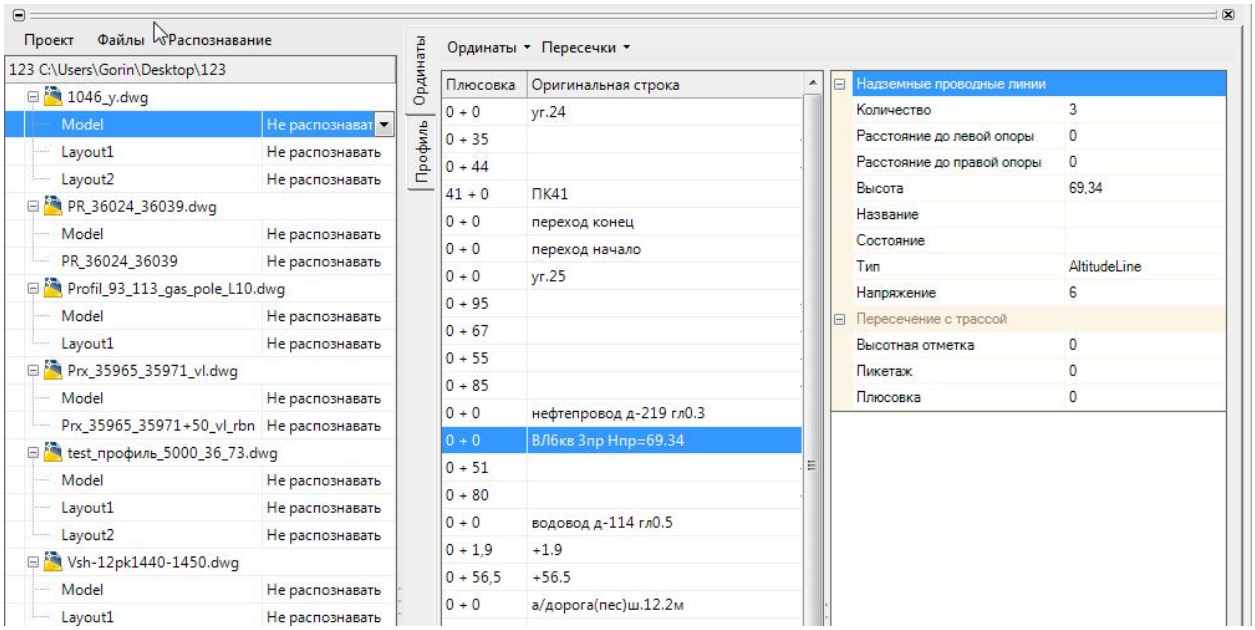


Рисунок 7 - GUI, позволяющее отображать и редактировать результат распознавания ординат

Обрабатываем текстовую информацию ординат, используя регулярные выражения, с целью получения информации о пересечках и характерных точках на линии профиля (см. рис. 6.). Результат распознавания ординат на одном из тестовых чертежей представлен на рис. 7.

Литература

1. http://fedoruk.comcor.ru/AI/ai_lect.html - Искусственный интеллект в САПР.
2. Б. И. Каменецкий, И. Г. Кошкин “Автомобильные дороги”,1979, стр. 37 – 40.
3. Горин Я.А. «Инженерные изыскания для строительства линейных сооружений». Отчет по научно-исследовательской практике. – МГТУ им. Н.Э. Баумана, 2011.
4. Горин Я.А. Модуль обработки чертежей САПР линейного транспорта: чертежи плана. – Сборник статей 14-ой международной научно-технической конференции "Наукоемкие технологии и интеллектуальные системы", Москва, 2012.

КО-АЛГОРИТМИЧЕСКАЯ ГИБРИДИЗАЦИЯ МЕТОДА РОЯ ЧАСТИЦ

Воробьева Е.Ю., Селиверстов Е.Ю.
Научный руководитель: д. ф.м. н., профессор, Карпенко А.П.
МГТУ им. Н.Э.Баумана, Москва, Россия

CO-ALGORITHMIC HYBRIDIZATION OF PARTICLE SWARM OPTIMIZATION (PSO)

Vorobeva E.Y., Seliverstov E.Y.
Supervisor: Dr., Prof., Karpenko A.P.
MSTU, Moscow, Russia

Аннотация
В статье рассматривается ко-алгоритмическая гибридизация двух алгоритмов роя частиц, использующих топологии соседства частиц типа клика и кольцо. Каждая из топологий алгоритма роя частиц имеет свои недостатки, которые можно устранить, используя их ко-алгоритмическую гибридизацию. Выполнено сравнение эффективности стандартного метода роя частиц и его ко-алгоритмической модификации, найдены оптимальные значения свободных параметров ко-алгоритма.

Abstract
In this paper the co-algorithm, based on two Particle Swarm Optimization (PSO) algorithms, is represented. Every topology of basic PSO has flaws. Co-algorithm hybridization of PSO algorithms can eliminate the flaws, because it has two swarms with click and ring neighborhood topologies. In this paper comparison of co-algorithm PSO and basic PSO efficiency is implemented and optimal value of free arguments are founded.

Введение

В основе любого оптимизационного ко-алгоритма лежит следующая основная идея [1, 2]. Одновременно в пространстве поиска эволюционирует несколько субпопуляций, базирующихся на выбранных (базовых) алгоритмах, которые решают данную задачу оптимизации. Во время заданного числа итераций каждая из этих субпопуляций «борется» за вычислительные ресурсы, которые по окончании этого числа итераций перераспределяются в пользу более эффективной из субпопуляций.

В настоящее время алгоритмы роя частиц (*PSO*) эффективно применяются для решения задач глобальной многомерной оптимизации [3]. Метод имеет большое число модификаций. Целью работы является исследование эффективности ко-алгоритмической гибридизации двух алгоритмов *PSO*. Реализация ко-алгоритма выполнена в среде пакета MATLAB.

Ко-алгоритмическая гибридизация алгоритмов PSO

Рассмотрим задачу многомерной глобальной безусловной оптимизации

$$\min_{X \in R^{|X|}} F(X) = F(X^*) = F^* .$$

Здесь и далее запись вида $|A|$, где A – некоторый вектор, означает размерность этого вектора.

Введем следующие обозначения: P_1 - субпопуляция, соответствующая алгоритму *PDO* с топологией типа клика; P_2 - аналогичная субпопуляция алгоритма *PSO* с топологией типа кольцо; P_Σ - ко-алгоритмическая популяция. В качестве функций приспособленности субпопуляций $\varphi_1(X), \varphi_2(X)$ используем лучшее значение целевой функции $F(X)$, достигнутое соответствующей субпопуляцией.

Схема ко-алгоритма имеет следующий вид [2].

1) Задание значений свободных параметров субпопуляций.

2) Инициализация субпопуляций - присвоение каждому агенту начального положения и скорости в пространстве поиска.

3) Независимое выполнение Δ_a итераций каждой из субпопуляций.

4) Оценка эффективности субпопуляции.

5) Перераспределение ресурсов субпопуляций.

6) Проверка выполнения условия останова (достижение заданного числа итераций $\hat{t}$, стагнация вычислительного процесса или исчерпание допустимого числа вычислений значений целевой функции) и если это условие не выполнено, возврат к шагу 3.

Перераспределение ресурсов на шаге 5 осуществляется путем сокращения размера проигравшей субпопуляции на величину штрафа n_p и увеличения размера победившей субпопуляции на ту же величину. Общий размер популяции P_Σ при этом не меняется. Победившей считается субпопуляция с лучшим (меньшим) достигнутым значением функции приспособленности.

Исследование эффективности ко-алгоритма

Сравнительное исследование эффективности ко-алгоритма и алгоритмов *PSO* с топологиями типа клика и кольцо выполнено методом мультистарта (пять стартов) для известных тестовых функций Розенброка (одноэкстремальная овражная функция), Химмельблау (четырехэкстремальная функция) и Растригина (многоэкстремальная функция). В качестве условия окончания итераций использовано условие достижение заданного числа итераций $t = \hat{t} = 50$. Результаты исследования показывают, что лучшее решение F^*, найденное ко-алгоритмом, всегда лучше лучших решений, найденных базовыми алгоритмами *PSO*.

В работе также выполнено исследование эффективности ко-алгоритма в зависимости от значений следующих его свободных параметров: интервал адаптации Δ_a; штрафной коэффициент n_p; величина минимального размера субпопуляции $|S|_{\min}$. В качестве меры эффективности субпопуляций P_1, P_2 и популяции P_Σ использована лучшее среднее по числу мультистартов значение целевой функции $\overline{F}^*$.

Результаты исследования эффективности ко-алгоритма в зависимости от параметра Δ_a иллюстрирует рисунок 1. Рисунок показывает, что зависимость величины $\overline{F}^*$ от параметра Δ_a имеет различный характер для субпопуляций P_1, P_2. При $\Delta_a = 5; 8; 9; 10$ эффективность *PSO* с топологией клика выше эффективности этого алгоритма с топологией кольцо. При $\Delta_a = 6; 7$ имеет место обратная ситуация. В результате ко-алгоритм имеет максимальную эффективность (минимальные значения величины $\overline{F}^*$) при $\Delta_a = 5; 9; 10$.

Рекомендованные значения штрафного коэффициента n_p при коэволюции генетических алгоритмов лежат в интервале 0,05÷0,3. В общем случае значения этого параметра выбирают из следующих соображений. Значение штрафа не следует назначать слишком маленьким, поскольку в таком случае субпопуляционные алгоритмы могут не ощутить на себе действия штрафа. Также значение величины n_p не должно быть слишком большим, так как в этом случае субпопуляционные алгоритмы могут быстро достичь своего минимально допустимого размера $|S|_{\min}$, при котором они априори малоэффективны. В работе рассмотрен интервал значений параметра n_p, равный [0,15; 0,4]. Исследование показало, что ко-алгоритм имеет максимальную эффективность при $n_p = 0{,}15; 0{,}2$.

В качестве диапазона возможных значений величины $|S|_{\min}$ рассмотрен интервал $[0{,}05 \cdot |S|; 0{,}3 \cdot |S|]$. Исследование влияния величины $|S|_{\min}$ на эффективность ко-алгоритма показало, что минимальные значения величины $\overline{F}^*$ достигается при $|S|_{\min} = 0{,}2 \cdot |S|$.

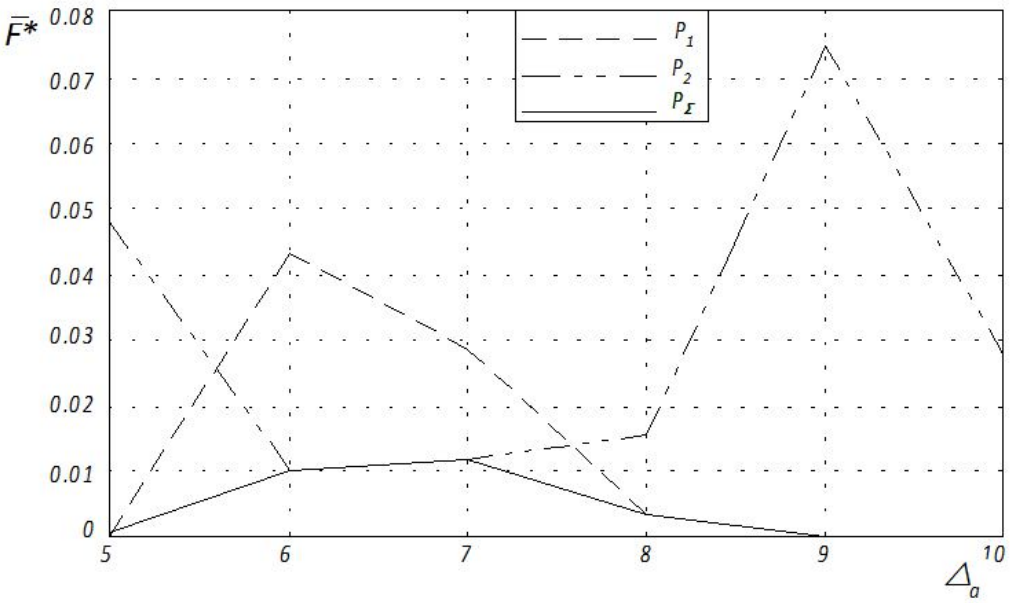


Рисунок 1 – Зависимость эффективности популяций $\overline{F}^*$ от параметра Δ_a: функция Розенброка; n_p = 0,15; $|S|_{\min}$ =0,25$\cdot|S|$

Значительный интерес представляет процесс перераспределения ресурсов между субпопуляциями. Данный процесс иллюстрирует рисунок 2, показывающий, что субпопуляция-лидер становится известной в течение примерно 30 итераций.

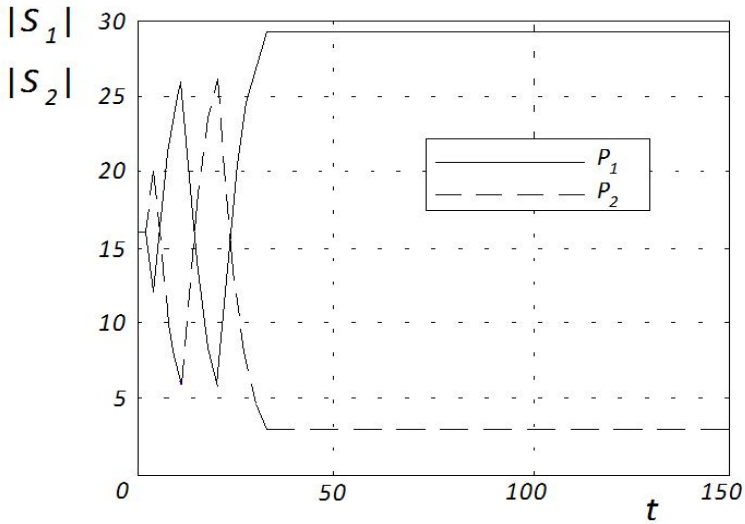
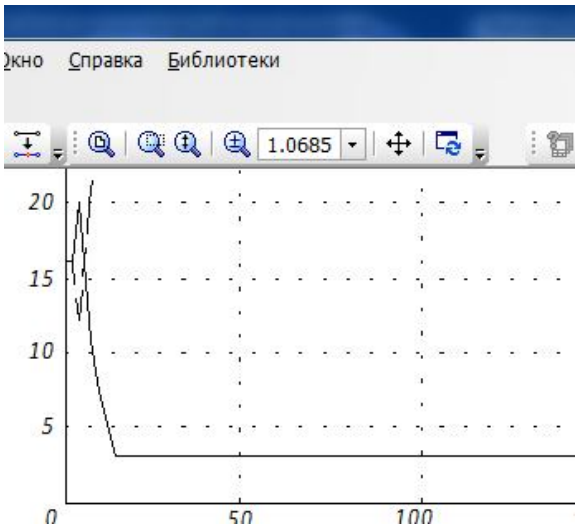
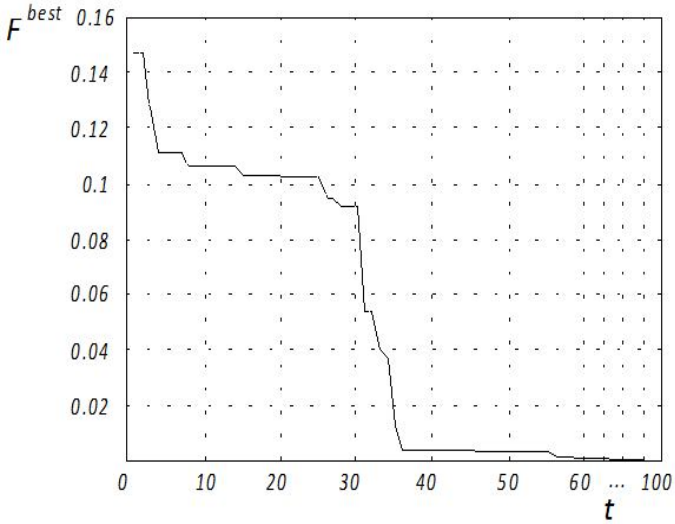


Рисунок 2 – Изменение размеров субпопуляций $|S_1|$, $|S_2|$ в процессе итераций: функция Розенброка; Δ_a = 9; n_p= 0,15; $|S|_{\min}$ = 0,2$\cdot|S|$

Эффективность ко-алгоритма особенно очевидна из сопоставления рисунков 3а, 3б, которые показывают динамику изменения значения лучшего значения целевой функции F^{best} в процессе итераций для ко-алгоритма и базового алгоритма *PSO*.



а) сходимость ко-алгоритма



б) сходимость базового алгоритма *PSO*

Рисунок 3 – Сходимость ко-алгоритма и базового алгоритма *PSO*:
функция Розенброка

Заключение

В работе предложена ко-алгоритмическая модификация алгоритма роя частиц. Выполнено исследование полученного алгоритма, показавшее, что ко-алгоритмическая гибридизация алгоритмов роя частиц может обеспечить значительное повышение эффективности базовых алгоритмов и заслуживает дальнейшего изучения.

Литература

1. Zhou Y., Pei Sh. A hybrid co-evolutionary particle swarm optimization algorithm for solving constrained engineering design problems. // China Journal of computers, 2010, V. 5, N. 6, P. 965-972.
2. Shi Yu., Krohling R. A. Co-evolutionary particle swarm optimization to solve min-max problems // Evolutionary Computation, 2002, V. 2, P. 1682-1687.
3. Карпенко А.П., Селиверстов Е.Ю. Глобальная оптимизация методом роя частиц. Обзор. // Информационные технологии, 2010, № 2, С. 25-34.
4. Процыков Г.В., Семенкин Е.С., Токмин К.А. Об эффективности коэволюционного подхода в практических задачах оптимизации. // Вестник Красноярского государственного университета, 2005, №4, С. 233-239.

АНАЛИЗ МЕТОДОВ РАСПОЗНАВАНИЯ ДИНАМИЧЕСКИХ ОБЪЕКТОВ

Марченков А.М.
Научный руководитель к.т.н. Волосатова Т.М.
МГТУ им. Н.Э. Баумана, Москва, Россия

ANALYSIS OF METHODS OF RECOGNITION DYNAMIC OBJECTS

Marchenkov A.M.
The scientific chief: Ph.D. senior lecturer Volosatova T.M.
Department RK6 BMSTU, Moscow, Russia

Аннотация
Приведен анализ методов распознавания динамических объектов. Предложено два алгоритма решения задачи распознавания. Проиллюстрированы основные моменты при реализации, предложенных алгоритмов.
Abstract
An analysis of methods for identification of dynamic objects is given. Two algorithms for solving the problem of recognition are proposed. The main points in the realization of the proposed algorithms are illustrated.

Введение

Простейшим способом обнаружения объектов движущихся на постоянном фоне является вычитание изображений. Он вполне применим, например, когда ставится задача определить нахождения объектов на ленте производственного конвейера.
На рисунке 1 приведена схема алгоритма, с помощью которого на изображении можно выделить область, где произошли изменения.

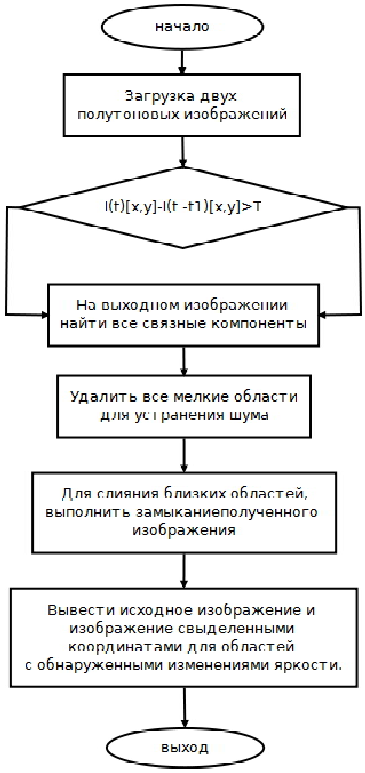


Рисунок 1 - Алгоритм вычитания двух изображений

Важным параметром в описанном выше алгоритме является порог, которому определяется разница между двумя изображениями. Если порог взять минимальным, то есть равным 0, то возникнет множество паразитных пикселей, что иллюстрируется рисунком 3. При сравнении двух изображений с различимыми человеком различиями (рис. 2), при

применении минимального порога результат остается сильно зашумленным и содержит множество ложных пикселей. Но при применении большего значения количество ошибок резко уменьшается, в тоже время становится очевидным смещение объектов на последовательности кадров.

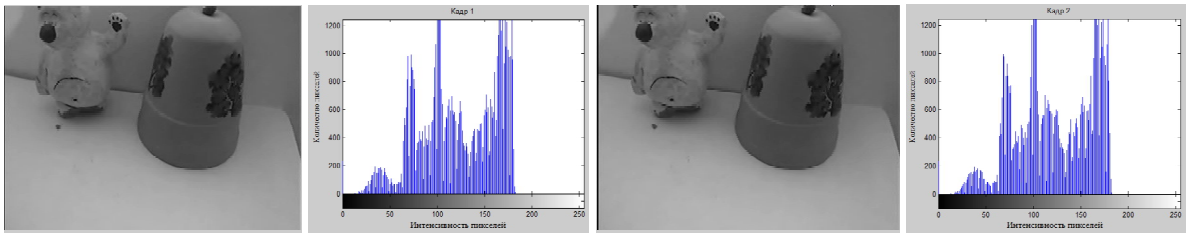


Рисунок 2 - Сравниваемые изображения с явными различиями и их

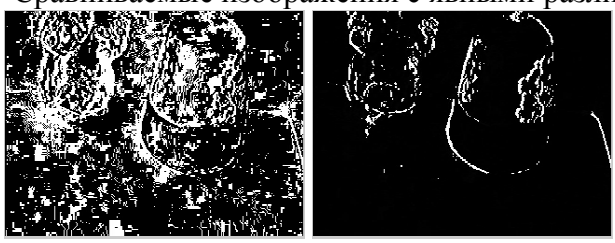


Рисунок 3 - Гистограммы результаты вычитаний с различными значениями порога (первое $T = 0$, второе $T = 5$)

На рисунке 2 исходные изображения приведены вместе с их гистограммами.

Вычисление векторов перемещения.

Полем движения называется двумерный массив двумерных векторов, представляющих движение точек трехмерной сцены. Вектора перемещения описывают смещения образов движущихся трехмерных точек. Вектор перемещения строится следующим образом: начало вектора – координаты образа трехмерной точки в момент времени *t*, конец – координаты образа той же точки в момент *t*+*delta*.

Поле движения используется как для распознавания объектов, так и для анализа их движения. На значения интенсивности пикселей изображения при вычислении векторов перемещения обычно накладывается ограничение в одной из двух форм. Можно предположить, что интенсивность образа трехмерной точки и некоторой ее окрестности в течение временного интервала ($t1$,$t2$), на котором выполняется оценка перемещения точки, остается примерно постоянной. Или же, можно предположить, что на временном интервале ($t1$,$t2$) постоянна разность интенсивностей на образах краев объекта.

Поток изображения – это поле движения, вычисленное в предположении о постоянстве интенсивности изображений в окрестностях соответствующих точек.

Для вычисления разреженного поля движения можно пользоваться парами соответствующих точек, найденных на двух изображениях для различных моментов времени $t1$ и $t1$+*delta*. Эти точки должны обладать некоторым характерным признаком, чтобы их можно было обнаружить на обоих изображениях. Например, и на полутоновых, и на цветных изображениях в качестве соответствующих точек можно выбрать угловые или граничные точки. На сегментированных полутоновых изображениях с движущимися областями постоянной формы в качестве соответствующих точек можно искать центры тяжести этих областей. Для угловых точек этот оператор вычисляет изменение яркости по вертикальным, горизонтальным и диагональным соседям пикселя *P*. Только, если минимальная из этих четырех разностей превышает некоторый порог, то пиксель считается точкой интереса (признаковой точкой). Можно использовать и другие признаковые операторы, например, текстурный оператор.

После обнаружения интересующих точек {*P*j} на изображении *I*1 для момента времени надо найти соответствующие точки изображения *I*2 для момента времени *t+delta*. Вместо выполнения аналогичных вычислений и подбора соответствующих пар точек на изображении *I*2 проще искать новые положения точек интереса, найденных на *I*1. Для этого можно пользоваться корреляционным методом. Для очередной интересующей точки *P*j с изображения *I*1 нужно взять небольшую область вокруг этой точки из *I*1 и найти для нее положение с максимальной корреляцией на изображении *I*2, предполагая, что возможное перемещение точки ограничено. Поиск точки нового положения *P*j ограничен прямоугольной областью с размерами CxR. С уменьшением размеров области ускоряется поиск и уменьшается вероятность ошибки из-за обнаружения неверной точки. Но уменьшать область имеет смысл, только если есть априорная информация о предельной скорости объектов [1]. Подробное описание развития алгоритмов слежения приведено в [3].

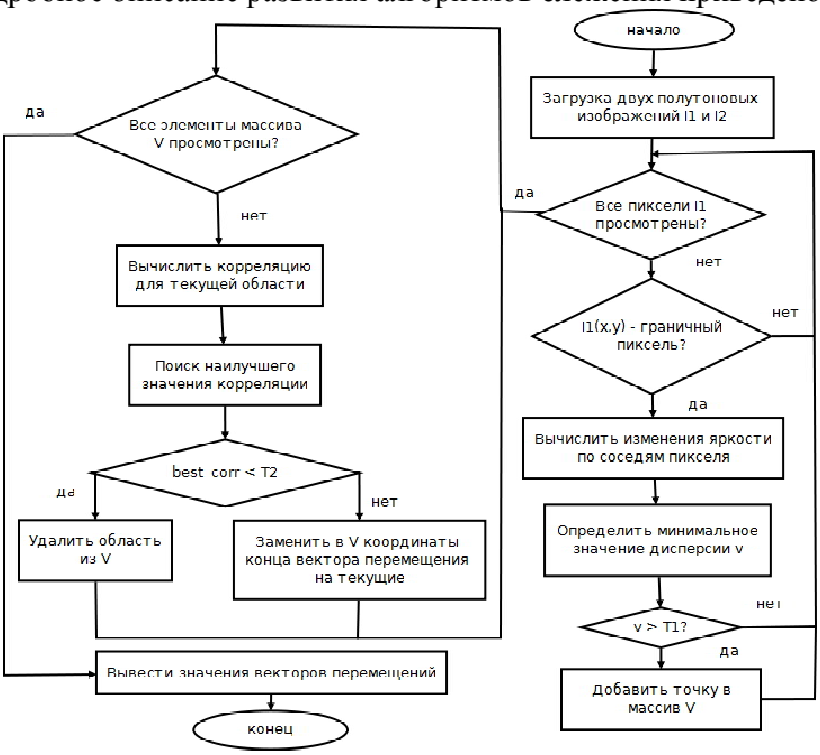


Рисунок 4 - Алгоритм построения векторов перемещения
(T_1 – яркостный порог для оператора интереса, T_2 - порог значения кросс-корреляции по окрестности точки интереса)

На рис. 4 представлен алгоритм нахождения векторов перемещений, его можно применять для циклической обработки пар кадров для непрерывного прослеживания точек интереса на видеопоследовательности с большим количеством кадров. При этом угловые точки, обнаруженные на кадре *t+delta*, можно после очередного применения алгоритма запомнить в качестве точек, обнаруженных на очередном кадре, но для вычисления кросс-корреляции лучше будет использовать новые, возможно, изменившиеся в окрестности. Подобным образом признаковые точки можно прослеживать на динамической сцене в условиях постепенного изменения окрестностей этих точек. В общем случае нужно предусмотреть действия алгоритма в случае появления новых или исчезновения старых угловых точек из-за перекрытия другими объектами [1].

Существует множество операторов для определения границ, на рисунке 5 приведены результаты их применения к одному из кадров видео ряда. Как видно из рисунка 5 наиболее точное определение границ объектов на исходном изображении дают 2 последних вида фильтров, но для решения поставленной задачи, нам не нужно определять все границы. Еще один недостаток этих фильтров в том, что происходит удвоение границы, что впоследствии

увеличит число операций в алгоритме, при его точность значительно не повысится, так будут рассматриваться окрестности смежных точек [2].

Фильтры Собеля и Превитта дают схожие результаты, поэтому в алгоритме можно использовать любой из них. Все дальнейшие действия буду производиться с изображением, полученным в результате применения фильтра Собеля.

На рисунке 6 приведены графики значений функции кросс-корреляции, вычисленные в области поиска для трех разных особых точек, выбранных произвольно.

Рисунок 5 - Результаты работы фильтров определения границ

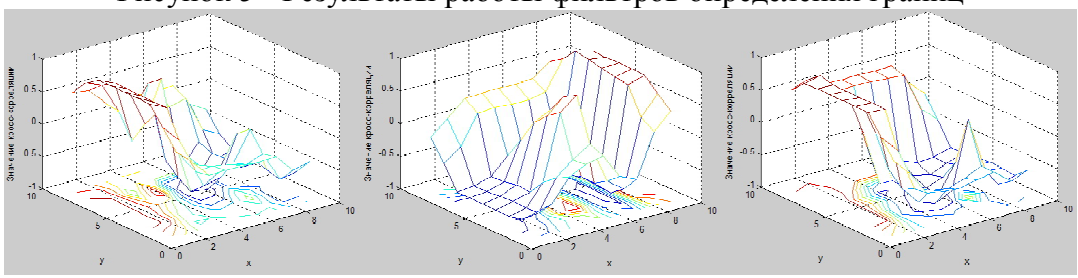


Рисунок 6 - Графики значений функции кросс-корреляции для трех произвольных точек

Во многих прикладных задачах надежность и скорость слежения можно увеличить за счет дополнительной информации. Если в каждой из m точек определены некоторые характерные признаки, то к вычислению гладкости можно добавить сравнение и согласование признаков. На основе частичной траектории до текущего момента времени t можно предсказать местоположение следующей точки траектории на кадре для момента времени $t+1$. Это позволяет существенно уменьшить время корреляционного поиска очередной точки траектории [1].

Предложенный выше алгоритм работает на видеоряде, который состоит из изображений с незначительными смещениями особых точек. Это следует из начальных предположений. Решение задачи слежения в случае поворота или масштабирования изображения описывается в [4].

Литература

1. Л. Шапиро, Дж. Стокман. Компьютерное зрение. Москва, «Бином. Лаборатория знаний» , 752с., 2006 г.
2. Р. Гонсалес, Р. Вуде – Цифровая обработка изображений. : перевод с английского под редакцией П. Н. Чочиа – «Техносфера», Москва, 2005.
3. Антон Конушин. Слежение за точечными особенностями сцены (Point feature tracking). Компьютерная графика и мультимедиа. Выпуск №1(5)/2003. http://cgm.computergraphics.ru/content/view/54.
4. Виктор Гаганов. Инвариантные алгоритмы сопоставления точечных особенностей на изображениях. Компьютерная графика и мультимедиа. Выпуск №7(1)/2009. http://cgm.computergraphics.ru/issues/issue17/invariant_features.

СОСТАВ РАСШИРЕННЫХ SCADA ДЛЯ СИСТЕМ УПРАВЛЕНИЯ ЭКСПЛУАТАЦИЕЙ СЛОЖНЫХ ТЕХНИЧЕСКИХ КОМПЛЕКСОВ

Беньков А. А.
Научный руководитель: к. т. н., доцент, Жук Д. М.
МГТУ им. Н. Э. Баумана, Москва, Россия

THE COMPOSITION OF EXPANDED SCADA FOR AUTOMATED EXPLOITATION MANAGEMENT SYSTEMS FOR COMPLEX TECHNICAL STRUCTURES

Benkov A. A.
Supervisor: Ph. D., associate professor, Zhuk D. M.
MSTU, Moscow, Russia

Аннотация
В связи с повсеместным использованием систем управления эксплуатацией (УЭ) технически сложными объектами появляется задача формализации методик построения комплексных систем УЭ. Неотъемлемой частью таких комплексов являются SCADA-системы. В этой статье рассмотрены существующие решения и основной состав современных SCADA-систем. Однако, в случае, когда объектом мониторинга и управления является сложная динамическая многопараметрическая система, средств, предоставляемых традиционными SCADA-системами становится недостаточно. На основе исследования проблематик современных систем были предложены дополнительные элементы, позволяющие расширить функционал существующих SCADA-систем для использования их в системах УЭ сложными техническими комплексами. Создание методик на основании материалов этой статьи, позволяющих разрабатывать расширенные SCADA-системы, значительно ускорит процесс разработки таких комплексов и позволит сократить финансовые затраты при их создании.

Annotation
Due to the widespread use of automated exploitation management (AEM) systems for technically complex objects the problem of the formalization of the developing complex AEM system methods appears. The Integral part of such complexes are SCADA-systems. In this article existing decisions and the basic structure of modern SCADA-systems are considered. However, in a case when object of monitoring and control is the complex dynamic multiple parameter system, the means given by traditional SCADA-systems becomes insufficiently. Based on the research of the advanced systems additional elements, which extend the functionality of existing SCADA-systems for use in the exploitation management systems for complex technical systems have been proposed. Creation of techniques on the basis of materials of this article, allowing to develop the expanded SCADA-systems, considerably will accelerate process of developing of such systems, and reduce financial costs at their creation.

Введение

Управление эксплуатацией любого технически сложного объекта было бы невозможно представить без использования SCADA-систем. Первые SCADA-системы появились еще в 70-х годах, но только к концу прошлого века они начали приобретать современный облик. Современные SCADA-системы позволяют решать типовые задачи [1] такие как:

- прием и сохранение в архивах принятой информации о контролируемых технологических параметрах от оборудования нижестоящего уровня (контроллеров и датчиков);
- обработка принятой информации в реальном масштабе времени: математическая обработка информации от «нижнего» уровня с целью получения реальной картины состояния всего технологического процесса;
- сравнение измеренных значений технологических параметров с заданными значениями и формирование сигналов управления, а также предупредительной и аварийной сигнализации;
- сохранение принятой информации в единой базе данных;
- отображение хода технологического процесса в виде мнемосхем, трендов (графиков изменения параметров во времени), индикаторов;

- хронометрирования основных технологических параметров, формирование протокола событий и архивных данных;
- обмен информацией с автоматизированной системой управления предприятием. прием команд оператора и передача их контроллерам и исполнительным механизмам;
- непосредственное автоматическое управление технологическим процессом в соответствии с заданными алгоритмами и технологическими регламентами;
- регистрация событий, связанных с контролируемым технологическим процессом и действиями персонала, ответственного за эксплуатацию и обслуживание системы;
- оповещение эксплуатационного и обслуживающего персонала об обнаруженных аварийных событиях, связанных с контролируемым технологическим процессом и функционированием программно-аппаратных средств АСУТП, с регистрацией действий персонала в аварийных ситуациях;
- формирование сводок и других отчетных документов на основе архивной информации;

Однако, в случае, когда объектом мониторинга и управления является сложная динамическая многопараметрическая система, средств, предоставляемых традиционными SCADA-системами становится недостаточно. Необходимость в расширении функционала SCADA-систем для управления сложными техническими объектами и процессами обуславливается непрерывным возрастанием сложности управляемых объектов и процессов с одновременным сокращением времени, отводимого лицам оперативно-диспетчерского персонала на анализ проблемной ситуации, идентификацию возникшего отклонения от нормального (штатного) режима функционирования объекта, поиск возможных корректирующих решений по воздействию на объект, прогнозирование ситуаций, оценку последствий принимаемых решений и, наконец, выдачу команд для отработки необходимых управляющих воздействий. Этот процесс требует много времени и высокой квалификации для того, чтобы точно и объективно оценить обстановку.

Для решения проблемы управления эксплуатацией сложными динамическими многопараметрическими системами необходимо рассмотреть существующие подходы к использованию SCADA-систем, существующие проекты, расширяющие возможности стандартных SCADA-систем, выделить функциональный состав в существующих проектах и сформулировать формализованный состав расширенных SCADA-систем.

Методы создания систем управления эксплуатацией

Как отмечалось в статье [1], системный интегратор или конечный пользователь, приступая к разработке прикладного программного обеспечения (ППО) для разрабатываемой системы управления эксплуатацией, выбирает один из следующих путей:
- программирование с использованием "традиционных" средств (традиционные языки программирования, стандартные средства отладки и пр.);
- использование существующих, готовых COTS (Commercial Off The Shelf) инструментальных проблемно-ориентированных средств.

Преимуществом программирования ППО при разработке системы управления эксплуатацией является тот факт, что в результате работы над проектом будет получено качественное, хорошо отлаженное ППО, написанное высококвалифицированным программистом специально для решения конкретных задач. Но следующую задачу по автоматизации другого объекта этот специалист вынужден решать опять практически с нуля. Процесс создания ППО для сложных распределенных систем становится недопустимо длительным, а затраты на его разработку очень высокими. Сегодня, в условиях всё более возрастающей доли ППО в затратах на создание конечной системы и, соответственно, всё большей интенсификации труда программистов, вариант с непосредственным программированием относительно привлекателен лишь для простых систем или небольших фрагментов большой системы, для которых нет стандартных решений (не написан,

например, подходящий драйвер) или они не устраивают по тем или иным причинам в принципе.

В любом случае процесс разработки собственного ППО важно упростить, сократить временные и прямые финансовые затраты на разработку ППО, минимизировать затраты труда высококлассных программистов, по возможности привлекая к разработке специалистов в области автоматизируемых процессов.

При современном уровне развития области SCADA и в связи с большим количеством реализованных проектов в этой области для минимизации временных и, в последствии, денежных затрат, более оправданным является использование уже готовых продуктов SCADA. Разработка современной SCADA-системы требует больших вложений и выполняется в длительные сроки. И именно поэтому в большинстве случаев разработчикам управляющего ППО, в частности ППО для АСУ ТП, представляется целесообразным идти по второму пути, приобретая, осваивая и адаптируя какой-либо готовый, уже испытанный универсальный инструментарий.

Проблема выбора SCADA-систем

Сегодня на российском рынке наиболее распространенными являются следующие SCADA-системы [1]:

1) In Touch (Wonderware, США);
2) FIX (Intellution, США);
3) Genesis (Iconics Co, США);
4) Citect (CI Technology, Австралия);
5) Factory Link (United States Data Co, США);
6) RealFlex (BJ Software Systems, США);
7) Sitex (Jade Software, Великобритания);
8) WinCC (Siemens, Германия);
9) Trace Mode (AdAstrA, Россия);
10) RTWin (SWD Real Time Systems, Россия);
11) Cimplicity (GE Fanuc, США);
12) RSView (Rockwell Automation, США);
13) САРГОН (НВТ - Автоматика, Россия);
14) IGSS (Seven-Technologies, Дания);
15) Image (Технолинк, Россия)
16) MIK$Sys (МИФИ, Россия).

В качестве альтернативы платным системам на сегодняшний день в линейке SCADA представлены так же проекты с открытым кодом, которые могут быть интересны в связи с открытостью исходных кодов и дешевизной внедрения таких решений. Основные системы - это:

1) FreeScada - SCADA система с открытым кодом для MS Windows, которая предоставляет конечному пользователю средства визуализации и интерактивного управления любым техническим процессом.Система разработана с использованием языка C#;
2) SZARP - полнофункциональная SCADA система с открытым кодом для систем Linux, клиентское приложение которой доступно как для Linux, так и для Windows. SZARP первоначально был разработан для управления муниципальными отопительными системами. Эта система содержит встроеный скриптовый язык Lua;
3) Argos;
4) OpenSCADA.

При таком многообразии SCADA-систем на нашем рынке возникает проблема выбора актуальной SCADA-системы для осуществления того или иного проекта автоматизации. Этот выбор представляет собой довольно сложную задачу, аналогичную задаче по принятию решения в условиях многокритериальности: большое количество показателей, критериев и

мало количественных оценок для принятия однозначного решения. Оценка SCADA-систем для использования в каждом конкретном проекте может быть осуществлена на основании трех групп показателей: технические, эксплуатационные и экономические.

Состав расширенных SCADA-системы

Современные SCADA-системы имеют схожие возможности и принципы функционирования, которые позволяют решать типовые задачи такие как: диспетчерский мониторинг и сбор данных о протекании технологического процесса, управление при наличии четких алгоритмов и полной формализованной модели объекта управления. Однако, при большом количестве входной информации, требующей анализа, и не всегда достаточной квалификации персонала-операторов могут возникать ошибки, приводящие к аварийным ситуациям на объекте. Анализ мирового опыта показывает, что при совершенствовании технологических процессов и автоматизации процесса принятия решений наиболее перспективным является использование информационных систем, основанных на знаниях, формализуемых в рамках технологии искусственного интеллекта и опыте высококвалифицированных специалистов, накапливаемом в базах знаний экспертных систем. [6]

Актуальной задачей при построении автоматизированных систем реального времени является перенос функций диспетчера по анализу данных, прогнозированию ситуаций и принятию соответствующих решений на компоненты интеллектуальных систем поддержки принятия и исполнения решений (СППИР) [2]. Концепция систем поддержки принятия и исполнения решений включает целый ряд средств, объединенных общей целью — способствовать принятию и реализации рациональных и эффективных управленческих решений.

В современных реалиях все усложняющихся технических систем управления эксплуатацией диспетчерский контроль становится все более сложной задачей. С другой стороны уровень развития техники и технологий управления эксплуатацией предлагает все более совершенные инструменты для решения этой задачи. В ряде проектов уже сейчас в составе автоматизированных систем управления присутствуют элементы, расширяющие основной функционал SCADA систем. В качестве примеров можно рассмотреть комплексы СПУРТ компании ATGS и Мрия компании Unis trans, в которых были добавлены функции интеллектуальной системы помощи в принятии решений [4,5]. Обобщая информацию, полученную в ходе разработки этого направления, и опыт в разработке сложных технических комплексов можно выделить следующие элементы, позволяющие перевести процесс управления технологическими процессами и техническими комплексами на новый уровень:

- Интеллектуальная система помощи в принятии решений при эксплуатации (ИСППР)
- Интеллектуальная система выявления неисправностей
- Адаптивная система контроля и планирования технического обслуживания и ремонта (ТОиР)
- Система автоматизированного восстановления функционирования в аварийных ситуациях (комплекс мер и правил в построении комплексов)

В первую очередь необходимо рассмотреть систему помощи в принятии решений. Сложность и большой объем протекающих процессов при эксплуатации сложных технических комплексов приводит к большому потоку анализируемых данных. В связи с этим на пульт оператора приходит большой объем информации, требующей обработки и анализа. При этом возникающие ошибки в анализе, несвоевременное реагирование на поступающие сигналы и неправильные действия по устранению возникающих ситуаций приводят к материальным затратам, потерям времени, возможным человеческим жертвам. При использовании системы ИСППР возможно существенно сократить объем информации предоставляющейся для контроля процессов на объекте управления. Большое количество данных может обрабатываться автоматической системой без участия человека на основе

экспертных данных, введенных в систему. При возникновении внештатной ситуации такая система генерирует набор действий, позволяющих решить возникшие проблемы, на основе тех же экспертных данных. Такой подход позволяет сократить объем анализируемой информации оператором в процессе эксплуатации, сократить время реагирования при возникновении внештатной ситуации и сократить ошибки при вмешательстве операторов в работу системы.

И, наконец, любой сложный технический комплекс часто оборудуется дублирующими системами, кроме того, отдельные элементы комплекса часто выполняют схожие функции и могут ограниченно заменить функционально смежные элементы. В таких комплексах в аварийных ситуациях для предотвращения простоя оборудования и, что более важно, предотвращения возможных негативных последствий такой аварии важно иметь возможность поддерживать ограниченный функционал комплекса до восстановления поврежденных элементов. Реализация такого подхода в рамках системы управления позволяет сократить эксплуатационные расходы и издержки в связи с поломками комплекса. Однако, для реализации этой концепции в полной мере необходимо не только расширять состав системы управления эксплуатацией (СУЭ), но и вносить ряд правил и мер по построению технических комплексов, требующих использования СУЭ.

Заключение

В заключение необходимо отметить, что описанные системы востребованы во многих областях, использующих системы автоматизации управления эксплуатацией. Такой подход позволяет сократить затраты при эксплуатации многих сложных технических комплексов. Однако рассмотренные существующие решения не позволяют в полной мере воспользоваться всеми преимуществами использования систем автоматического контроля. Разработка более сложных систем, позволяющих существенно расширить существующий функционал, происходит каждый раз фактически с нуля. Такой подход приводит к увеличению временных и финансовых затрат при построении комплексных систем, в которых SCADA является только составной частью. В связи с этим возникает необходимость создания формализованной методики построения такого рода систем для сокращения времени на разработку и как следствие предотвращение существенных финансовых затрат.

Литература

1. Куцевич Н. SCADA-системы, или муки выбора [Электронный ресурс] / Надежда Куцевич. - М. - Режим доступа : http://www.asutp.ru, свободный. - Загл. с экрана.
2. Береза А.С. Концепция развития функций SCADA-системы TRACE MODE на основе технологии экспертных систем принятия и исполнения решений / А.С. Береза [и др.] // Журнал «ИСУП». - 2005. - №1 (5). - С. 30-39.
3. Шерешевский Л.А. Вопросы открытости современных SCADA-систем на примере SIMATIC WinCC / Л.А. Шерешевский // Мир компьютерной автоматизации. – 2003. – №4. – С. 86-88.
4. Радкевич В.В. Использование некоторых SCADA-систем в газовой отрасли / В.В. Радкевич, А.В. Реунов // Журнал «ИСУП». - 2007. - №2 (14). - С. 1-10.
5. Зельдин Ю. М. Концепция построения современной информационно-управляющей системы в диспетчерском центре газотранспортного общества ОАО «Газпром» / Ю. М. Зельдин, А. А. Ковалев // Информационный сайт o-asutp.ru, 2011. - Режим доступа: http://o-asutp.ru/staty.php?c=1312873033&s=1318485620, свободный, - Загл. с экрана.
6. Подлипский О. К. Построение баз знаний группой экспертов / О. К. Подлипский // Журнал «Компьютерные исследования и моделирование». - 2010. - т. 2, №1. - С. 3 - 11
7. Домнич В. С., Иващенко В. А. Построение баз знаний для поиска причин аварийных ситуаций при формировании листового стекла / Управление большими системами. Выпуск 33. М.: ИПУ РАН, 2011. С.218-232

НАУЧНЫЕ ВЫЧИСЛЕНИЯ В ОБЛАКАХ

Романова А.О.
Научный руководитель д. ф.-м. наук, профессор, Карпенко А.П.
МГТУ имени Н.Э.Баумана, Москва, Россия

SCIENTIFIC CLOUD COMPUTING

Anastasia Romanova
Supervisor: Dr., Prof. Karpenko A.P.
MSTU, Moscow, Russia

Аннотация
Рассмотрены преимущества облачных вычислений, дана краткая характеристика облачной парадигмы. Сформулированы недостатки использования облаков в научных вычислениях и предложен метод их решения. Приведены результаты вычислительного эксперимента, выполненного в облаке.

Abstract
This article discusses the advantages of cloud computing, summary of the cloud paradigm is given. Disadvantages of using cloud computing for scientific computing are defined and the method of solving them is suggested. The text presents the results of numerical experiments in cloud.

Введение

Облачные или рассеянные вычисления представляют собой технологию обработки данных, в которой компьютерные ресурсы и мощности предоставляются пользователям как Интернет-сервисы [1]. Пользователь не должен заботиться об инфраструктуре, операционной системе и собственно программном обеспечении, к которому предоставляется доступ. Абстракция ресурсов происходит на самом высоком уровне: оборудование объединяется с виртуализированными приложениями, позволяя ИТ-администратору предоставлять информационные технологии в качестве услуги для потребителей. Таким образом, можно создать каталог ИТ-услуг, и конечный пользователь будет выбирать то, что ему нужно, исходя из его запросов на аппаратное и программное обеспечение.

Главными преимуществами в использовании облаков в вычислениях являются:

- быстрое изменение конфигурации выделенных ресурсов;
- плата только за использование ресурсов.

По сравнению с персональным компьютером, вычислительная мощь, доступная пользователю облаков, практически ограничена лишь размером облака, то есть общим числом серверов. Пользователи могут запускать более сложные задачи, с большим количеством необходимой памяти и места для хранения данных, тогда, когда это необходимо. Иными словами, пользователи могут при желании легко и дешево поработать с суперкомпьютером без каких-либо фактических приобретений.

Следует отметить так же некоторые недостатки такой модели:

- необходимость постоянного соединения с сетью Интернет;
- риск кражи или потери данных.

В настоящее время все программные приложения, обеспечивающие вычисления в облаках, по сути, реализуют новую парадигму взаимодействия между IT-инфраструктурой и бизнесом и не поддерживают научные вычисления. В данной работе исследуется возможность использования облачной модели в научных вычислениях.

Постановка и методы решения задачи

Облака предоставляют широкий спектр возможностей. Но возникает вопрос – какую конфигурацию серверов и какое их число следует заказывать для решения стоящей перед пользователем задачи. Современное облачное программное обеспечение не решает данной проблемы, и решение ее в рамках облачной парадигмы в настоящее время не предполагается (исходя из ориентации этой парадигмы на бизнес и бизнес-решения).

В работе рассматривается две взаимосвязанных задачи:

- исходя из спецификации задачи пользователя, разработать методы, алгоритмы и соответствующее программное обеспечение для выбора оптимального числа и параметров виртуальных серверов (архитектуры облака);
- разработать методы, алгоритмы и программное обеспечение для выбора оптимальных параметров алгоритма решаемой задачи для фиксированной архитектуры облака.

Решение указанных выше задач требует разработки следующих моделей и методов:

- модель архитектуры облака,
- модель алгоритма, с помощью которого решается задача пользователя,
- метод оптимального отображения алгоритма на архитектуру облака.

*Модель облака***.** В настоящее время нет оснований полагать, что в облаках будет предоставляться доступ к специализированным вычислителям, например, типа графических процессорных устройств. Поэтому мы исходим из того, что все возможные виртуальные серверы являются универсальными (концепция гомогенного облака) и каждый из них может решить любую из задач пользователя.

Физические сервера облака находятся, как правило, в центрах обработки данных (ЦОД), связаны между собой «широкими» каналами связи и число их не слишком велико. На этом основании мы полагаем, что все виртуальные серверы облака связаны между собой напрямую, т.е. граф их связей представляет собой клику.

Виртуальный сервер $P_i, i \in [1:N]$ определяется следующими параметрами:

N_i^{CPU} – число процессоров в сервере;

p_i - производительность сервера, MFLOP/c;

m_i^{CPU} – стоимость использования одного процессора в час;

m_i^{RAM} – стоимость использования заказанного объема оперативной памяти в час;

m_i^{HDD} – цена использования заказанного объема памяти на жестком диске в час

Отметим следующее обстоятельство. Реальная производительность виртуального сервера может несколько отличаться от заданной по следующим причинам:

- неизвестна фактическая загрузка физического сервера, на котором функционирует данный виртуальный сервер;
- виртуальный сервер может находиться в состоянии миграции (задача штатного балансировщика облака перераспределения виртуальных серверов по физическим серверам).

Модель алгоритма. Полагаем, что задача сводится к циклическому решению совокупности подзадач $T_j, j \in [1:n]$ (рисунок 1). Вычислительная сложность $q_j = q_j(t)$ (число операции, необходимых для решения подзадачи T_j на итерации t) полагается априори неизвестной, различной на различных итерациях и многократно превышающей вычислительную сложность подзадач T_0, T_Σ. В таком виде можно, например, представить модели популяционных алгоритмов решения задачи глобальной оптимизации таких, как алгоритм роя частиц, колонии пчел и т.д.

Метод оптимального отображения. Отображение подзадач $T_j, j \in [1:n]$ на архитектуру облака задаем отображающей $(N \times n)$-булевой матрицей $X = X(t)$, в которой $x_{i,j} = 1$ означает, что подзадача T_j должна быть выполнена на виртуальном сервере P_i, а $x_{i,j} = 0$ - что указанная подзадача не должна выполняться на сервере P_i; $i \in [1:N]$

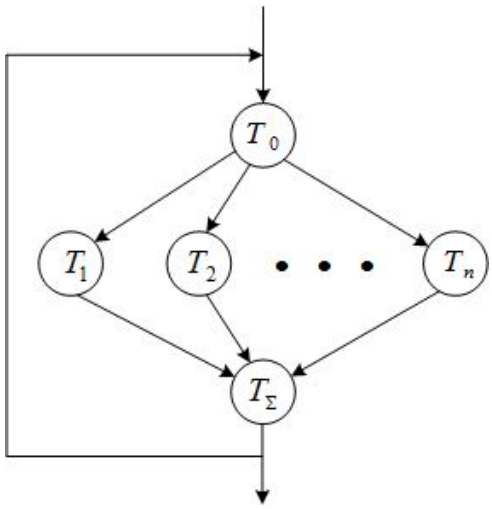


Рисунок 1 — Модель задачи

Полагаем, что после решения задачи T_j на том из процессоров виртуального сервера, которому она назначена на выполнение на итерации t, становится известной оценка $q_j(t)$ текущей вычислительной сложности этой задачи. Полагаем также, что эта вычислительная сложность «не слишком быстро» меняется в процессе итераций.

В качестве критерия оптимальности отображения используем суммарную стоимость решения задачи

$$M(\mathbf{X}) = \sum_{t=1}^{\hat{t}} m(X(t)) = \sum_{t=1}^{\hat{t}} \sum_{i=1}^{N} \left(N_i^{CPU} \; m_i^{CPU} + m_i^{RAM} + m_i^{HDD} \right) h(X(t)) \;,$$

где $\mathbf{X} = (X(1), X(2), ..., X(\hat{t}))$ - множество всех отображающих матриц; $m(X(t))$, $h(X(t))$ - стоимость и время решения задачи на итерации t соответственно; $\hat{t}$ - общее число итераций. Время $h(X(t))$ определяется с помощью таймера программного обеспечения облака.

На каждой из итераций $t \in [1:\hat{t}]$ задачу оптимального отображения ставим как задачу балансировки загрузки

$$\max_{i \in [1:N]} \sum_{j=1}^{n} p_i \; q_j(t-1) \; x_{i,j}(t) \underset{X(t)}{\rightarrow} \min \;, \qquad (1)$$

т.е. как задачу отыскания такой отображающей матрицы $X^*(t)$, которая на данной итерации максимально равномерно загружает все виртуальные сервера. Заметим, что сумма в формуле (1) представляет собой оценку времени решения виртуальным сервером P_i всех распределенных ему подзадач.

Задачу балансировки загрузки (1) решаем жадным алгоритмом упаковки в контейнеры [2], где контейнеры формализуют производительности виртуальных серверов, а объекты упаковки - вычислительные сложности подзадач.

Вычислительный эксперимент

Был проведен эксперимент по решению в облаке задачи многокритериальной оптимизации методом нейронечеткой аппроксимации функции предпочтений лица, принимающего решения [3]. Исследовалась зависимость достигнутого ускорения вычислений от числа виртуальных серверов и вычислительная сложность целевой функции. Некоторые результаты эксперимента иллюстрирует рисунок 2. Вычислительный эксперимент выполнен в облаке, разработанном кампанией *Parallels* [4]. В качестве коммуникационной библиотеки использована известная библиотека *MPI*.

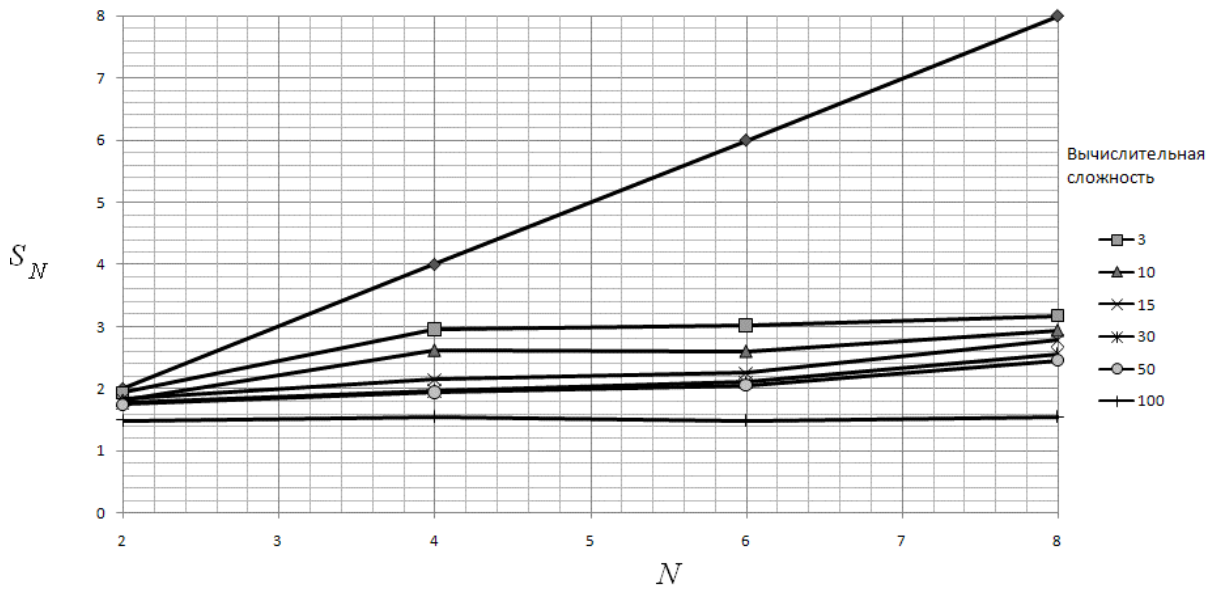


Рисунок 2 — Ускорение S_N в функции числа виртуальных серверов

Под вычислительной сложностью понимаем число стартовых точек используемого метода мультистарта.

Заключение

Результаты работы показывают, что

- облачная среда может быть адаптирована для ресурсоемких научных вычислений,
  - приемлемое ускорение может быть получено только при высокой вычислительной сложности подзадач,
  - для оптимизации научных вычислений в облаках необходима разработка специальных методов, алгоритмов и программного обеспечения,
  - предложенный в работе подход может быть положен в основу разработки указанного программного обеспечения.

Литература

1. Mell P., Grance T. The NIST Definition of Cloud Computing. Recommendations of the National Institute of Standards and Technology. NIST (20 October 2011).
2. Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К. Жадные алгоритмы // Алгоритмы: построение и анализ / Под ред. И. В. Красикова - 2-е изд.- М.: Вильямс, 2005.- 1296 с.
3. Карпенко А.П., Мухлисуллина Д.Т., Овчинников В.А. Нейросетевая аппроксимация функции предпочтений лица, принимающего решения, в задаче многокритериальной оптимизации // Информационные технологии, 2010, №10, с.2-9.
4. Parallels Automation for Cloud Infrastructure (PACI)

СОЗДАНИЕ И ИСПОЛЬЗОВАНИЕ БАЗЫ ДАННЫХ ПАТЕНТОВ ПО МИКРО И НАНОТЕХНОЛОГИЯМ

Тихомиров Д. В.
Научный руководитель : к.т.н, доцент, Резчикова Е.В.
МГТУ им. Н. Э. Баумана, Москва, Россия

CREATING AND EXPLOITING DATABASE OF PATENTS ON MICRO AND NANOTECHNOLOGIES

D. Tikhomirov
Supervisor*: Docent, Rezchikova E.V.*
Moscow State Technical University n.a. N.E. Bauman, Moscow, Russia

Аннотация
Множество методов было разработано для учета достижений технологического прогресса, и одним из них является анализ патентной информации. Методы визуализации считаются эффективными для представления массивов патентной информации и результатов ее анализа. Однако, предлагаемые настоящее время методы визуализации для патентного анализа, патентные карты, имеют некоторые недостатки. Поэтому в данной статье предлагается альтернативный метод визуализации. При кластеризации результатов, формируют семантическую сеть ключевых слов с учетом даты внесения. А потом формируют патентую карту путем перестановки каждого ключевого слова семантической сети в соответствии с его ранней датой внесения и частотой его обнаружения в патентных документах. Предложенный в статье подход позволяет создавать карту патентов, которая содержит, одновременно, структурированную и не структурированную информацию из патентного документа при помощи алгоритма интеллектуального поиска по базе данных патентной информации, который позволяет осуществлять поиск минимум по шести параметрам, вносимых в патентные документы, с дальнейшим отображением при помощи веб-интерфейса. Тем самым создается возможность визуализации четкого представления о патентной информации в наиболее доступной форме, что позволяет пользователям сформировать представление о существующих запатентованных достижениях в области микро и нано технологий и прогнозировать их тенденции в будущем.

Abstract
Many methods have been developed to recognize those progresses of technologies, and one of them is to analyze patent information. And visualization methods are considered to be proper for representing patent information and its analysis results. However, current visualization methods for patent analysis patent maps have some drawbacks. Therefore, we propose an alternative visualization method in this paper. With collected keywords from patent documents of a target technology field, we cluster patent documents by the k-Means algorithm. With the clustering results, we form a semantic network of keywords without respect of filing dates. And then we build up a patent map by rearranging each keyword node of the semantic network according to its earliest filing date and frequency in patent documents. Our approach contributes to establishing a patent map which considers both structured and unstructured items of a patent document. Besides, differently from previous visualization methods for patent analysis, ours is based on forming a semantic network of keywords from patent documents. And thereby it visualizes a clear overview of patent information in a more comprehensible way. And as a result of those contributions, it enables us to understand advances of emerging technologies and forecast its trend in the future.

Введение

Актуальная задача заключается в учете технологических тенденций не только для избегания ненужных инвестиций, но и для получения источника для технологического развития.

Так, многие методы были разработаны для учета достижений технологического прогресса, и одним из них является анализ патентной информации. Тем не менее, не для специалистов трудно анализировать патентную информации, потому что источники патентной информации содержат большое количество специальной технической терминологии. Следовательно, патентная информация должна быть преобразована во что-то более простое и доступное для понимания.

Методы визуализации считаются пригодными для представления патентной информации и результатов ее анализа. Как правило, различные методы визуализации - это один из лучших способов анализа и усвоения данных, потому что методы графического представления часто дают лучшие результаты по сравнению с другими обычными методами.

Это особенно важно для руководителей, которые принимают решения для инвестирования технологии. Методы визуализации более обозримы и интуитивно понятны по сравнению с обычными способами, такими как текстовый, табличный, а также в виде списка.

Эти методы визуализации для патентного анализа в целом называются патентными картами. Патентные карты - это визуализированное выражение общего числа результатов патентного анализа, созданные для восприятия общей патентной информации легко и эффективно.

Как правило, патентный документ содержит множество элементов для анализа, которые могут классифицироваться в виде структурированных и неструктурированных групп элементов. Структурированные элементы - одинаковые по смыслу и формату во всем патентном документе, такие как номер патента, дата подачи, страна, приоритет и т.д..

С другой стороны, неструктурированные элементы – это произвольные и различные по длине и содержанию наборы символов в патентном документе, такие как названия, аннотации, формулы или описания изобретений.

Таким образом, современные методы визуализации для патентного анализа основаны на составлении карт патентной информации с результатами их анализа.

Однако в настоящее время патентные карты имеют определенные недостатки. Во-первых, большинство из них построены на основе времени, на основе рейтинга или матриц, которые содержат лишь один аспект между структурированными элементами или неструктурированных элементами каждого патентного документа.

Очевидно, что системный и сбалансированный подход должен предоставлять собой общую структуру патентной информации более эффективно. Во-вторых, карты являются сложными комплексами патентных документов, хотя используют разные методы.

Поэтому при использовании они делают результаты патентного анализа непонятными и неясными для анализатора.

Следовательно, эти виды патентных карт не дают достаточного интуитивного понимания в выбранной области технологии. И это третий недостаток, который можно отметить. Следовательно, для развивающихся технологий, таких как микро и нанотехнологии, основным становится выявление их преимуществ и оценка развития в будущем.

Таким образом, предлагается альтернативный метод визуализации для патентного анализа, для того, чтобы преодолеть недостатки текущих патентных карт. Предлагаемый подход способствует формированию патентных карт, которые содержат как структурированные, так и неструктурированные элементы патентного документа. Это достигается при помощи алгоритма интеллектуального поиска по базе данных патентной информации, позволяющего осуществлять поиск минимум по шести параметрам, которые вносятся в патентные документы, с дальнейшим отображением при помощи веб-интерфейса. Кроме того, в отличие от ранее описанных методов визуализации патентного анализа, предлагаемый метод основан на формировании списка схожих патентов при помощи ключевых слов в алгоритме поиска из патентных документов. И тем самым он дает четкое представление о патентной информации в более доступной форме.

И в результате этого анализа, он позволяет получить представление о современных технологиях при прогнозировании тенденции высоких технологий в будущем.

Обзор существующих методов анализа

Непосредственно переходя к исследованиям, с использованием патентной информации, можно выделить два главных направления. Одним из них является

исследование способов визуализации патентной информации и ее аналитических результатов.

В данной работе уделено внимание рассмотрению и формированию области для исследования, использования патентной информации из существующих патентов.

На основе знаний о патентах, Японское патентное бюро занимается производством и обеспечивая более 200 видов патентных карт для нескольких технологий, начиная с 1997 года (JIII, 2000). Кроме того, во многих других странах, таких как Корея (Ryoo & Kim, 2005), Италии (Camus & Brancaleon, 2003; Фаттори, Pedrazzi и Turra, 2003) и США (Моррис, DeYong, By, Салман, и Yemenu, 2002) также предоставляют различные виды патентных карты.

В настоящее время, большинство видов патентного анализа используют патентные цитаты для представления наиболее значимых отношений в патентной информации. Но известно, что патентное цитирование имеет несколько серьезных недостатков. По словам Yoon и Park (2004) , есть четыре основных недостатка патентного цитирования.

Во-первых, трудно охватить общие отношения между патентными документами. Во-вторых, связанные с первым пунктом проблемы, возможности анализа и информативность данных ограничены. В-третьих, цитата не создает возможности учитывать внутренние связи между патентными документами. Наконец, патентное цитирование - это трудоемкая задача, которая использует выборочный поиск по патентному документу.

В связи с этим, Yoon & Park (2004) предложили альтернативный метод патентного анализа, называемый сетевым. Было проведено огромное количество исследований для поиска оптимального метода патентного анализа. Были разработаны нейро-сетевые методы для анализа научно-технической информации (статьи, патенты), а также оказания помощи пользователю в выполнении сложного процесса анализа большого количества подобной информации фирмами Lamirel, Shehabi, Hoffmann, and Francois (2002). На основе текстовых методов анализа Tseng, Wang, Juang, and Lin (2005) создали реальную патентую карту для важной области технологии: углеродных нано-трубок.

Другое основное направление - это патентная классификация. В Black and Ciccolo (2004) , была разработана технология машинной классификации текстов, позволяющая автоматически различать патенты, касающиеся биотехнологий и те, которые к ним не относятся. Fall, ToËrcsvar ri, Fiervet, and Karetka (2004) представили результаты применения различных алгоритмов обучения компьютеров для экспертных систем при выполнении задачи патентной классификации на Немецком языке.

Методология

Предлагаемый метод визуализации патентной информации сводится к следующему. При помощи параметров запроса (№ патента, ключевое слово, индекс МПК, авторы, фирмы, страна, дата регистрации) для выбранной технологической области, производится кластеризация патентных документов благодаря алгоритму интеллектуального поиска. Далее следует процесс обработки параметров запроса при помощи алгоритма поиска по БД патентной информации. Затем происходит формирование дерева патентных документов в базе данных, содержание которых удовлетворяет параметрам запроса. Последовательность выполняемых операций представлена на Рис. 1.

Рисунок 1 - Последовательность операций поиска

Построенное дерево, содержащее результаты запроса, выводится в веб интерфейс, разработанный для удобства работы с БД патентной информации.

Параметры поиска

Рассматриваемый метод визуализации патентной информации начинается с введения пользователем параметров поиска по БД патентной информации. Разрабатываемый алгоритм позволяет производить поиск по БД патентной информации минимум по 7 параметрам:

- Нумерационный – номер зарегистрированного патента,
- Предметный – ключевое слово(а), задаваемое(ые) для поиска в патентном документе,
- МПК – индекс международной патентной классификации,
- Авторский – автор(ы) и/или заявители зарегистрированного патента,
- Фирменный – фирма-обладатель патента,
- По странам – страна, где зарегистрирован патент,
- Временной – дата регистрации патента (приоритет).

В результате введения параметров поиска, они передаются для обработки в алгоритм интеллектуального поиска по БД патентной информации.

Обработка запроса алгоритмом поиска

Для наиболее точного поиска по БД патентной документации следует ввести как можно больше параметров поиска. Работа алгоритма поиска по БД патентной информации представлена на рис.2.

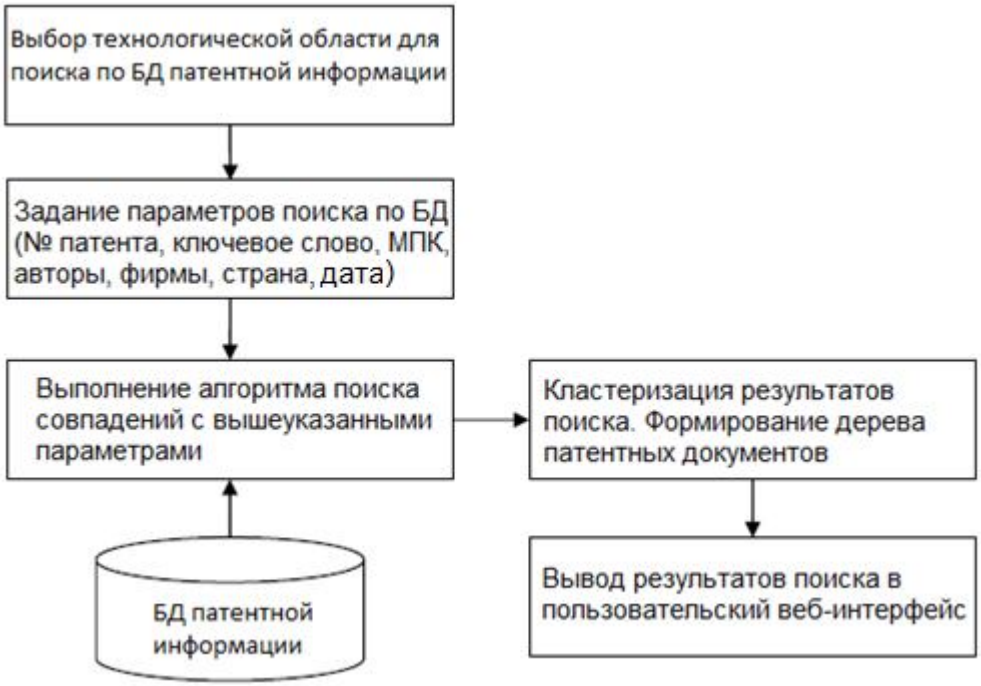


Рисунок 2 – Работа алгоритма поиска

Вначале, выбирается технологическая область, в которой необходимо произвести поиск патентной документации. Затем задаются параметры поиска запроса (№ патента, ключевое слово, индекс МПК, авторы, фирмы, страна), чем больше, тем точнее будет поиск. Далее производится поиск по документам, содержащимся в БД патентной информации, осуществляется выборка патентной документации, удовлетворяющей параметрам поиска. После производится кластеризация патентных документов, т.е. автоматическое выявление групп семантически похожих документов среди заданного фиксированного множества патентных документов. Последним шагом является вывод дерева патентных документов, удовлетворяющих параметрам поиска в пользовательский веб-интерфейс.

Формирование дерева патентных документов

На следующем шаге для кластеризованных патентных документов формируется дерево, которое позволяет видеть непосредственную группировку патентных документов по двум параметрам: ключевые слова, дата регистрации патента (см.Рис.3).

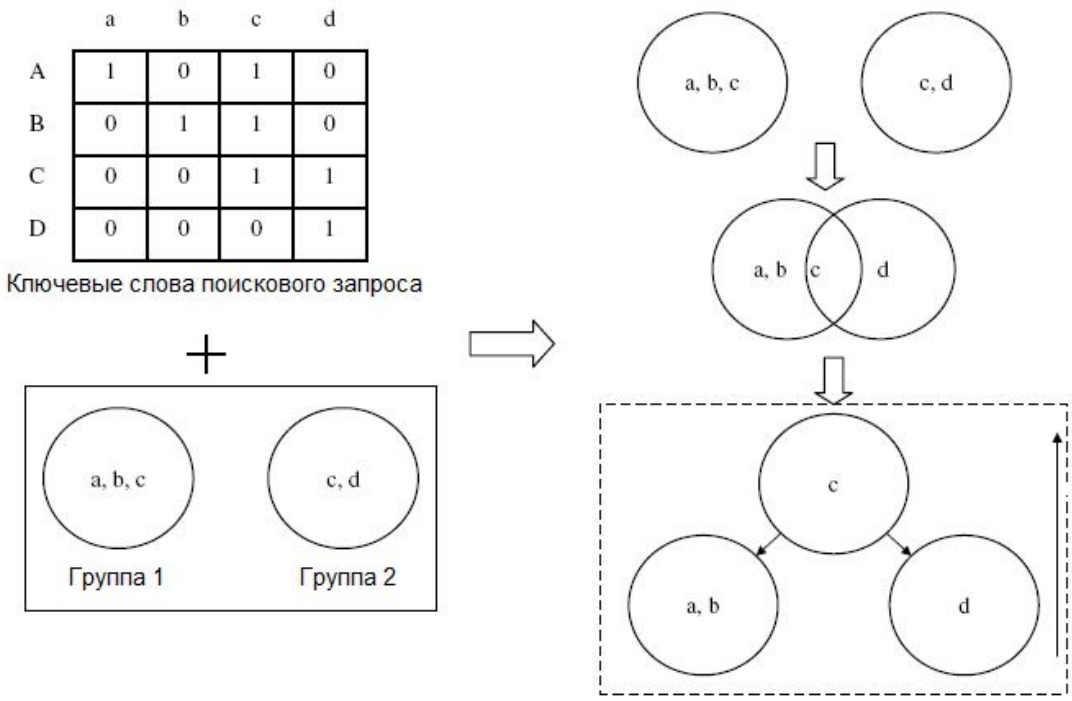


Рисунок 3 – Формирование дерева патентных документов по ключевым словам

Например, предположим, что патентные документы "A" и "B" относятся к группе 1. В соответствии с матрицей в рис.3 в патентном документе 'A' есть ключевые слова «a» и «c». А патентный документ "B" содержит ключевые слова «b» и «c». Затем для 1-й группы имеем набор из трех ключевых слов 'a', ‘b' и 'c'.

Таким образом, исследуются ключевые слова для каждой группы и, используя список ключевых слов для каждой группы, объединяются патентные документы по признаку наличия ключевых слов. Соответственно, для 1-й группы получают ключевые слова "a", "b" и "c". С другой стороны, для 2-й группы получают ключевые слова "c" и "d". Далее видно, что только одна группа имеет ключевое слово "b" и, следовательно, отношения между двумя группами можно представить в виде трех узлов: (a, b), (c) и (d). Здесь общий узел выше, чем другие, поэтому стрелки связи идут от (c) к (a, b) и (d) – рис.4.

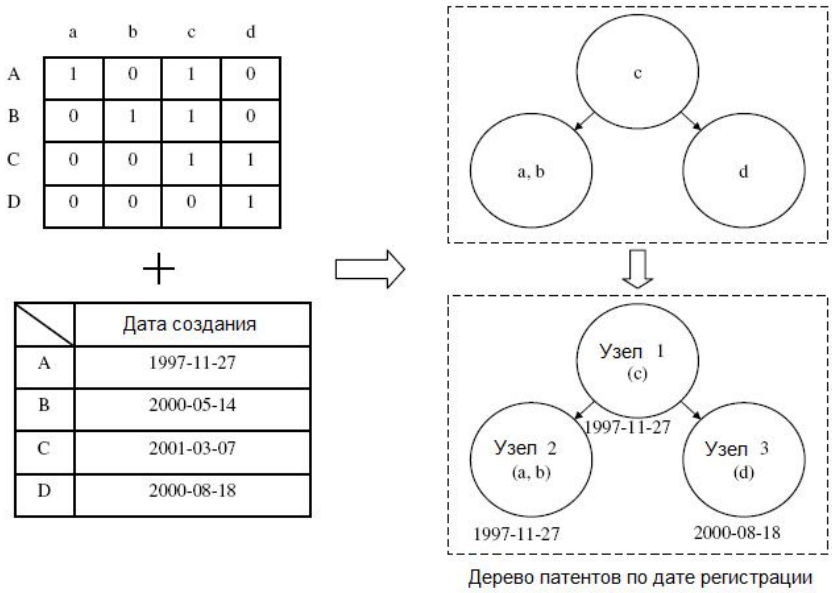


Рисунок 4 – Формирование дерева патентных документов по дате регистрации

Изменяя количество и значение ключевых слов, можно легко изменять вид дерева патентной документации. В результате получают дерево с n-узлами.

Далее для каждого узла проверяются даты регистрации патентных документов. Дата регистрации для каждого узла – это наиболее ранняя дата регистрации патентного документа, которые имеют ключевые слова данного узла. На рис.4 представлено формирование дерева патентных документов по дате регистрации.

Предположим, что сформировано дерево патентных документов, как показано на рис.3. Необходимо рассмотреть дату регистрации патентных документов для каждого узла в дереве. Дата регистрации для каждого узла – это наиболее ранняя дата регистрации патентного документа, которые имеют ключевые слова данного узла. Например, на рис 4. узел 2 состоит из ключевых слов "a" и "b". И если "a" относится к патентному документу «A», а “b” , в свою очередь, принадлежит патентному документу "B", то датой регистрации для данного узла будет наиболее ранняя дата регистрации патентных документов среди документов “A” и “B”. Таким образом, дата регистрации для узла 2 является '1997-11-27'. Кроме того, датой регистрации для узла 3 является '2000-08-18’.

Таким образом, добавляют даты регистрации патентных документов для каждого узла в дерево, сформированное при помощи ключевых слов. При помощи этих двух шагов, сформированное дерево имеет как структурированные элементы – даты регистрации, так и неструктурированные элементы - ключевые слова.

Отображение результатов поиска

Вывод сформированного дерева патентных документов, удовлетворяющих заданным параметрам поиска, производится в виде отображения данных в пользовательском интерфейсе, который позволит легко получать необходимые сведения о выбранном патентном документе, производить просмотр выбранных документы.

Заключение

В данной работе предложен альтернативный метод визуализации для патентного анализа, лишенный недостатков существующих методов формирования патентных карт. По сравнению с другими методами, описанными в разделе 2, разработанный подход позволяет работать как со структурированными элементами, так и с неструктурированными элементами патентных документов. Таким образом, можно получить четкое представление о патентной информации в наиболее доступной форме. С помощью пользовательского интерфейса, можно определить, какие виды патентов имелись ранее, какие появились недавно, и как эти патенты были объединены и дифференцированы с течением времени.

Кроме того, предлагаемый способ визуализации патентной информации позволяет получить полное представление о новых технологиях. Также он позволяет получить достаточное представление о существующих патентах в выбранной технологической области, для избегания ненужных инвестиций и достоверного поиска информации для патентования новых технических решений.

Литература

1 Y. G. Kim, J. H. Suh, S.C. Park. Visualization of patent analysis for emerging technology. Elsevier, 2008, 9

РАЗРАБОТКА МЕТОДИКИ КАЛЕНДАРНОГО ПЛАНИРОВАНИЯ ПРОИЗВОДСТВЕННЫХ ПРОЦЕССОВ НА БАЗЕ МЕТОДОЛОГИИ БЕРЕЖЛИВОГО ПРОИЗВОСТВА

Э.Т. Мухлисуллина
Научный руководитель Ю.В. Берчун
МГТУ им. Н.Э.Баумана, Москва, Российская Федерация

DEVELOPMENT OF THE METHOD OF PRODUCTION PROCESS SCHEDULING BASED ON LEAN PRODUCTION METHODOLOGY

E.T. Mukhlisullina
Scientific adviser Y.V. Berchun
Bauman Moscow State Technical University, Moscow, Russian Federation

Аннотация
В данной работе рассматривается один из вариантов организации календарного планирования производства с учетом нестандартных критериев оптимальности, таких как равномерность производства, суммарное время опережения или суммарное время рассогласования. Также в работе рассматривается механизм оценки указанных нестандартных критериев в рамках систем управления проектами. Разработано приложение, позволяющее автоматизировать этапы формирования исходных данных проекта и осуществлять расчёт критериев оптимальности в рамках предложенной методики.

Abstract
In this paper we consider a version of production scheduling, taking into account non-standard optimality criteria, such as uniformity of production, total lead time or total displacement time. In addition a technique for evaluating these non-standard criteria is considered. Proposed application can be used in small-scale production to improve the efficiency of production and process control to optimize production process by minimizing the non-standard optimality criteria.

1. Формализация постановки задачи

Рассмотрим формальную постановку задачи календарного планирования производства. Пусть O – множество заказов, J – множество работ, I – множество станков, K – множество стадий, объединяющих однотипные станки, так что $N = \sum_{j \in J} k_j$ [4].

Варьируемые параметры модели:

1) $X = \left\{ x_{i,j,l}, i \in I, l \in [1:k_j], j \in J \right\}$, где $x_{i,j,l}$ – момент начала выполнения операции l работы j на станке i;

2) $Y = \left\{ y_{i,j,l} \in \{0,1\}, i \in I, l \in [1:k_j], j \in J \right\}$, где $y_{i,j,l} = 1$ если l-ая операция j-ой работы выполняется на i-ом станке; $y_{i,j,l} = 0$, если l-ая операция j-ой работы не выполняется на i-ом станке;

3) $Z = \left\{ z_{i,j,l} \in [0:N], i \in I, l \in [1:k_j], j \in J \right\}$, где $z_{i,j,l}$ - номер по порядку выполнения l-ой операции j-ой работы на i-ом станке.

В качестве ограничений модели рассмотрим следующие условия.

1) Каждая операция любой работы выполняется на одном станке

$$\sum_{i \in I} y_{i,j,l} = 1, l \in [1:k_j], j \in J.$$

2) Начало любой операции может наступить лишь после завершения всех операций, ей предшествующих по технологии:

ЕСЛИ $y_{i,j,l} = 1$ И $y_{s,j,l-1} = 1$, ТО $x_{i,j,l} \geq x_{s,j,l-1} + t_{s,j,l-1}, l \in [2:k_j], i,s \in I, j \in J.$

3) Начало выполнения любой работы на станке может начаться лишь после завершения выполнения на этом станке предыдущей работы:

$$\text{ЕСЛИ } y_{i,j,l}=1 \text{ И } y_{i,v,s}=1 \text{ И } z_{i,j,l}=z_{i,v,s}+1,$$

$$\text{ТО } x_{i,j,l} \ge x_{i,v,s}+t_{i,j,l}+\tau_{i,v,j},\ s\in[1:k_v],\ l\in[1:k_j],\ i\in I,\ j,v\in J.$$

4) Момент начала выполнения самой первой для станка операции может наступить лишь после наладки станка на эту работу

$$x_{i,j,l} \ge r_{i,j,l}, \text{ если } z_{i,j,l}=1,\ i\in I,\ l\in[1:k_j],\ j\in J\ .$$

5) Естественные условия на введенные переменные

$$x_{i,j,l}\ge 0,\ y_{i,j,l}\in\{0,1\},\ z_{i,j,l}\in[0:N],\ i\in I,\ l\in[1:k_j],\ j\in J.$$

Очевидно, реализация указанных ограничений может быть осуществлена с использованием инструментальных средств систем управления проектами (СУП) [3].

2. Формулировка критериев оптимальности

В рамках методологии управления проектами в качестве основного критерия оптимальности рассматривается суммарная длительность проекта (в данном случае – время изготовления всей партии изделий) [3].

В то же время для производственных систем часто оказываются более важными другие показатели эффективности, что лежит в основе таких современных методологий управления и организации производства как «точно вовремя» (JIT, just-in-time) [2] и «бережливое производство» (lean production) [1]. К числу их важнейших концепций относится сокращение запасов и незавершённого производства, обеспечение непрерывности производственного процесса. Это требование можно формализовать как стремление привести сроки выполнения планируемых процессов (цепочек связанных задач) к их идеальной длительности с точки зрения технологии. Обозначим его как *суммарное время рассогласования (СВР)* $\sum \Delta t_i$ и потребуем его минимизации: $\sum \Delta t_i \to \min$. Время рассогласования для каждого процесса определяется как $\Delta t_i = t_{изгот.i} - t_{изгот.идеал.}$, где $t_{изгот.идеал.} = const$ – идеальное время выполнения технологического процесса, $t_{изгот.i}$ – планируемое время выполнения *i*-ого технологического процесса с учётом загрузки ресурсов (см. рис. 1) после их выравнивания с использованием стандартного инструментария СУП [3].

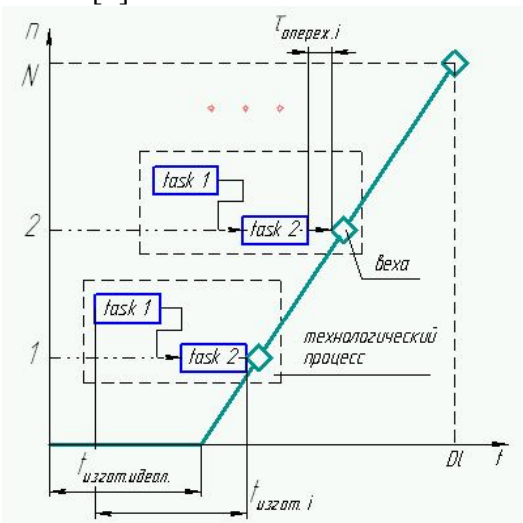


Рисунок 1 - График производства партии изделий

Другой важной концепцией при организации производства является его *равномерность*. Это означает, что программа выпуска должна выполняться с одинаковым темпом, график выпуска при этом представляет собой прямую линию (см. рис. 1). Это требование можно формализовать как *суммарное время опережения (СВО)* $\sum \tau_i = \sum \tau_{опереж.i}$, $\sum \tau_i \to \min$; для фиксации идеального времени завершения очередного процесса используется механизм *вех* [3], при этом все задачи планируются с опцией *«начать как можно позже»*.

3. Жизненный цикл календарного планирования

Предлагаемый в работе вариант календарного планирования производства с учетом нестандартных критериев оптимальности состоит из 3 этапов, представленных ниже.

1 этап. Подготовка исходных данных. Представим, что есть технологический процесс, который состоит из связанных подзадач. Необходимо автоматически продублировать данный технологический процесс *N* раз (в соответствии с программой выпуска, которая должна быть выполнена к определенному сроку завершения проекта *Dl* (*deadline)*). После дублирования универсальные ресурсы (*generic resource*) (в производственных процессах это, как правило, станки, рабочие места) распределяются по подзадачам всех процессов.

2 этап. Начальная расстановка вех для реализации равномерности производства. Каждой последняя подзадача каждого процесса связывается с вехой, которая изначально фиксируется для случая идеально равномерного производства (см. рис. 1). Тем самым, изначально достигнут оптимум по критерию СВО.

3 этап. Применение алгоритма многокритериальной оптимизации. Выполняется оценка критерия СВР. Если оно не является приемлемым, осуществляется сдвиг вех, что увеличивает значение критерия СВО, но может сократить СВР. Вопрос о выбор метода оптимизации на текущий момент находится на стадии исследования.

4. Программная реализация

На данном этапе реализованы 1 и 2 этапы, рассмотренные выше. Были расширены возможности существующего пакета *Microsoft Project 2010* с точки зрения описания процессов, для чего реализовано приложение (add-in) на языке C#. Данное приложение позволяет дублировать множества однотипных проектов с автоматическим отслеживанием изменений событий их атрибутов (например, название задачи, ее длительность, связи с предшествующими и последующими задачами).

Для разработки приложения под *MS Project 2010* была выбрана среда *Visual Studio 2010*, язык программирования C# [5]. В *Visual Studio 2010* представлены шаблоны проектов, которые можно использовать для создания надстроек уровня приложения для *Microsoft Office Project*. Надстройки позволяют автоматизировать приложение *Project*, а также расширить функциональные возможности и настроить пользовательский интерфейс этого приложения.

При расширении приложения *Microsoft Office* путем создания надстройки уровня приложения код создается непосредственно для класса *ThisAddIn* данного проекта. Этот класс можно использовать для получения доступа к объектной модели ведущего приложения *Microsoft Office*, настройки пользовательского интерфейса приложения, а также для предоставления объектов созданной надстройки другим решениям *Office*.

Написание кода в проектах надстройки в некоторых аспектах отличается от написания кода в других проектах *Visual Studio*. Многие из этих отличий связаны с тем, каким образом объектные модели *Microsoft Office* представлены управляемому коду.

Создание кода надстройки начинается в классе *ThisAddIn*. Visual Studio автоматически создает этот класс в файле кода *ThisAddIn.cs* (для C#) проекта надстройки. Среда выполнения *Visual Studio Tools for Office* автоматически создает экземпляр этого класса при загрузке надстройки приложением *Microsoft Office*.

В классе *ThisAddIn* существует два обработчика событий по умолчанию. Для выполнения кода при загрузке надстройки следует добавить код в обработчик событий *ThisAddIn_Startup*. Для выполнения кода непосредственно перед выгрузкой надстройки следует добавить код в обработчик событий *ThisAddIn_Shutdown*.

Каждое приложение, которое может быть автоматизировано при помощи Visual Studio имеет объектную модель. В объектной модели приложения *Project* предоставляется доступ к различным типам, которые можно использовать для автоматизации *Project*. Эти типы

позволяют создавать код для выполнения общих задач, таких как создание и изменение задач проекта программными средствами.

Для доступа к объектной модели Project из надстройки используется поле *Application* класса *ThisAddIn* проекта. В поле *Application* возвращается объект *Microsoft.Office.Interop.MsProject.Application* , который представляет текущий экземпляр *Project*.

5. Результат работы приложения

При запуске *MS Project 2010* всплывает дополнительное окно *Form1* (рисунок 2). С ячейкой для количества экземпляров и кнопкой *Duplicate*. В *Project* создаем свой проект, пусть он состоит, например, из трех задач, связанных между собой. Последняя задача – веха. Во вкладке «Лист ресурсов» назначаем задачам универсальные ресурсы *s1* и *s2*. Для лучшего понимания, пусть это будут станки 1 и 2, соответственно. Пусть *s1*=3, *s2*=2. Возвращаемся в Диаграмму Ганта и выделим при помощи мыши наши задачи, которые мы хотим объединить в одну группу и продублировать. Продублируем, например, наш проект 5 раз. Для этого вводим в ячейку число 5 и нажимаем на кнопку *Duplicate*. Получаем 5 идентичных проектов. Во время дублирования проекта происходит также дублирование ресурсов и вех.

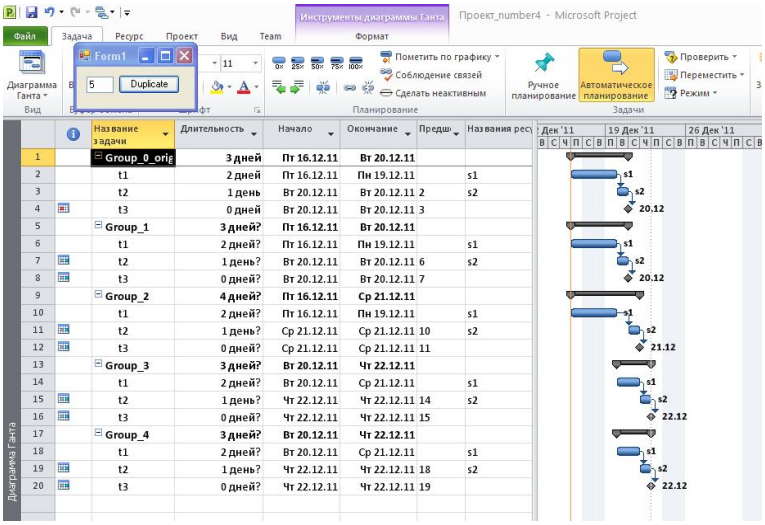


Рисунок 2 - Результат работы приложения

Следует заметить распределение производственного процесса по универсальным ресурсам, число станков 1го типа было 3, а 2го – 2. Если требуется изменить Название *t1*, например на *t0*, то это название изменится во всех группах. Аналогично и с длительностью, началом и окончанием задач и подзадач.

Литература

1. «Your Online Resource for Lean-Based Information and Tools», Электронный ресурс, http://www.leanproduction.com/, доступ свободный.
2. William J. Stevenson, Системы «точно-в-срок» (just-in-time): разработка и внедрение, Электронный ресурс, http://www.elitarium.ru, свободный доступ.
3. Куперштейн В. «Microsoft Project 2010», изд. «БХВ-Петербург», Санкт-Петербург, 2011.
4. Прилуцкий М.Х., Власов С.Е. Многостадийные задачи теории расписаний с альтернативными вариантами выполнения работ. //Системы управления и информационные технологии, ООО "Издательство "Научная книга", 2005.
5. «Начните разрабатывать на платформе Майкрософт», Электронный ресурс, http://msdn.microsoft.com/ru-ru/, доступ свободный.