

СЕКЦИЯ 1

ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ

Среда, 24 апреля 2013 г., читальный зал преподавателей (ауд.229), гл. корпус МГТУ им.Н.Э.Баумана.

Начало в 10.00.

Председатель: профессор, д.ф-м.н. Карпенко А.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., В.Б.МАНИЧЕВ, МАРТЫНЮК В. А., ТРУДОНОШИН В. А., С.В.ГРОШЕВ, КНЯЗЕВА С. Ю., РОДИОНОВ С. В., ФЕДОРУК Е. В., ШЕСТАКОВ С. А., БЕЛОУС В. В., ПИВОВАРОВА Н. В.

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.

НЕЧЕТКИЕ БАЗЫ ДАННЫХ И ЗАПРОСЫ

Валиуллин А.М.

Научный руководитель: к.т.н., доцент, Волосатова Т.М.

МГТУ им. Н.Э.Баумана, кафедра РК-6, Москва, Россия

FUZZY DATABASE SYSTEMS AND QUERIES

Valiullin A.M.

Supervisor: Dr., Prof., Volosatova T.M.

MSTU, Moscow, Russia

Аннотация

В статье рассматриваются вопросы обработки нечеткой информации, необходимость использования нечетких баз данных, способы хранения и извлечения такой информации в базах данных, описаны методы перевода нечетких запросов и модели построения "точных" запросов к классическим реляционным базам данных. Также нечеткие запросы могут быть описаны на расширенном языке FSQL для нечетких систем. Показана эффективность и результативность методов построения на приведенных в статье примерах.

Abstract

In this paper we examine problems of fuzzy data processing, necessity of using fuzzy databases, techniques used in the storage and retrieval of fuzzy data in a database, methods of translating fuzzy queries into precise queries for classical relational databases. Also fuzzy queries can be expressed in the Fuzzy SQL language. The efficiency and effectiveness of our approach is demonstrated by experimental results.

Введение

Все чаще в наши дни в запросах к реляционным базам данных, которые формулирует человек, часто присутствуют неточности и неопределенности. В целях увеличения применимости традиционных СУБД были введены новые методы для работы с неточной и неопределенной информацией. Использование теории нечетких множеств в базах данных привело к развитию нечетких баз данных.

Впервые проблема математического описания, методов и средств обработки данных в условиях нечеткости и неопределенности была описана в 1984 году. С тех пор было предложено много моделей описания и представления нечеткой информации в реляционных базах данных. Модель Баклиса-Петри [1] была первой моделью, которая отображала подобные связи в реляционной модели. Звиели и Чен первыми предложили подход [2] к объединению нечеткой логики с ER (Entity-Relationship) моделью. Эта модель разрешает нечеткие атрибуты в сущностях и связях. Чен и Керре [3] представили нечеткое расширение нескольких основных EER (Extended Entity-Relationship) концепций. Чодхри, Мойн и Рунденштейнер предложили метод [4] для построения нечетких реляционных баз данных согласно расширениям ER модели, предложенные Звиели и Ченом. Галиндо, Уррутиа и Пиаттини [5] описали использование нечеткой EER модели для моделирования базы данных и представления смоделированной нечеткой информации в реляционных базах.

1 Неопределенность, неточность, нечеткость и несоответствие данных

Различают 4 основных типа дефекта в описании информации: неопределенность, неточность, нечеткость и несоответствие. Все эти понятия имеют разный смысл. Различия между неопределенностью и неточностью детально описаны в трудах Дюбуа и Прада (1988).

Неопределенность означает отсутствие достоверной информации о состоянии объекта. Неопределенность возникает из-за неполноты знаний, относящихся к некоторому событию или наличию свойства какого-либо объекта. Например: «Завтра в 17:00 будет идти дождь.» Как мы знаем, прогноз погоды на завтра может сбыться, а может и не сбыться, то есть с какой-то долей вероятности дождя может завтра и не быть. Описания такого рода носят неопределенный характер.

Неточность относится к способу описания самого события, свойства. Например: «Возраст Михаила между 20 и 25 годами.» Это означает, что возраст Михаила с равной вероятностью может быть равен целым числам от 20 до 25. Такие описания несут дефект неточности.

Нечеткое высказывание содержит расплывчатое описание события или свойства. Например: «Михаил молодой исследователь.» Возраст Михаила описывается с помощью лингвистического термина «молодой». Нужно заметить, что значение этого термина зависит от контекста высказывания, так как большая бабочка меньше, чем небольшой слон.

Несоответствие информации означает, что мы имеем высказывания, которые не имеют пересечения. Например: «Михаилу 25 лет.» и «Возраст Михаила между 28 и 30 годами.» В этом случае мы не можем описать значение возраста Михаила, который будет удовлетворять этим двум высказываниям. Решение таких проблем было описано в работах Дюбуа, Ланга и Прадо (1992).

Итак, неопределенность означает, что хорошо описанное событие или явление может произойти, а может и не произойти, тогда, как неточность связана с неточным описанием и плохим определением события, явления или переменной. Нечеткость появляется с использованием лингвистического термина. Несоответствие возникает в ситуациях, когда нет возможности определения события, которое бы удовлетворяло различным источникам.

2 Нечеткие запросы

Нечеткие запросы - перспективное направление в современных системах обработки информации, они появились в связи с необходимостью смягчить булеву логику в реляционных базах данных. Этот инструмент дает возможность формулировать запросы, встречающиеся в повседневной жизни «Вывести список недорогих иномарок в Москве», что невозможно при использовании стандартного механизма запросов.

Механизм нечетких запросов к реляционным базам данных базируется на теории нечетких множеств, которая была впервые предложена Лотфи Заде.

Продемонстрируем ограниченность четких запросов на следующем примере.

Пример. Пусть требуется получить сведения о менеджерах по продажам не старше 25 лет, у которых сумма годовых сделок превысила 200 т.р. по такому-то региону. Данный запрос можно записать на языке SQL следующим образом:

```
SELECT * FROM Managers WHERE (Age <= 25 AND Sum > 200000);
```

Менеджер по продажам 26 лет с годовой суммой продаж в 400 т.р., или 19 лет с суммой в 198 т.р. не попадут в результат запроса, хотя их характеристики почти удовлетворяют требованиям запроса.

Нечеткие запросы помогают справиться с подобными проблемами «пропадания» информации.

3 Функции принадлежности

Существуют различные пути решения проблем нечетких запросов. В данной работе более подробно рассмотрим построение нечетких запросов с помощью введения функций принадлежности. Задача поиска информации в таких запросах решается с помощью формализованного описания множества значений термов логического выражения. Моделью таких структур является лингвистическая переменная и функция принадлежности [6].

Функцией принадлежности называется функция, которая позволяет вычислить степень принадлежности произвольного элемента x множества X к нечеткому множеству F .

Для примера формализуем нечеткое понятие «Возраст сотрудника компании». Это и будет название соответствующей лингвистической переменной. Зададим для нее область определения $X = [18; 70]$ и три лингвистических термина - "Молодой", "Средний", "Выше среднего". Последнее, что осталось сделать - построить функции принадлежности для каждого лингвистического термина.

Выберем трапециевидные функции принадлежности со следующими координатами:
 - "Молодой" = [18, 18, 28, 34],
 - "Средний" = [28, 35, 45, 50],
 - "Выше среднего" = [42, 53, 70, 70].

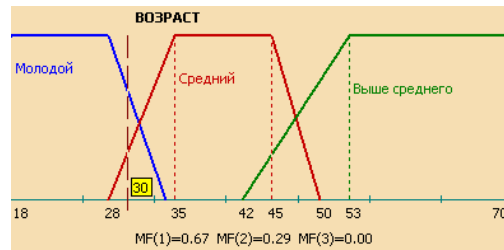


Рисунок 1 - Функции принадлежности термов переменной «Возраст сотрудника»

Теперь можно вычислить степень принадлежности сотрудника к каждому из нечетких множеств:

$$\mu[\text{“Молодой”}](30) = 0.67$$

$$\mu[\text{“Средний”}](30) = 0.29$$

$$\mu[\text{“Выше среднего”}](30) = 0$$

Основное требование при построении функций принадлежности – значение функций принадлежности должно быть больше нуля хотя бы для одного лингвистического термина.

Вернемся к примеру с менеджерами о продажах. Для простоты предположим, что вся необходимая информация находится в одной таблице со следующими полями: ID - номер сотрудника, AGE - возраст и SUM - годовая сумма сделок:

ID	AGE	SUM
1	23	120 500
2	25	164 000
3	28	398 000
4	31	489 700
5	33	251 900

Лингвистическая переменная «Возраст» была задана ранее. Определим еще одну лингвистическую переменную для поля SUM с областью определения $X = [0; 600000]$ и аналогично построим для них функции принадлежности:

- “Малая” = [0, 0, 0, 200000],

- “Средняя” = [90000, 180000, 265000, 330000],

- “Большая” = [300000, 420000, 600000, 600000].



Рисунок 2 - Функции принадлежности термов переменной «Годовая сумма сделок»

К такой таблице можно делать нечеткие запросы. Например, получить список всех молодых менеджеров по продажам с большой годовой суммой сделок:

*SELECT * FROM Managers WHERE (age = "Молодой" AND sum = "Большая");*

Рассчитав для каждой записи агрегированное значение функции принадлежности μ , получим результат нечеткого запроса:

ID	AGE	SUM	MF
3	28	398 000	0,82
4	31	489 700	0,50

Записи 1, 2, 5 не попали в результат запроса, т.к. для них значение функции принадлежности равно нулю. Записей, точно удовлетворяющих поставленному запросу ($\mu = 1$), в таблице не нашлось. Менеджер по продажам 28 лет и годовой суммой 398000 соответствует запросу с функцией принадлежности 0,82. Аналогичный четкий запрос мог бы быть сформулирован, например, так:

*SELECT * FROM Managers WHERE (age <= 28 AND sum >= 420000);*

Его результат является пустым. Однако если мы немного расширим рамки возраста в запросе, то рискуем упустить других сотрудников с чуть более большим или меньшим возрастом. Поэтому можно сказать, что нечеткие запросы позволяют расширить область поиска в соответствии с изначально заданными человеком ограничениями.

Заключение

В данной работе показан эффективный метод обработки нечетких запросов с помощью введения функций принадлежности. Существует много различных методов задания функций принадлежности. Выбор того или иного метода зависит от задачи, существующей ситуации (например, наличия экспертов) и других параметров.

Реляционная модель представления и реализации нечетких моделей может стать связующим звеном для интеграции с существующими реляционными базами данных. Также в настоящее время активно развивается расширение языка SQL, так называемый стандарт FSQL. Расширенный язык FSQL позволит получить новый инструмент для извлечения новых ранее недоступных данных и знаний.

Литература

1. В.Р. Buckles, F.E. Petry (1982). A fuzzy representation of data for relational databases. Fuzzy Sets and Systems no. 7, pages 213-226.
2. A. Zvieli, P. Chen (1986). ER modelling and fuzzy databases. In Proceedings of the Second International Conference on Data Engineering. pages 320-327, LA, USA.]
3. G.Q. Chen, E.E. Kerre (1998). Extending ER/EER concepts towards fuzzy conceptual data modelling. Proceedings of the IEEE International Conference on Fuzzy Systems. Pages 1320-1325, Anchorage, AK, USA.]
4. N. Chaudhry, J. Moyne, E. Rundensteiner (1994). A design methodology for databases with uncertain data. Proceedings of the Seventh International Working Conference on Scientific and Statistical Database Management. Pages 32-41, Charlottesville, VA, USA.
5. J. Galindo, A. Urrutia, M. Piattini (2006). Fuzzy Databases: Modelling Design and Implementation. Hershey, USA: IDEA Group.]Введение в математическое моделирование. Учебное пособие. Под ред. П. В. Трусова. — М.: Логос, 2004. 200 с.
6. Технологии анализа данных [Электронный ресурс]. М., 1995-2013. URL: <http://www.basegroup.ru>. (Дата обращения: 10.02.2013).

АВТОМАТИЧЕСКОЕ ПОСТРОЕНИЕ ПАНОРАМНОГО ИЗОБРАЖЕНИЯ ПО НАБОРУ АЭРОФОТОСНИМКОВ ДЛЯ ИСПОЛЬЗОВАНИЯ В ГЕОИНФОРМАЦИОННОЙ СИСТЕМЕ

Афиногенов Е.И.

Научный руководитель: к.т.н., доцент, Волосатова Т.М..

МГТУ им. Н.Э.Баумана, кафедра РК6, Москва, Россия

AUTOMATIC PANORAM CREATION FROM LARGE COLLECTION OF IMAGES

Afinogenov E.I.

Supervisor: associate professor Volosatova T.M.

MSTU, Moscow, Russia

Аннотация

В статье рассматривается задача автоматического построения панорамного изображения по множеству аэрофотоснимков. Решение задачи осуществляется в 3 этапа: поиск ключевых точек на изображениях, поиск соответствий с использованием эффективной схемы индексирования на основе словарного дерева и непосредственной склейки изображений с выполнением необходимых преобразований (поворот, масштабирование).

Abstract

The article deals with automatic creation of panoram from large collection of images. It is proposed 3 stages for solve this task: keypoints detection, search of correspondences using effective indexing scheme based on vocabulary tree, image transformation (rotation and scaling) and stitching.

Введение

Несмотря на наличие множества автоматизированных геоинформационных систем (ГИС) широкого назначения, остается проблема оперативной обработки данных, полученных средствами воздушной разведки. Сбор первичных данных об объектах и их последующая обработка для решения военных задач и хозяйственных вопросов становится все более актуальным.

Одна из проблем, возникающих при создании законченных геоинформационных систем – проблема преобразования географических данных из аэрофотоснимков в электронную форму для последующей работы с ними. На этапе предварительной подготовки данных часто осуществляется склейка имеющихся фотоснимков в единое панорамное изображение. При этом процесс склейки изображений зачастую сопровождается выставлением опорных точек (маркеров) и последовательным ручным добавлением снимка в панораму. Многие вопросы сбора и автоматизированной обработки данных в геоинформационных системах (ГИС) успешно решены, однако это сложные и дорогостоящие АРМ. В данной работе рассматривается задача автоматического объединения множества аэрофотоснимков с частичным перекрытием в единое панорамное изображение.

Постановка задачи

Пусть имеется база данных аэрофотоснимков некоторых частично перекрывающихся участков местности. Необходимо построить панорамное изображение по данным фотоснимкам с целью облегчения последующей обработки при использовании в специализированной ГИС. При этом изображения могут быть получены при различном масштабе и ориентации в пространстве, а также содержать шум, дефокусировку и смаз.

Упрощенная схема предлагаемой системы панорамирования представлена на рис. 1.

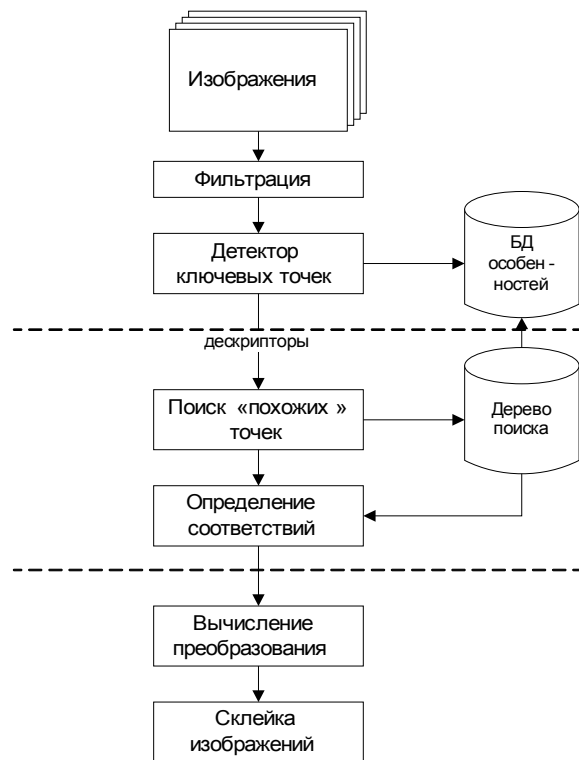


Рисунок 1 - Схема предлагаемой системы панорамирования

Весь процесс создания панорамы можно разделить на два больших этапа:

- 1) Извлечение особых (ключевых) точек на аэрофотоснимках с предварительной фильтрацией изображений.
- 2) Выполнение склейки путем нахождения соответствующих точек и выполнения преобразования изображений для приведения к единому виду (ориентация в пространстве, масштабирование).

1 Поиск особых (ключевых) точек на изображении

Обнаружение и сопоставление точечных особенностей на изображении является важной задачей в области компьютерного зрения. Рассмотрим задачу более детально.

Пусть нам даны два изображения некоторой сцены. Требуется найти набор пар точек $(x_{1,i}, y_{1,i})(x_{2,i}, y_{2,i}) \quad i = 1 \dots N$ таких, что $(x_{1,i}, y_{1,i})(x_{2,i}, y_{2,i})$ являются изображениями одной и той же точки.

Заметим, что для сопоставления подходят далеко не все точки изображения. Например, очень сложно найти соответствующую точку для некоторой точки изображения однородной поверхности. Поэтому для сопоставления используются так называемые особые точки, то есть такие точки изображения, окрестности которых можно отличить от окрестности любой другой точки изображения.

Для сопоставления обнаруженных особенностей используются дескрипторы, представляющие собой вектора числовых характеристик окрестностей ключевых точек $D(x) = [f_1(w(x)), \dots, f_n(w(x))]$.

В настоящее время разработано огромное количество методов поиска особенностей. Наиболее известными являются уголкового детектор Харриса [3], SIFT (Scale-Invariant Feature Transform) [4], SURF (Speed-Up Robust Features) [5], Ferns [6], GLOH (Gradient Localization-Orientation Histograms) [1].

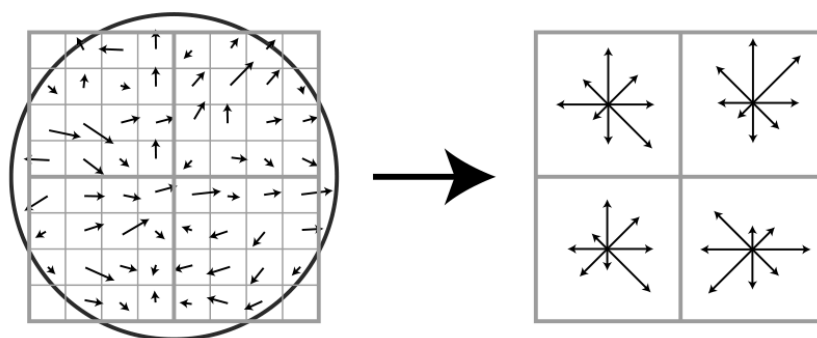


Рисунок 2 - SIFT-дескриптор

В условиях поставленной задачи дескриптор должен обладать инвариантностью к повороту, масштабированию и изменениям в освещенности, а также быть устойчивым к некоторому наличию шума в изображениях. Наиболее подходящим в данном случае является SIFT-дескриптор, который представляет собой локальную нормализованную гистограмму градиентов интенсивностей пикселей в окрестности некоторой точки (рис. 2).

2 Поиск соответствий

После того, как найдены ключевые точки для всех изображений, осуществляется поиск соответствий для последующей склейки. С учетом того, что изображений в базе данных может быть от нескольких сотен до нескольких тысяч, а количество ключевых точек на одном изображении в среднем составляет несколько сотен, то прямое сравнение каждой точки со всеми найденными является крайне медленным (особенно при использовании высокоразмерных дескрипторов).

Существуют различные способы ускорения данной процедуры за счет использования многомерных деревьев поиска, хэширования и т.д. В данной работе для обеспечения эффективного поиска изображений, содержащих определенные вектора признаков, предлагается использовать схему на основе обратного индекса, получившую распространение при поиске текстов по набору ключевых слов. Основная идея заключается в том, чтобы построить из признаков (дескрипторов) словарь «визуальных слов» [7].

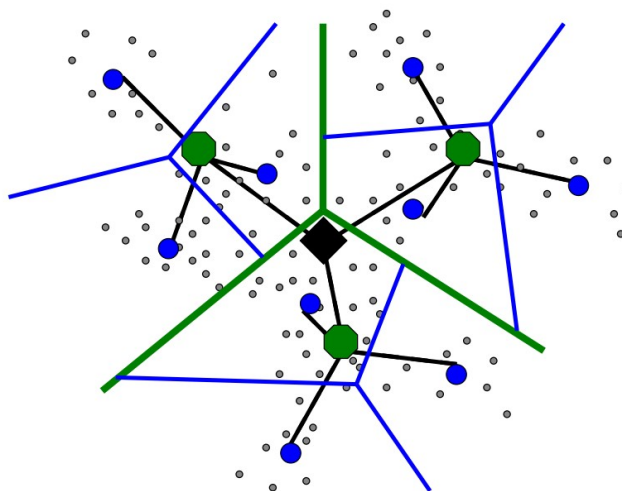


Рисунок 3 - Пример построения словарного дерева (для $k = 3$)

Отобразим дескрипторы особых точек изображений в слова путем квантования пространства признаков (рис. 3). Квантование осуществляется с помощью иерархической кластеризации пространства методом k -средних. Таким образом, словарь организуется в виде дерева. Каждый узел дерева (обратный индекс) хранит идентификаторы изображений и частоту соответствующего слова в изображении.

После того, как словарное дерево построено, поиск соответствий на изображениях начинается с определения кластера по входному дескриптору ключевой точки. Извлекаемый индекс определяет изображения, где содержатся «похожие» точки. Изображения, содержащие наибольшее количество близких особенностей, выбираются в качестве кандидатов на склейку.

Использование данной схемы индексирования позволяет осуществлять эффективный поиск близких фрагментов среди большого числа изображений (нахождение заданного фрагмента в базе данных из 50000 изображений менее, чем за 0.5 с [7]).

3 Вычисление преобразования

После определения достаточного количества общих точек на изображениях, осуществляется попарная склейка путем приведения к единой системе координат и масштабу. Для решения данной задачи используется устойчивая схема факторизации на основе разложения по сингулярным значениям (SVD – Singular Value Decomposition) [2].

Заключение

В данной работе рассмотрена задача автоматического построения панорамного изображения по множеству аэрофотоснимков для использования в специализированной геоинформационной системе. Склейка изображений осуществляется на основе выделенных особых точек с помощью алгоритма SIFT. Благодаря использованию специальной схемы индексирования на основе словаря «визуальных слов» поиск одинаковых близких фрагментов осуществляется за доли секунды даже при огромной базе изображений (десятки тысяч). Приведение аэрофотоснимков к единой системе координат осуществляется с помощью устойчивой схемы факторизации на основе SVD-разложения.

Литература

1. Szeliski, R. Computer Vision: Algorithms and Applications. Springer, Sept. 3, 2010 draft.
2. Tomasi C., Kanade T. Shape and Motion from Image Streams: a Factorization Method Full Report on the Orthographic Case, 1992
3. C. Harris and M.J. Stephens. A combined corner and edge detector. In Alvey Vision Conference, pages 147-152, 1988.
4. David G. Lowe. Object recognition from local scale-invariant features. International Conference on Computer Vision, pages 1150-1157, 1999.
5. 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.
6. V. Lepetit and P. Fua. Keypoint recognition using randomized trees. IEEE Transactions on Pattern Analysis and Machine Intelligence, 28(9): 1465-1479, Sept. 2006.
7. D. Nistér and H. Stewénius, “Scalable recognition with a vocabulary tree,” IEEE Conference on Computer Vision and Pattern Recognition, 2006

АНАЛИЗ И СРАВНЕНИЕ ПРОСТЕЙШИХ МЕТОДОВ ФИЛЬТРАЦИИ ДАННЫХ

Алексеев М.А.

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

ANALYSIS AND COMPARE SIMPLE METHODS OF DATA FILTRATION

Alekseev M.A.

Supervisor: Candidate of technical science, reader Vlasov A.I.
MSTU, Moscow, Russia

Аннотация

В статье рассмотрены два простейших методов фильтрации данных, на примере данных, полученных с акселерометра. Проведён анализ полученных теоретических и практических результатов работы фильтров. Даны рекомендации по использованию рассмотренных методов.

Abstract

The article discussed in detail two simple methods of data filtration, in terms of data taken from accelerometer. Carried out analysis of received theoretical and practical results of filters working. The recommendation about usage of filtration methods discussed was given.

Введение

В настоящее время в технике применяется большое количество разнообразных цифровых датчиков. По мере развития технологий точность цифровых датчиков постоянно растёт, но также появляются проблемы, связанные с зашумлением показаний этих датчиков. Таким образом, тема фильтрации показаний приобретает всё большую актуальность. В данной работе рассмотрены два простейших метода фильтрации. В первом методе используется среднее арифметическое значение предыдущих n измерений:

$$x = \frac{\sum_{j=1}^n m_j}{n}, \text{ где}$$

x – среднее значение n измерений,

m_j – показание датчика в j -ый момент времени,

n – число измерений.

Особенностью данного метода, что при увеличении количества измерений, при плавных изменениях показаний, разность между средним и действительным значением уменьшается, но это делает данный метод плохо применимым при быстро изменяющихся показаниях.

Второй метод, «Альфа-фильтр», является простейшим рекурсивным фильтром:

$$X_i = X_{i-1} + \alpha \cdot (m_i - X_{i-1}), \text{ где}$$

X_i - отфильтрованное значение в i -ый момент времени,

α - весовой коэффициент,

m_i - измерение в i -ый момент времени.

Моделирование фильтров было осуществлено в программе Mathcad 14.0. Сначала заданный чистый сигнал зашумлялся, а затем фильтровался. Таким образом были получены графики, показывающие результат моделирования работы фильтров (рисунки 2 и 3).

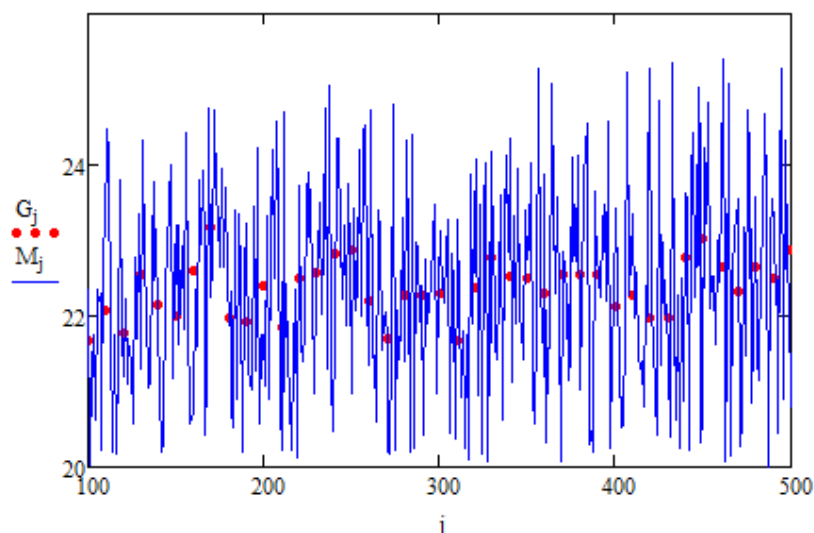


Рисунок 2 – Результаты моделирования работы метода «среднего арифметического»

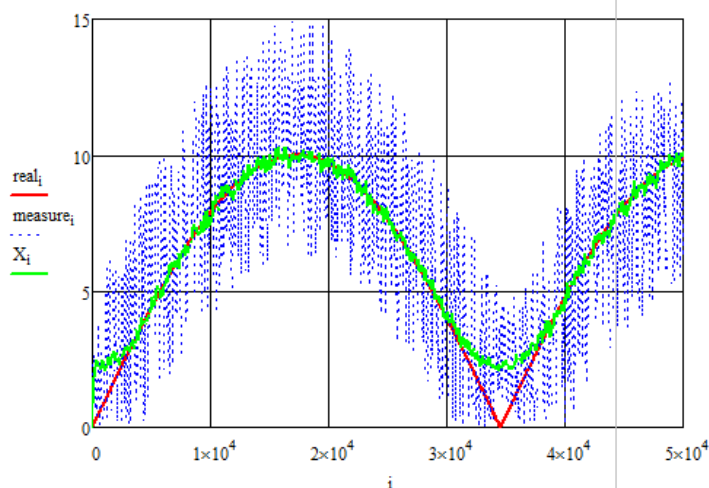


Рисунок 3 – Результаты моделирования работы «Альфа-фильтра»

Получение экспериментальных данных

Для получения экспериментальных данных был использован цифровой акселерометр. Измерения показаний датчика осуществлялось при периодических отклонение его разные стороны вдоль оси Y, в результате этих отклонений показания оси X изменялись. Считываемые данные поступали на персональный компьютер для дальнейшей обработки.

Постобработка данных

Проведение постобработки полученных экспериментальных данных была осуществлена в программе OriginPro 8.5, в соответствии с описанными ранее методами фильтрации. В OriginPro 8.5 были получены графики значений до фильтрации и после. На рисунке 7 показан результат применения метода «среднего арифметического».

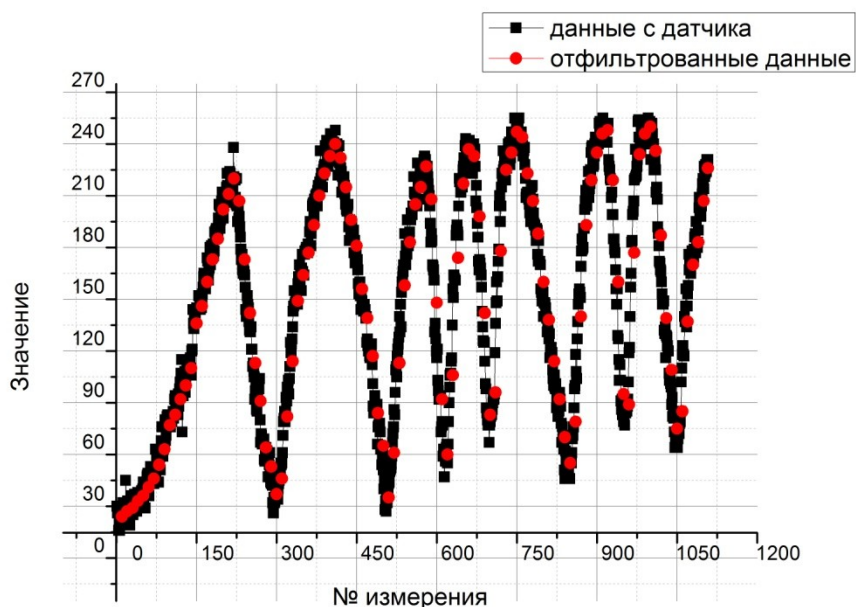


Рисунок 7 – Результат работы метода «среднего арифметического» с частотой одна коррекция на десять измерений

Также, для сравнения, была проведена фильтрация с частотой одна коррекция на три измерения (рисунок 8).

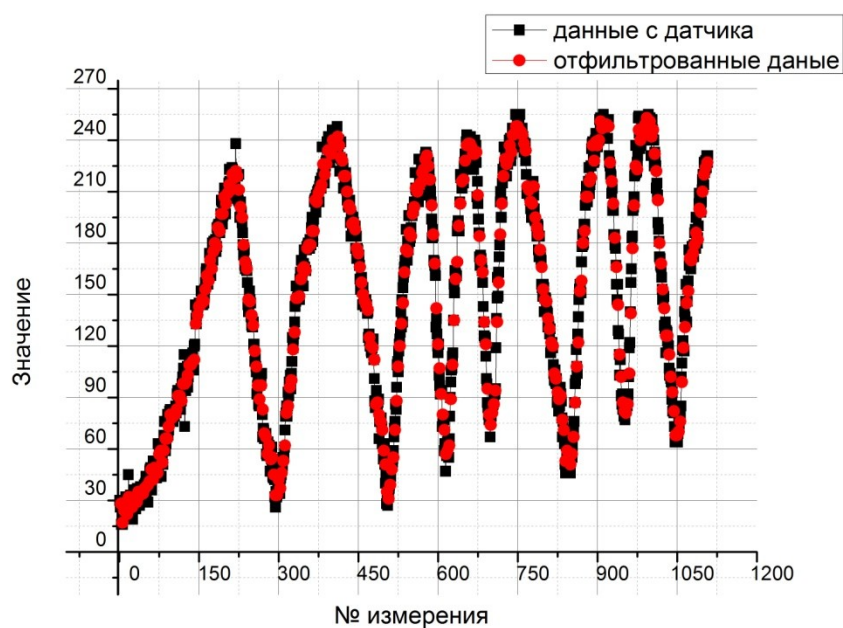


Рисунок 8 – Результат работы метода «среднего арифметического» с частотой одна коррекция на три измерения

Далее представлены результаты работы «Альфа-фильтра», основной особенностью которого является значение весового коэффициента $0 < \alpha < 1$, поэтому ниже представлены графики при различных весовых коэффициентах α (рисунки 9, 10, 11).

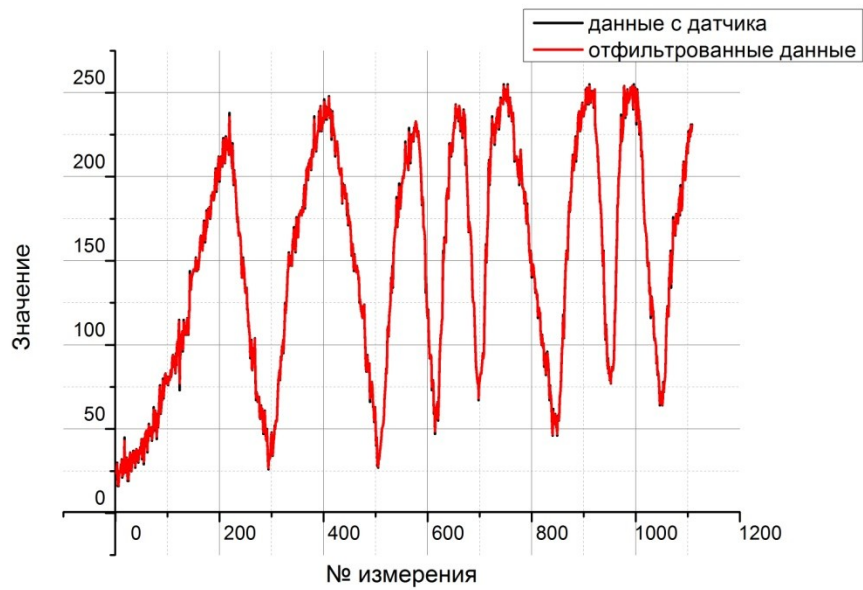


Рисунок 9 – Результат работы «Альфа-фильтра» с весовым коэффициентом $\alpha=0.9$

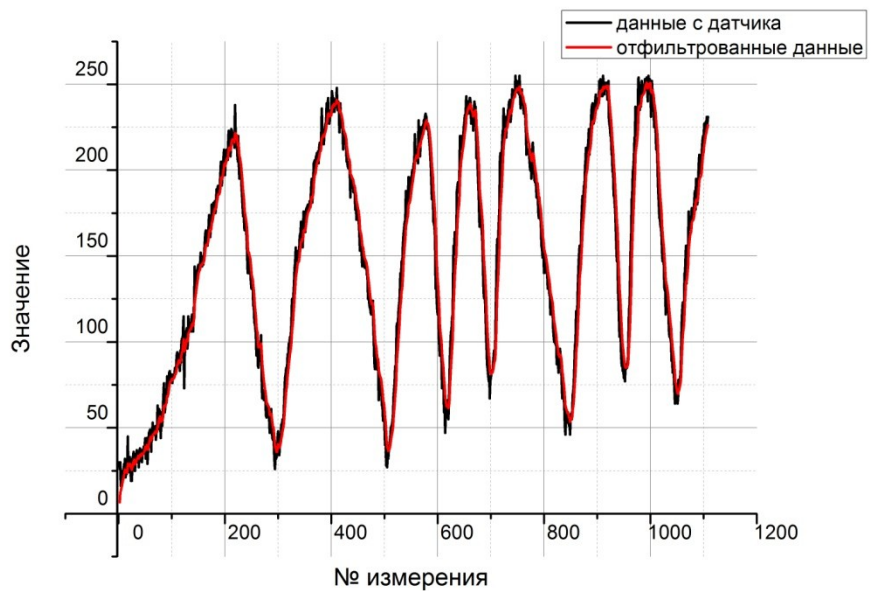


Рисунок 10 – Результат работы «Альфа-фильтр» с весовым коэффициентом $\alpha=0.25$

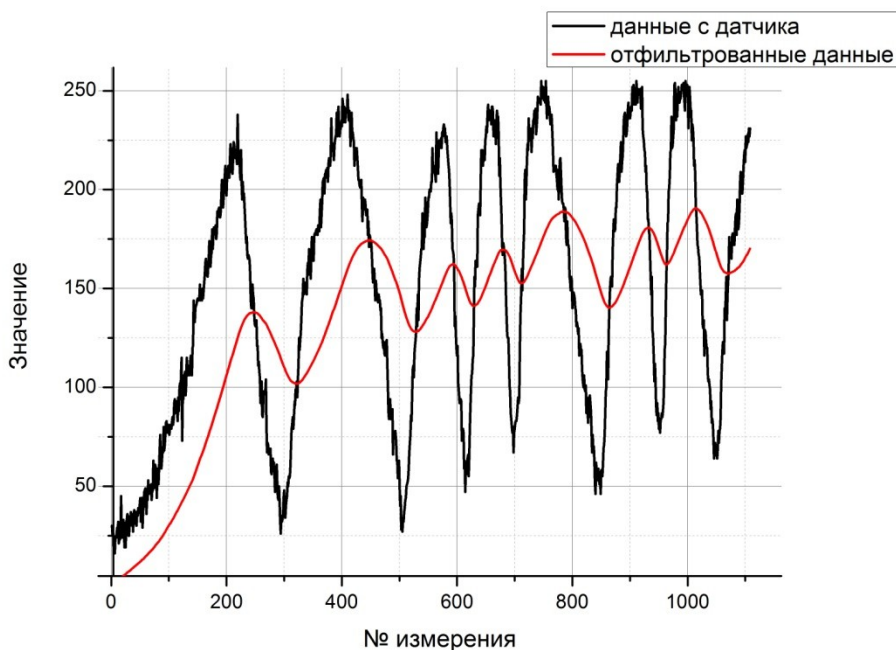


Рисунок 11 – Результат работы «Альфа-фильтра» с весовым коэффициентом $\alpha=0.01$

Исходя из приведённых выше графиков, можно сделать заключение, что результат работы фильтра улучшается с уменьшением весового коэффициента α , но, в то же время, ухудшается быстродействие. Таким образом, значения весового коэффициента α необходимо подбирать опытным путём, исходя из условий поставленной задачи.

Анализируя полученные результаты можно сделать следующие выводы:

Метод «среднего арифметического» хорошо подходит для задач, где не нужно полное избавление от шумов. Так же метод «среднего арифметического» целесообразно использовать там, где скорость изменения показаний не велика. Плюсами первого метода является простота, минусами – невозможность вносить корректировку в каждое измеренное значение.

«Альфа-фильтр» стоит использовать в тех задачах, где необходимо вносить корректировки в каждое измеренное значение. Плюсы – возможность делать корректировки для каждого полученного измерения, минусы – сложность правильного подбора весового коэффициента α .

Заключение

В данной работе было произведено математическое моделирование двух простейших фильтров данных с помощью программы Mathcad 14.0. Были получены экспериментальные данные, на основе которых была проведена постобработка и проведён анализ работы фильтров, даны рекомендации по их применению.

Отдельные результаты работы получены в рамках Гранта № 14.В37.21.0453 по ФЦП «Научные и научно-педагогические кадры инновационной России» на 2009 – 2013 годы.

Литература

1. Микушин А. В. Цифровые устройства и микропроцессоры. — БХВ-Петербург, 2010. – 832 с.
2. Антонью А. Цифровые фильтры. Анализ и проектирование. — Радио и связь, 1983. – 546 с.
3. <http://easyelectronics.ru/>

МОДЕЛИРОВАНИЕ ШАГАЮЩЕГО ШЕСТИНОГО РОБОТА

Андриановский П.А., Седов А.И.
научный руководитель: д.ф.м.н, профессор Карпенко А.П.
МГТУ имени Баумана, кафедра РК6, Москва, Россия

MODELLING OF WALKING 6-LEGS ROBOT

Andrianovskiy P.A., Sedov A.I.
scientific chief: professor Karpenko A.P.
BMSTU, RK6 Dep., Moscow, Russia
pavel.andrianovskiy@gmail.com, a.sedov14@gmail.com

Аннотация

В докладе рассмотрен процесс моделирования и проектирования шагающего шестиногого робота – *hexapod*. Приведено описание этапов проектирования, включая разработку программного обеспечения для плат системы управления.

Abstract

This paper introduces the modeling and design process of a walking 6-legs robot – hexapod. Also it contains description of design stages, including development of software for control circuits.

Введение

Шагающий шестиногий робот – устойчивое средство передвижения, обладающее высоким уровнем проходимости. В основе принципов движения шестиногого робота лежит анализ передвижения пауков. Обладая высокими свойствами проходимости по пересеченной местности и малыми размерами, они могут оставаться в устойчивом положении даже при упоре на три ноги.

Способ передвижения с помощью конечностей: шаг, бег, прыжок, как известно, является наиболее распространенным в живой природе. Однако в технике он еще не получил заметного применения, прежде всего из-за сложности управления. Развитие робототехники создало необходимую научно-техническую основу для реализации этого принципиально нового для техники способа передвижения и для создания нового типа транспортных машин – шагающих.

В задачу системы управления шагающей машины входят:

- Стабилизация в процессе движения, положения корпуса машины в пространстве на определенной высоте от грунта, независимо от рельефа местности;
- Обеспечение движения по определенному маршруту с обходом препятствий;
- Связанное управление ногами, реализующее определенную походку с адаптацией к рельефу местности.

Поскольку основное назначение шагающих машин – передвижение по пересеченной местности, управление ими обязательно должно быть адаптивным, при передвижении должны учитываться факторы внешней среды: рельеф и наклон местности, препятствия на пути. В системе управления при этом обычно выделяют следующие три уровня управления:

- Первый уровень – управление приводами степеней подвижности ног;
- Второй уровень – построение походки, т.е. координация движений ног со стабилизацией при этом положения корпуса в пространстве;
- Третий уровень – формирование типа походки, направления и скорости движения, исходя из заданного маршрута в целом.

Для реализации поставленной задачи был спроектирован шестиногий робот, конечности которого имеют по три степени свободы, что даст в будущем возможность реализовать все виды передвижений по плоскости: вперед, назад, влево, вправо и разворот, включая движения каждой конечностью отдельно.

1 Подготовка к проектированию

В ходе первого этапа подготовки к проектированию был произведен анализ уже существующих моделей роботов. Кроме того, был проведен анализ доступных материалов несущей конструкции робота и двигательных элементов – сервоприводов. По результатам этого этапа были выбраны оптимальная форма и конструкционный материал.

Конструкция модели была спроектирована с учетом требований по весу, жесткости и устойчивости робота. Особенностью данной конструкции является строение крепления сервоприводов, образующее плечевое сочленение. В действительности, оно похоже на подобное сочленение у живых организмов: есть плечо, предплечье и конечность. К этим деталям прикреплены три сервопривода таким образом, чтобы обеспечивать три степени свободы каждой конечности, что приводит к довольно большому размеру рабочей области конечности. Это приводит к большой устойчивости конструкции. В качестве среды проектирования всей конструкции робота, включая также полную сборку модели с крепежными элементами и сервоприводами, использовалась учебная версия продукта Autodesk Inventor 2013. Далее 3D-модели деталей были переданы специалисту, который из выбранного нами материала вырезал на фрезерном станке с ЧПУ все составные части конструкции робота. В качестве конструкционного материала был выбран композитный алюминий. Данный материал обладает высокой прочностью, очень низким весом (даже относительно цельного алюминия), и при этом небольшой ценой. Для реализации трех степеней свободы были выбраны сервоприводы MG995 Tower Pro, обладающие следующими характеристиками: рабочее напряжение +5 В, максимальный момент 10 кг/см. Данные сервоприводы выгодно отличаются от многих других, представленных на рынке, наличием в своем составе металлического редуктора, в то время как многие другие оборудуются пластиковым.

Аппаратное обеспечение, кроме двигателей-сервоприводов, также включало в себя электронную систему управления. Проанализировав поставленную задачу, было принято использовать две платы MSP430 LaunchPad с микроконтроллером MSP430G2553. Данный микроконтроллер – 16-битная система, работающая на частоте 16 МГц. Подробные характеристики приведены в таблице 1. Данные платы были выбраны по причинам доступности, низкой цены, сравнительно неплохих характеристик, а также по причине простоты написания кода – исходный код пишется на языке С, при этом доступно большое число примеров и уроков для работы с данным микроконтроллером. Однако, в качестве минуса можно указать, что пришлось использовать две платы, по причине того, что число информационных выходов на этой плате оказалось меньше, чем число сервоприводов.

Таблица 1 – Характеристики микроконтроллера MSP430G2553

Тактовая частота	16 МГц
Flash	16 Кб
SRAM	512 байт
Цифровые входы/выходы	24
Таймеры	2
Последовательный порт	USCI, SPI, HW UART
АЦП	8x10 бит

Помимо указанных в таблице характеристик, микроконтроллер имеет в своем составе датчик температуры, сторожевой таймер и компаратор. Ключевой фактор, позволивший использовать данные платы для работы с сервоприводами – встроенные 16-битные таймеры имеют поддержку ШИМ.

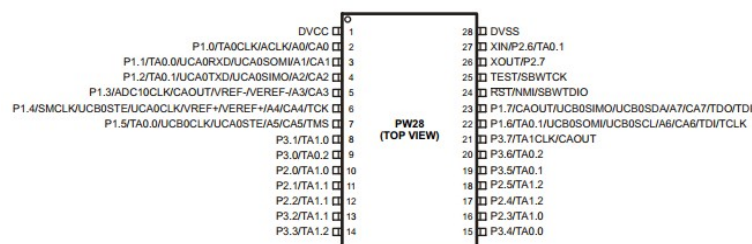


Рисунок 1 – Обозначение выходов микроконтроллера MSP430G2553

Сервоприводы управляются ШИМ-сигналом, а именно отношением «нуля» и «единицы» за единицу времени. Получая на вход скважность, сервопривод поворачивает вал на угол, пропорциональный скважности. Широтно-импульсная модуляция – это способ кодирования аналогового сигнала путем изменения ширины (длительности) прямоугольных импульсов несущей частоты. Таким образом, имея в распоряжении лишь два логических уровня, «единицу» и «ноль», можно получить любое промежуточное значение аналогового сигнала.

2 Разработка ПО

Одним из основных вопросов стала разработка программного обеспечения для плат системы управления. Первоначальная версия ПО заключалась в том, что на каждый сервопривод будет приходиться по таймеру с минимум двумя регистрами счета. Недостатки данного способа – потребность большого числа таймеров. Достоинства – отсутствие нагрузки на основной процессор.

Микроконтроллер MSP430G2553 имеет в наличии два таймера, с возможностью организации на них ШИМ-сигнала, но, к сожалению, на каждую плату приходится по девять сервоприводов, соответственно, первый вариант не подходит. Был разработан второй вариант, в котором ШИМ-сигнал образовывался двумя таймерами и прерыванием. Принцип работы состоял в следующем. Сначала запускается нулевой таймер, который отсчитывает 20 мс. По прерыванию данного таймера на все сервоприводы подается высокий уровень. Количество Ticks устанавливается в ноль, запускается таймер 1. Последовательно для каждого подключенного сервопривода значение Ticks сверяется с некоторым заданным числом. Если значения совпадают – то на выход подается низкий уровень. Достоинства способа – для обеспечения работы достаточно 2 таймеров. Недостаток – высокая нагрузка на основной процессор, так как каждые 20 мс запускается прерывание, после чего каждые 32 такта запускается прерывание с последующим блоком из нескольких условных переходов. К сожалению, микроконтроллер MSP430G2553 не смог управлять тремя ногами (девятью сервоприводами), из-за высокой вычислительной нагрузки на основной процессор, в ходе эксперимента было установлено, что максимальное число сервоприводов – три.

Заключение

Таким образом, для решения поставленной задачи был принят третий вариант. Данный способ не рекомендуется использовать, но на двух платах реализовать другие два способа не представляется возможным. Изначально инициализируются все выходы, устанавливаются на низкий уровень. Далее запускается бесконечный цикл. Последовательно каждый выход устанавливается в высокий уровень, после чего происходит задержка (delay) и устанавливается низкий уровень. Достоинства метода – простота реализации. Недостатки – невозможность проводить вычислительные процессы, так как ресурс процессора занят операцией задержки, невозможность изменить положение ног.

Литература

1. Интернет-портал MSP430 // <http://processors.wiki.ti.com/index.php/LaunchPads>
2. MSP430x1xx Family Users's Guide. // www.ti.com/lit/ug/slau049f/slau049f.pdf
3. ШИМ и PWM – что это такое? // http://myrobot.ru/stepbystep/rce_pwm.php

ВЫБОР МОДЕЛИ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ ИСПОЛЬЗУЕМОЙ ПРИ СОЗДАНИИ БАЗЫ ЗНАНИЙ ДЛЯ СИСТЕМ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ ДЛЯ ЭКСПЛУАТАЦИИ РОБОТИЗИРОВАННЫХ ПАРКИНГОВ

Беньков А. А.

Научный руководитель: к. т. н., доцент, Жук Д. М.
МГТУ им. Н. Э. Баумана, кафедра РК6, Москва, Россия

THE MODEL OF REPRESENTATION OF KNOWLEDGE SELECTION USED FOR DEVELOPING THE KNOWLEDGE BASE FOR DECISION SUPPORT SYSTEM FOR THE OPERATION OF THE ROBOTIC PARKING

Benkov A. A.

Supervisor: Ph. D., associate professor, Zhuk D. M.
MSTU, Moscow, Russia

Аннотация

Рассмотрены вопросы разработки методики построения системы управления эксплуатацией роботизированных паркингов. Для выполнения поставленной задачи он должен разработать методику построения всех компонентов, входящих в систему управления эксплуатацией роботизированных паркингов. Одним из компонентов системы управления эксплуатацией является систем поддержки принятия решений. Базы знаний являются неотъемлемой частью систем поддержки принятия решений. По этой причине разработка методики построения базы знаний для систем поддержки принятия решений для эксплуатации роботизированных паркингов является первым шагом разработки методики построения таких систем поддержки принятия решений. Для представления знания в базах знаний разрабатываются различные модели представления знания. По этой причине выбор подходящей модели представления знания является первичной задачей разработки методики построения базы знаний. В статье кратко освещены логические, продукционные, фреймовые, сетевые, объектно-ориентированные, специальные и комплексные модели, проанализированы преимущества и недостатки каждой из моделей, и произведен выбор необходимой модели для решения поставленной задачи.

Annotation

The author is working on the development of the method for creating maintenance management system for robotic parking. To accomplish his work he must develop a method for creating all of the components of the maintenance management system for robotic parking. One of the components of the maintenance management system is the decision support system. Knowledge base is an integral part of the decision support system. For this reason the development of the method for creating a knowledge base for decision support system for the operation of the robotic parking is the first step of the development of the method for creating such decision support systems. Various models of knowledge representation were developed for representing the knowledge in knowledge bases. For this reason, the choice of a suitable model of representation of knowledge is the primary objective of developing the method for creating a knowledge base. The paper briefly reviews the logic, productional, frame, network, object-oriented, special and complex models, the advantages and disadvantages of each model, and the choice of the required model for the main task was made.

Введение

Эксплуатация сложных технических комплексов сопряжена с высокими экономическими и эксплуатационными рисками. В общем случае риски эксплуатации связаны с отказами технических средств, возникновением нештатных и аварийных ситуации по вине обслуживающего персонала, а так же с экономическими потерями при несвоевременном проведении технического обслуживания компонентов комплексов [1, 2].

Применение автоматизированных систем поддержки принятия решения (СППР) представляется целесообразным на этапе использования технического комплекса по назначению [2].

Роботизированный паркинг является сложным техническим комплексом по своему составу. По этой причине в рамках системы управления эксплуатацией роботизированных паркингов целесообразно использовать СППР.

Использование системы хранения информации и знаний, собираемых в процессе разработки, строительства, эксплуатации и ремонта роботизированных паркингов является неотъемлемой частью СППР в системах управления эксплуатации роботизированных паркингов. Для этих целей целесообразно использовать комбинацию баз знаний (БЗ) для хранения связей и закономерностей предметной области, и баз данных (БД) для хранения объектов предметной области [2, 3].

Постановка задачи

Выбор способа представления знаний в интеллектуальной системе является ключевым в разработке систем хранения знаний. С точки зрения человека, желательно, чтобы описательные возможности используемой модели были как можно выше. С другой стороны, сложное представление знаний требует специальных способов обработки (усложняется механизм вывода), что затрудняет проектирование и реализацию системы хранения знаний [4].

В настоящее время используется семь классов моделей знаний: логические, продукционные, фреймовые, сетевые, объектно-ориентированные, специальные и комплексные [5].

Для определения модели представления знаний, которую необходимо применить для создания базы знаний, входящей в состав СППР для эксплуатации роботизированных паркингов, необходимо рассмотреть используемые классы моделей знаний и произвести обоснованный выбор наиболее подходящей модели.

Обзор существующих решений

На сегодняшний день представлено множество СППР для различных областей знаний. Существуют обобщенные методики построения СППР для решения задач, выдвигаемых различными видами деятельности [6]. Однако, данные методики не включают в себя задачи выбора моделей знаний для построения БЗ в связи с тем, что СППР должна носить индивидуальный характер и учитывать особенности конкретного технического комплекса, управления технологическими процессами и организационной структуры, обслуживающей эти процессы [2].

Существуют методики для построения СППР для управления эксплуатацией сложных технических комплексов [7, 2, 8, 9, 10]. Между тем, в рассмотренных работах предлагаются различные модели знаний в зависимости от специфики использования СППР.

Работ по построению СППР для управления эксплуатацией роботизированных паркингов и методов выбора моделей знаний для построения такого рода СППР в открытом доступе не представлено.

Цели работы

В связи с данными, полученными на основании проведенного обзора, целью данной работы является определение модели представления знаний, которая может быть применена для построения БЗ, входящей в состав СППР для эксплуатации роботизированных паркингов.

Решаемые задачи

В данной работе будет проведен обзор наиболее распространенных моделей знаний, включающий в себя описание достоинств и недостатков каждой рассматриваемой модели. На основе полученного обзора будет проведен выбор модели знаний, наиболее подходящей для построения БЗ используемой в рамках СППР для эксплуатации роботизированных паркингов.

Синтез модели представления знаний

Применяемые на сегодняшний день классы моделей знаний имеют свои преимущества и недостатки. Для выбора модели представления знаний в БЗ для СППР для эксплуатации роботизированных паркингов необходимо произвести обзор всех используемых на текущий момент моделей, произвести выбор критериев отбора и произвести выбор наиболее подходящей модели знаний.

В логических моделях знания представляются в виде совокупности правильно

построенных формул какой-либо формальной системы (ФС), которая задается четверкой

$$(T, P, A, R) \quad (1)$$

где T – множество базовых (терминальных) элементов, из которых формируются все выражения; P – множество синтаксических правил, определяющих синтаксически правильные выражения из терминальных элементов ФС; A – множество аксиом ФС, соответствующих синтаксически правильным выражениям, которые в рамках данной ФС априорно считаются истинными; R – конечное множество отношений $\{r_1, r_2, \dots, r_n\}$ между формулами, называемыми правилами вывода, позволяющих получать из одних синтаксически правильных выражений другие.

Для знания, входящих в базу знания, можно считать, что множество A образуют все информационные единицы, которые введены в базу знания извне, а с помощью правил вывода из них выводятся новые производные знания. Другими словами формальная система представляет собой генератор порождения новых знания, образующих множество выводимых в данной системе знания. Данное свойство логических моделей широко используется в базах знания. Оно позволяет хранить в базе лишь те знания, которые образуют множество A , а все остальные знания получать из них по правилам вывода [11].

Наиболее распространенной логической моделью, используемой при разработке БЗ, является исчисление предикатов I-го порядка, когда предметная область или задача описываются в виде набора аксиом. Чаще всего эти логические модели строятся при помощи декларативных языков логического программирования, наиболее известным представителем которых является язык Пролог (Prolog) [12].

Достоинствами логической модели представления знаний являются:

- единственность теоретического обоснования и возможность реализации системы формально точных определений и выводов [13];

- согласованность знания как единого целого, облегчающая решение проблем верификации БЗ, оценки независимости и полноты системы аксиом и т.д.;

- единые средства описания как знания о предметной области, так и способов решения задач в этой предметной области, что позволяет любую задачу свести к поиску логического вывода некоторой формулы в той или иной ФС [5];

Такое единообразие и простота формализации влечет за собой основной недостаток модели – сложность использования в процессе логического вывода эвристик, отражающих специфику предметной области. При решении сложных задач попытка представить неформализованные знания эксперта, среди которых преобладают эвристики, в системе строгой логики наталкивается на серьезные препятствия. Это связано с тем, что, в отличие от строгой логики, так называемая «человеческая логика» обладает нечеткой структурой. Поэтому большая часть достижений в области систем с базами знания до настоящего момента была связана с применением нелогических моделей [13].

Кроме того, можно выделить следующие недостатки использования данной модели:

- представление знания в таких моделях ненаглядно: логические формулы трудно читаются и воспринимаются;

- ограничения исчисления предикатов первого порядка не допускают квантификации предикатов и использования их в качестве переменных;

- обоснованность обозначения свойств и отношений однотипными пропозиционными функциями вызывает сомнения [14, 15];

- описание знания в виде логических формул не позволяет проявиться преимуществам, которые имеются при автоматизированной обработке структур данных [5].

- возможность возникновения «комбинаторного взрыва» при формировании БЗ;

- слабость структурированности описаний. [11]

Продукционные модели можно считать наиболее распространенными моделями представления знания. Продукционная модель – это модель, основанная на правилах, позволяющая представить знание в виде предложения типа: «Если условие, то действие»

[13].

Продукционные модели состоят из множества продукции или правил вывода. Каждая такая продукция в общем виде может быть представлена выражением

$$(W_i, U_i, P_i, A_i \rightarrow B_i, C_i), \quad (2)$$

где W_i – сфера применения i -й продукции, определяющая класс ситуации в некоторой предметной области (или фрагменте рабочей предметной области), в которых применение данной продукции правомерно; U_i – предусловие i -и продукции, содержащее информацию об истинности данной продукции, ее значимости относительно прочих продукции и т.п.; P_i – условие i -и продукции, определяемое факторами, непосредственно не входящими в A_i истинностное значение которого разрешает применять данную продукцию; $A_i \rightarrow B_i$ – ядро i -й продукции, соответствующее правилу «если..., то...»; C_i – постусловие i -и продукции, определяющее изменения, которые необходимо внести в систему продукции после выполнения данной продукции [5].

Системы, основанные на продукционной модели, состоят из трех типовых компонентов: базы правил (продукции), базы фактов, содержащей декларативные знания о предметной области, используемые в качестве аргументов в условиях применимости продукции, и интерпретатора продукции, реализующего функции анализа условия применимости продукции, выполнения продукции и управления выбором продукции (управления выводом в продукционной системе). [3]

Продукционная модель чаще всего применяется в промышленных экспертных системах. Она привлекает разработчиков своей наглядностью, высокой модульностью, легкостью внесения дополнений и изменений и простотой механизма логического вывода [12].

Реализация логических и продукционных моделей знаний базируется на языках типа ПРОЛОГа.

Основные преимущества продукционных систем:

- простота и гибкость выделения знаний;
- отделение знаний от программы поиска;
- модульность продукционных правил (правила не могут «вызывать» другие правила);
- возможность эвристического управления поиском;
- возможность трассировки «цепочки рассуждений»;
- независимость от выбора языка программирования;
- продукционные правила являются правдоподобной моделью решения задачи человеком [11].

Недостатком продукционной модели является то, что при накоплении достаточно большого количества (порядка нескольких сотен) продукции они начинают противоречить друг другу. Рост противоречивости продукционной модели может быть ограничен путем введения механизмов ограничения и возвратов. Механизм исключения означает, что вводятся специальные правила-исключения. Их отличает большая конкретность в сравнении с обобщенными правилами. При наличии исключения основное правило не применяется. Механизм возвратов означает, что логический вывод может продолжаться даже в том случае, если на каком-то этапе вывод привел к противоречию: просто необходимо отказаться от одного из принятых ранее утверждений и осуществить возврат к предыдущему состоянию [12].

Кроме этого, недостатками продукционной модели являются:

- сложность управления выводом, неоднозначность выбора конкурирующих правил;
- низкая эффективность вывода в целом, негибкость механизмов вывода;
- неоднозначность учета взаимосвязи отдельных продукции;

- несоответствие психологическим аспектам представления и обработки знания человеком;
- сложность оценки целостного представления предметной области [5].

Основой **фреймовой модели знания** служит понятие фрейма — структуры данных, представляющей некоторый концептуальный объект или типовую ситуацию. Фрейм идентифицируется уникальным именем и включает в себя множество слотов. В свою очередь, каждому слоту соответствует определенная структура данных. В слотах описывается информация о фрейме: его свойства, характеристики, относящиеся к нему факты и т.д. Кроме того, слоты могут содержать ссылки на другие фреймы или указания на ассоциируемые с ними присоединенные процедуры. Представление предметной области в виде иерархической системы фреймов хорошо отражает внутреннюю и внешнюю структуры объектов этой предметной области [5].

Модель фрейма является достаточно универсальной, поскольку позволяет отобразить все многообразие знаний о мире через:

- фреймы-структуры, для обозначения объектов и понятий (заем, залог, вексель);
- фреймы-роли (менеджер, кассир, клиент);
- фреймы-сценарии (банкротство, собрание акционеров, празднование именин);
- фреймы-ситуации (тревога, авария, рабочий режим устройства) и др.

Различают *фреймы-образцы*, или *прототипы*, и *фреймы-экземпляры*, которые создаются для отображения реальных фактических ситуаций на основе поступающих данных.

Традиционно структура фрейма может быть представлена как список свойств:

(ИМЯ ФРЕЙМА:

(имя 1-го слота: значение 1-го слота),

(имя 2-го слота: значение 2-го слота),

.....

(имя N-го слота: значение N-го слота)).

В качестве значения слота может выступать имя другого фрейма – так образуются сети фреймов.

Реализация фреймовой модели знания базируется на языках линии LISP, FRL, KRL [5, 11.1].

Важнейшим свойством теории фреймов является заимствование из теории семантических сетей — так называемое *наследование свойств*. И во фреймах, и в семантических сетях наследование происходит по *АКО-связям* (*A-Kind-Of = это*) Слот АКО указывает на фрейм более высокого уровня иерархии, откуда неявно наследуются, то есть переносятся, значения аналогичных слотов. [3]

Основным преимуществом фреймовой модели являются ее наглядность, гибкость, однородность, высокая степень структуризации знания, соответствие принципам представления знания человеком в долговременной памяти, а также интеграция декларативных и процедурных знания [5].

Вместе с тем, для фреймовой модели характерны сложность управления выводом и низкая эффективность его процедур [5].

Во фреймовых моделях фиксируется жесткая структура информационных единиц [16]. Это существенно снижает гибкость такой модели.

Наилучший метод представления знания реализован как **сетевая модель знания**. В качестве носителя знания в этой модели выступает семантическая сеть, вершины которой соответствуют объектам (понятиям), а дуги – отношениям между понятиями. Кроме того, и вершинам, и дугам присваиваются имена (идентификаторы) и описания, характеризующие семантику объектов и отношения предметной области.

Семантические сети – наиболее мощный класс математических моделей для представления знания о предметной области, одно из важнейших направлений ИИ. В общем случае под семантической сетью понимается структура

$$S = (O, R), \quad (3)$$

где $O = \{O_i, i=1, n\}$ – множество объектов предметной области; $R = \{R_j, j=1, k\}$ – множество отношений между объектами.

Семантическая сеть – это ориентированный граф, вершины которого – понятия, а дуги – отношения между ними. В качестве понятия обычно выступают абстрактные или конкретные объекты, а отношения – это связи.

Разновидностью сетевой модели является онтология. Быстрое развитие онтологического подхода в последние годы обусловлено распространением Internet-технологии, использующих онтологические модели [5].

Для реализации семантических сетей существуют специальные сетевые языки, например NET, язык реализации систем SIMER+MIR и др. Широко известны экспертные системы, использующие семантические сети в качестве языка представления знаний – PROSPECTOR, CASNET, TORUS.

Недостатком данной модели представления знания является сложность организации процедуры поиска вывода на семантической сети [12].

Система, выполненная с помощью рассматриваемой модели, является целостной и не позволяет разделить базу знаний и механизм выводов [17].

Кроме того, если допускать в сетевой модели произвольные типы отношений и связей, не являющихся отношениями в математическом смысле (например, ассоциативные связи), то резко возрастает сложность работы с такой моделью. Также построение сетевой модели зависит от взгляда на проблему самого разработчика. Поэтому разные инженеры по знаниям могут спроектировать разные модели одной предметной области, что может сказываться негативно на реализации процедур обработки знаний [4].

Объектно-ориентированная модель знания получила широкое применение в современных технологиях проектирования разнообразных программных и информационных систем. В настоящее время существуют два основных подхода к моделированию знания, базирующихся на объектной парадигме. Это модель MDA (Model Driven Architecture) [21] консорциума Object Management Group (OMG) и модель ODP (Model of Open Distributed Processing), зафиксированная в стандарте ISO/ITU [18-21].

Особенностью данной модели знаний является тот факт, что она была разработана и используется в основном для построения БЗ для СППР для помощи при программировании [22].

Класс **специальных моделей знания** объединяет модели, отражающие особенности представления знания и решения задач в отдельных, относительно узких предметных областях. Характеристики ряда специальных моделей даны в [14,15,20]. В качестве примера подобного способа формализации знания можно привести модель представления предметной области «объект-признак», используемую в автоматизированных системах поиска аналогов и построения классификации [20]. Применение на практике того или иного способа формализации обуславливается спецификой задачи, для решения которой планируется использовать БЗ.

Наиболее перспективны смешанные или **комплексные модели**, интегрирующие преимущества рассмотренных выше базовых моделей представления предметной области [5].

Были рассмотрены основные модели представления знаний, используемые на сегодняшний день. На практике при разработке системы редко удаётся обойтись рамками одной модели представления знаний за исключением самых простых случаев [23].

На основании проведенного обзора можно сделать вывод, что для построения базы знаний для построения СППР при управлении эксплуатацией роботизированных паркингов необходимо использовать сочетание продукционной и логической моделей. Продукционная модель была выбрана в связи с большим количеством неформальных записей и запросов от операторов паркингов, и отсутствием формальных связей внутри БЗ в связи с высокой сложностью объектов управления, наглядностью, высокой модульностью, легкостью внесения дополнения и изменения и простотой механизма логического вывода. Логическая модель была выбрана в связи с большим объемом предполагаемой БЗ, в связи с большим

количеством объектов управления и большим количеством составных частей каждого из объектов. Совместное использование предлагаемых моделей позволяет избавиться от части недостатков каждой из моделей в отдельности:

- снижаются требования к качеству и полноте хранящихся знаний, предъявляемые к логической модели;
- увеличивается наглядность представления знаний в связи с использованием продукционной модели;
- снижается противоречивость продукционной модели;
- увеличивается эффективность обработки продукции.

Заключение

В данной статье рассмотрена проблема построения базы знаний, используемой в рамках СППР для управления эксплуатацией роботизированных паркингов. В связи с тем, что база знаний является основой для дальнейшей работы над СППР, в данной статье было важно определить способ построения рассматриваемой БЗ. Метод построения БЗ в первую очередь, зависит от выбора модели представления знаний. В связи с этим был проведен обзор и сравнение существующих моделей представления знаний для последующего выбора наиболее подходящей модели представления знаний.

Для реализации базы знаний для СППР для эксплуатации роботизированных паркингов выбрано сочетание продукционной и логической моделей. Такое представление знаний обладает рядом преимуществ: снижаются требования к качеству и полноте хранящихся знаний, предъявляемые к логической модели; увеличивается наглядность представления знаний в связи с использованием продукционной модели; снижается противоречивость продукционной модели; увеличивается эффективность обработки продукции.

Предложенная модель представления знаний позволяет построить БЗ, наиболее подходящую для использования в рамках СППР для эксплуатации роботизированных паркингов в связи с высокой эффективностью работы с большими объемами информации, простотой расширения полученной БЗ и возможностью обработки и хранения неформальных знаний экспертов, которые необходимо обрабатывать в связи с высокой сложностью объекта управления.

Литература

1. Михаленко В. А. Система поддержки принятия решений при эксплуатации магистральных газопроводов : дисс. Михаленко В. А. канд. технич. Наук : 05.13.06. - Тула. - 2000. - 192 с.
2. Постников Е. В. Построение систем поддержки принятия решений в процессах управления эксплуатацией технических комплексов / Постников Е. В., Романенко Д. А., Беляев С. А., Павловский М. Г. // Компьютерные и информационные технологии в науке и образовании - 2000. – спец. выпуск - С. 79-83.
3. Гаврилова Т.А. Базы знаний интеллектуальны систем / Гаврилова Т.А., Хорошевский В.Ф. – СПб.: Питер, 2000. - 384 С.
4. Елисеев Д. В. Модель представления знаний при создании адаптивной информационной системы [Электронный ресурс] / Елисеев Д. В // Наука в образовании: Электронное научное издание. - 2010. - №03. - М. : Наука в образовании: Электронное научное издание, 2010. - Режим доступа : <http://technomag.edu.ru/doc/139659.html>, свободный. – Загл. с экрана.
5. Сеченов М.Д. Анализ неформальных моделей представления знаний в системах принятия решений / Сеченов М.Д., Щеглов С.Н. // Известия южного федерального университета. Технические науки. - 2010. - №7. - С. 135-140.
6. Симанков В. С. Методологические аспекты построения систем поддержки принятия

решении / Симанков В. С., Владимиров С. Н., Денисенко А. О., Черкасов А. Н. // Вестник донского государственного технического университета. - 2008. - №3. - С. 258-267.

7. Черняховская Л. Р. Поддержка принятия решения при управлении сложными производственными системами на основе онтологической базы знаний / Черняховская Л. Р., Старцева Е. Б., Муксимов П. В., Макаров К. А. // Вестник УГАТУ. Управление, ВТ и И. - 2007. - №7(25). - С. 41-46.

8. Загорулько Ю. А. Подход к разработке системы поддержки принятия решений на примере нефтегазодобывающего предприятия / Загорулько Ю. А., Ануреев И. С., Загорулько Г. // Известия ТПУ. - 2010. - №5. - С. 127-131.

9. Герасимов Б. М. Система поддержки принятия решения в АСУ реального времени / Герасимов Б. М., Глуцкий В. И., Рабчун А. А. // Журнал «Штучний інтелект». - 2000. - №3. - С. 39-47.

10. Иванов С.В. Особенности проектирования и разработки высокопроизводительных систем поддержки принятия решений для управления сложными техническими объектами / Иванов С. В., Нечаев Ю. И., Бухановский А. В. // XVII всероссийская научно-методическая конференция "телематика'2010". - Секция: D. Технологии распределенных вычислений и компьютерного моделирования в образовании и науке. - 2010.

11. Смагин А. А. Интеллектуальные информационные системы. Учебное пособие / Смагин А. А., Липатова С. В., Мельниченко А. С. - Ульяновск Ульяновский государственный университет, 2010. - 137 С.

12. Советов Б. Я. Представление знаний в информационных системах : учебник для студ. учреждений высш. проф. образования / Советов Б. Я., Цехановский В. В., Чертовской В. Д. - 2-е изд., стер. - М.: Издательский центр "Академия", 2012. - 144 С.

13. Белоус Е. С. Современные модели представления знаний в обучающих системах / Белоус Е. С., Кудинов В. А., Желнин М. Э. // Ученые записки. Электронный научный журнал курского государственного университета. - 2010. - №1. - С. 9-14.

14. Джексон П. Введение в экспертные системы. – М.: Издательский дом «Вильямс», 2001. - 624 с.

15. Городецкий В.И., Самойлов В.В., Малов О.А. Современное состояние технологии извлечения знаний из баз и хранилищ данных. Ч.II // Новости искусственного интеллекта. – 2002. – N4. – С. 3-9.

16. Искусственный интеллект. – В 3-х кн. Кн. 2. Модели и методы: Справочник / Под ред. Д.А. Пospelова – М.: Радио и связь, 1990. – 304 с.

17. Представление и использование знаний: Пер. с япон./Под ред. Х. Уэно, М. Исидзука. - М.: Мир, 1989. - 220 С.

18. Осипов Г.С. Построение моделей предметных областей. Ч. I. Неоднородные семантические сети // Известия РАН. Техническая кибернетика. – 1990. – N5. – С. 32-35.

19. Городецкий В.И., Самойлов В.В., Малов О.А. Современное состояние технологии извлечения знаний из баз и хранилищ данных. Ч. I // Новости искусственного интеллекта. – 2002. – N3. – С. 3-12.

20. Городецкий В.И., Самойлов В.В., Малов О.А. Современное состояние технологии извлечения знаний из баз и хранилищ данных. Ч. II // Новости искусственного интеллекта. – 2002. – N 4. – С. 3-9.

21. Щеглов С.Н. Анализ процесса подготовки и принятия решений в условиях «нечеткой» определенности информационного потока // Труды конгресса по интеллектуальным системам и информационным технологиям «AIS– IT'09». Научное издание в 4-х томах. – М.: Физматлит, 2009, Т.1. – С. 225-233.

22. Сеченов М. Д. Язык программирования в экспертных системах / Сеченов М. Д., Щеглов С. Н. // Известия ТРТУ. - 1996. - С. 92-99.

23. Андрейчиков А.В., Андрейчикова О.Н. Интеллектуальные информационные системы. – М.: Финансы и статистика, 2004. – 424 С.

ВЛИЯНИЕ РАЗМЕРА ШАГА В АЛГОРИТМЕ ПОИСКА КУКУШКИ ДЛЯ РЕШЕНИЯ ЗАДАЧ ГЛОБАЛЬНОЙ БЕЗУСЛОВНОЙ ОПТИМИЗАЦИИ

Бенза Н.Н.

Научный руководитель: д.ф.-м.н., профессор, Карпенко А.П.

МГТУ им. Н.Э.Баумана, кафедра РК6, Москва, Россия

INFLUENCE OF STEP SIZE IN 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 длины шага. Кратко представлено использование в алгоритме поиска кукушки полетов Леви - разновидности случайных блужданий. 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 step size 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

Поиск кукушки – это алгоритм оптимизации, разработанный Xin-she Yang и Susan Deb в 2009 году [1]. На его создание авторов вдохновило поведение кукушек в процессе так называемого вынужденного гнездового паразитизма, когда некоторые виды кукушек подкладывают яйца в гнезда птиц других видов [2].

Поисковый метод кукушки теоретически описывает подобное поведение размножения, и таким образом может применяться для различных задач оптимизации.

1 Постановка задачи и схема алгоритма

Рассматриваем задачу глобальной безусловной минимизации вида

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

где x - n -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 [3]. 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 яйца, 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.

2 Полеты Леви

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

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

где α – вектор размеров шагов, $\alpha_i > 0$; $L(\lambda)$ – вектор направления движения, $L_i(\lambda) = Lévy(\lambda)$ – распределение Леви. Это уравнение является стохастическим уравнением для случайного блуждания.

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

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

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 длины шага α для задачи (1). В докладе представлены результаты численных экспериментов при поиске минимума функции Растригина, которая имеет вид

$$F(x) = 10n + \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i)).$$

Результаты исследования иллюстрируют рисунки 1, 2 и таблица 1, на которых принято, что n_f – число испытаний (вычислений значений целевой функции), $nests$ – число гнезд.

Рассмотрим влияние параметра α на вычисления. Для этого возьмем функцию Растригина и будем менять длину шага α от 0,01 до 2,00. Результаты показаны на рисунке 1.

Поскольку алгоритм является стохастическим, то для анализа его работы используем методы математической статистики.

Введем некоторые определения. Квантиль в математической статистике — значение, которое заданная случайная величина не превышает с фиксированной вероятностью:

- 0,25-квантиль называется первым (или нижним) квартилем (от лат. *quarta* — четверть);
- 0,5-квантиль называется медианой (от лат. *mediāna* — середина) или вторым квартилем;
- 0,75-квантиль называется третьим (или верхним) квартилем.

Интерквартильным размахом (англ. *interquartile range*) называется разность между третьим и первым квартилями. Интерквартильный размах является характеристикой разброса распределения величины.

На рисунке 1 белая линия соответствует медиане, черная область – это интерквартильный размах, в нем заключено 50% всех полученных значений. Серая область включает 95% всех значений.

В исходном алгоритме значение $\alpha = 0,01$. Очевидно, что из-за этого алгоритм совершает слишком маленькие шаги и часто сваливается в локальный минимум, однако после

разрушения некоторого числа гнезд, которое определяется параметром p_a , он все-таки находит глобальный минимум. Примем $\alpha=1,00$ для дальнейших вычислений.

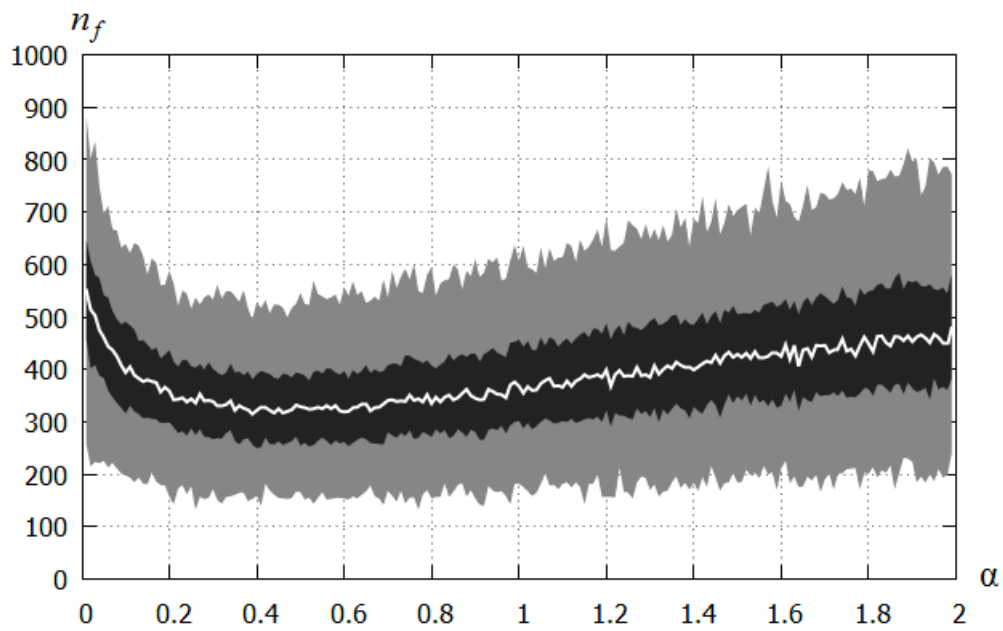


Рисунок 1 – Зависимость числа вычислений n_f целевой функции от параметра α : функция Растргина; $n=2$; $nests=8$; $p_a=0,3$

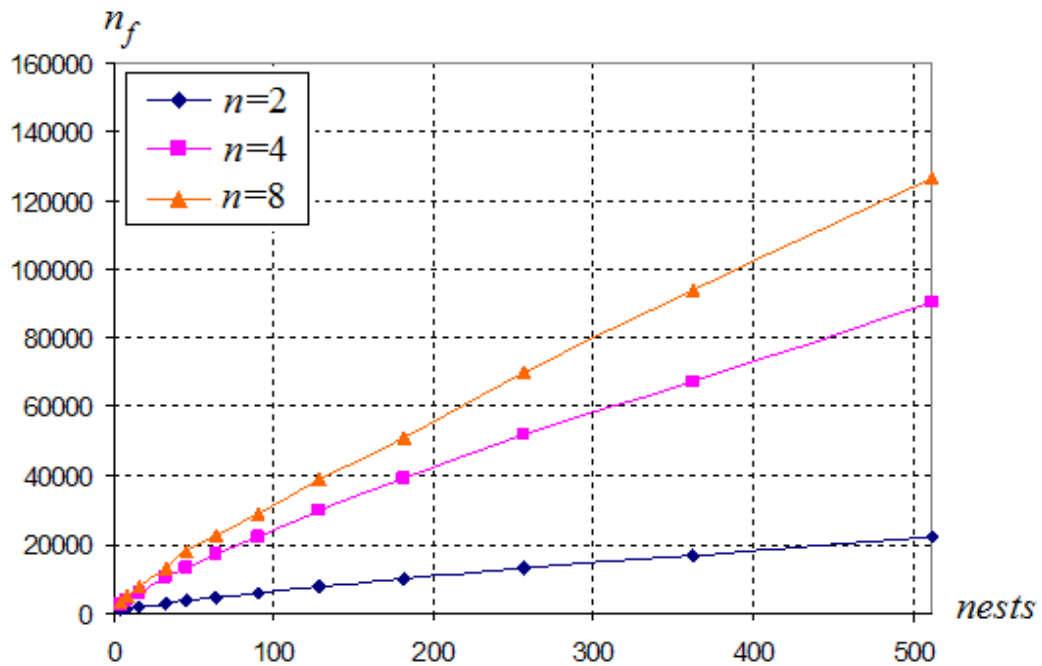


Рисунок 2 – Зависимость числа вычислений n_f целевой функции от числа гнезд $nests$: функция Растргина; длина шага $\alpha=1,00$; $p_a=0,3$

Таблица 1 – Оценка вероятности нахождения минимума функции Растригина при различных значениях параметров $nests$ и n : $p_a=0,3$

$nests$	$n=2$	$n=4$	$n=8$
4	87	42	2
8	99	98	45
16	100	100	98
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
362	100	100	100
512	100	100	100

Рисунки показывают, что при увеличении числа гнезд возрастает число вычислений значений целевой функции. При увеличении размерности пространства n заметно возрастает суммарное число испытаний.

По данным таблицы 1, в которой показано распределение оценки вероятности нахождения минимума тестовых функций, можно сделать выводы о выборе оптимального значения параметра $nests$.

Вероятность нахождения минимума функции Растригина не одинакова для всех размерностей и соответствующих значений параметра $nests$. Для $n=2$ и $n=4$ оценка вероятности нахождения минимума равна 100% при выборе числа гнезд больше восьми. Для $n=8$ следует выбирать значение $nests$ больше 16 гнезд.

Заключение

В работе проведено исследование влияния длины шага в алгоритме поиска кукушки. При выборе слишком маленького значения параметра α увеличивается число вычислений целевой функции. Для сложной многоэкстремальной функции Растригина оптимальным можно считать значение $\alpha=1,00$.

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 (IJAI), 2011, Vol.2, No.3, pp. 36-43.

АРХИТЕКТУРА ТОНКОГО КЛИЕНТА ДЛЯ СОЗДАНИЯ СХЕМ РАЗЛИЧНОЙ ФИЗИЧЕСКОЙ ПРИРОДЫ

Бочаров Ф.А.

Научный руководитель: доцент, к. т. н., Волосатова Т.М.

МГТУ им. Н.Э.Баумана, кафедра РКБ, Москва, Россия

THIN-CLIENT TO CREATE A CIRCUIT OF DIFFERENT PHYSICAL NATURE

Bocharov F.A.

Supervisor: Associate Professor, Ph.D., Volosatova T.M.

MSTU, Moscow, Russia

Аннотация

В статье рассматриваются принципы построения тонкого клиента для создания и редактирования схем различной физической природы. Подробно рассмотрены особенности реализации приложения. Кратко представлено сравнение возможностей современных технологий создания тонкого клиента. В заключении представлены рекомендации по выбору технологий.

Abstract

The article describes the principles of the thin client to create and edit circuits of different physical nature. Detail the features of the implementation application. Summarizes the comparison of the capabilities of modern technology to create a thin client. In conclusion, presented recommendations for the choice of technology.

Введение

Целью работы является создание редактора схем различной физической природы. Реализация приложения в виде тонкого клиента позволяет достичь кроссплатформенности, кроссбраузерности. Тонкий клиент в компьютерных технологиях — это компьютер или программа-клиент, который переносит все или большую часть задач по обработке информации на сервер. Примером тонкого клиента может служить компьютер с браузером, использующийся для работы с веб-приложениями. Кроссплатформенность— свойство программы работать более чем на одной аппаратной платформе или операционной системе. Кроссбраузерность — свойство веб-приложения отображаться и работать во всех популярных браузерах идентично. Под идентичностью понимается отсутствие существенных различий внешнего вида графического интерфейса, а так же его одинаковое поведение и логика работы. Достоинством данной архитектуры является уменьшение нагрузки на клиентскую часть приложения и уменьшение системных требований.

Построение схем является неотъемлемой частью инженерных задач. Автоматизация данного процесса позволяет ускорить их решение, снизить затраты и уменьшить количество ошибок. Разработка приложения велась с использованием систем командной разработки, таких как система контроля версий.

Инновационным подходом в создании приложений подобной направленности является клиент-серверная архитектура. Большинство существующих аналогов требуют наличия дополнительного программного обеспечения для корректной работы.

Архитектура приложения WebCAD

Приложение WebCAD имеет клиент-серверную архитектуру. Текущая реализация приложения является эволюцией предыдущей версии WebCAD, использующей JavaScript и HTML5 в качестве клиентских технологий. Недостатком предыдущей версии являлась поддержка ограниченного числа браузеров, меньшая скорость работы. Клиент-сервер — вычислительная или сетевая архитектура, в которой задания или сетевая нагрузка распределены между поставщиками услуг (сервисов), называемыми серверами, и

заказчиками услуг, называемыми клиентами. Нередко клиенты и серверы взаимодействуют через компьютерную сеть и могут быть как различными физическими устройствами, так и программным обеспечением. Среди прочих, положительными сторонами данного подхода являются отсутствие дублирования кода программы-сервера программами-клиентами.

Клиент

Клиент представляет собой приложение, написанное с использованием технологии Silverlight. Плагин Silverlight реализован под большинство современных браузеров, он позволяет запускать приложение. Microsoft Silverlight — это программная платформа, включающая в себя плагин для браузера, который позволяет запускать приложения, содержащие анимацию, векторную графику и аудио-видео ролики, что характерно для RIA (Rich Internet application).

Сервер

Серверная часть реализована с помощью технологии WCF (Windows Communication Foundation). WCF - платформа для обмена данными между приложениями входящими в состав .NET Framework. Серверная часть представляет собой совокупность системы управления базами данных MS SQL2008 R2 и веб-сервис для предоставления доступа к данным. Для представления данных в приложении используется объектно-ориентированная технология доступа к данным Entity Framework. ADO.NET Entity Framework — объектно-ориентированная технология доступа к данным, является object-relational mapping (ORM) решением для .NET Framework от Microsoft. Она позволяет сохранять объекты, используемые приложением в различных системах управления базами данных. За счет использования ORM программный код приложения значительно меньше связан с конкретной базой данных, что позволяет, при необходимости, осуществить миграцию на другую СУБД.

Сеанс работы с WebCAD

При открытии пользователем веб-страницы приложения браузер скачивает код приложения Silverlight на свой компьютер. С помощью плагина Silverlight браузер запускает скаченное приложение. Приложение Silverlight обращается к веб-сервису для получения данных инициализации. После авторизации в системе приложение запрашивает веб-сервис о существующих чертежах пользователя.

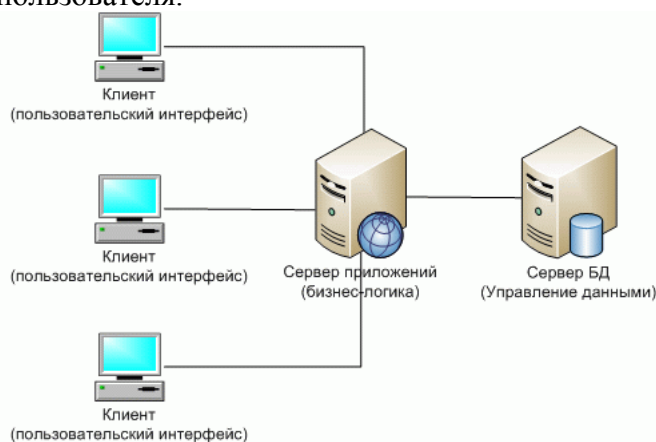


Рисунок 1 – Клиент-серверная архитектура

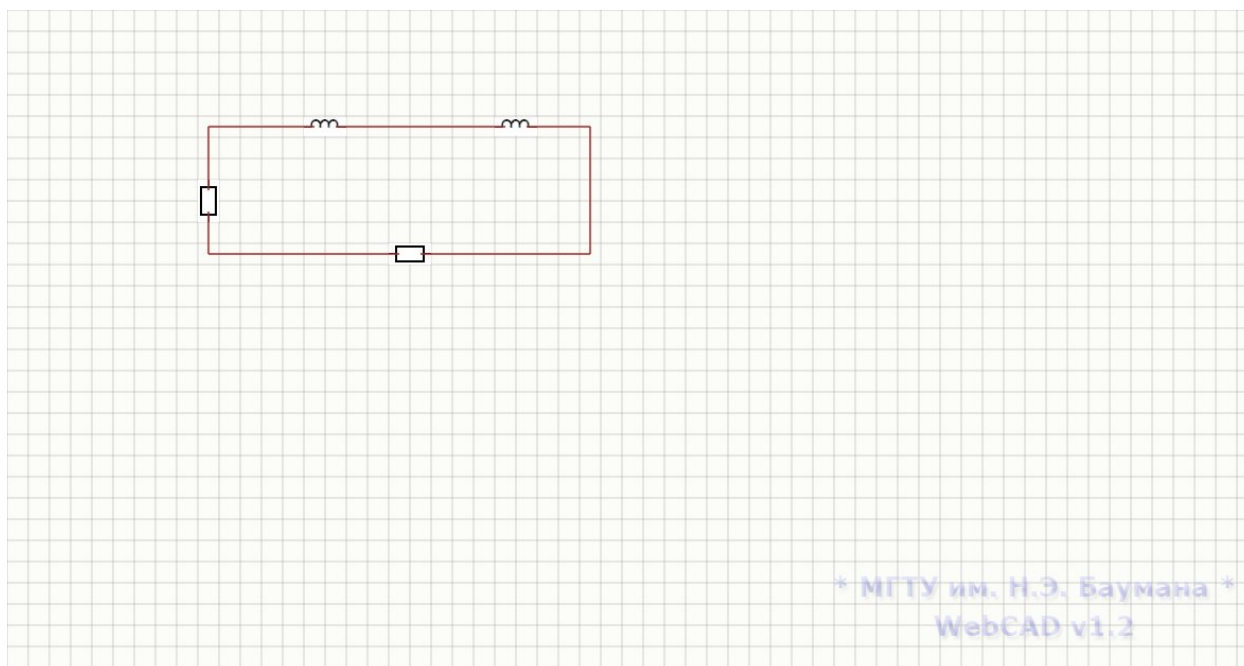


Рисунок 2 – Графический интерфейс пользователя

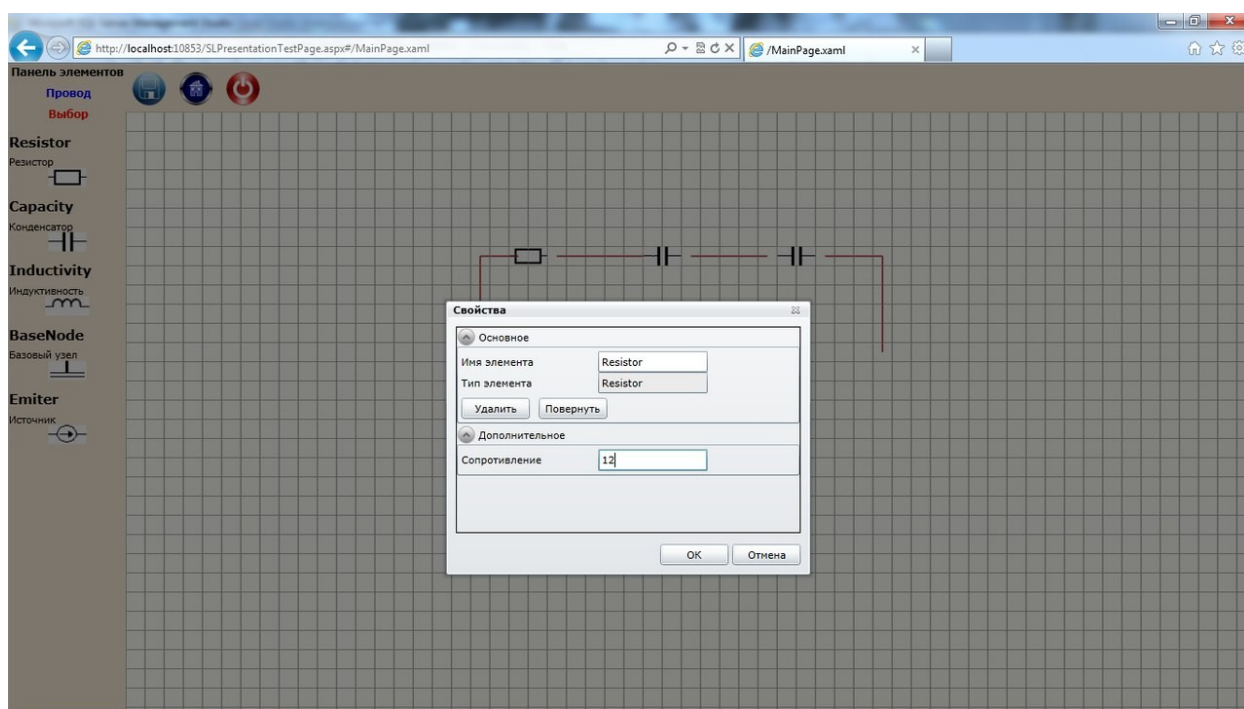


Рисунок 3 - Окно свойств элемента

Веб-сервис обращается к базе данных путем SQL-запросов и получает необходимые данные. Данные получает клиент. При открытии чертежа действует аналогичная схема взаимодействия клиента и сервера.

Особенности реализации

Одной из особенностей приложения является его модульная архитектура. Использование данной архитектуры позволяет добавлять новую функциональность без существенной переработки приложения. На базе приложения WebCAD разработано несколько модулей с реализацией стеганографических методов.

В приложении реализована система авторизации, защищающая пользователей от несанкционированного доступа. Для автоматизации процесса создания чертежа конечным пользователем в приложении реализована поддержка макросов. Пользователь имеет возможность автоматизировать свои действия, создавая макросы на языке IronPython. Макросы вводятся в специальном консольном окне ввода и могут взаимодействовать с объектами чертежа, такими как элементы схемы и графический интерфейс пользователя. Такой подход позволяет избавить пользователя от выполнения рутинных задач. IronPython – реализация скриптового языка высокого уровня Python для платформы Microsoft .NET Framework.

Еще одной особенностью является возможность сохранения чертежа в формате XML (eXtensible Markup Language, расширяемый язык разметки). Документы в данном формате содержат подробную информацию о своем содержимом, а главное – формат XML распознает множество программ и браузеров на всех существующих операционных системах. XML — текстовый формат, предназначенный для хранения структурированных данных (взамен существующих файлов баз данных), для обмена информацией между программами, а также для создания на его основе более специализированных языков разметки (например, XHTML).

Заключение

В результате данной работы было реализовано приложение WebCAD. Основным отличием системы является независимость от платформы, невысокие системные требования, модульность приложения. В отличие от другого программного обеспечения с подобным функционалом, пользователю не требуется сторонних приложений для работы, кроме веб-браузера с установленным плагином Silverlight. Поскольку все данные о чертежах хранятся на сервере, пользователь может редактировать или просматривать свои чертежи на любом устройстве. При узком канале связи скорость работы приложения WebCAD может уступать аналогам. В рамках данного проекта поставленные задачи были выполнены. Более того, имеется техническая возможность для расширения функционала за счет модульности приложения.

Литература

1. Троелсен Э., Язык программирования C# 2010 и платформа .NET 4.0. — М.: Вильямс, 2011. 1344 с.
2. Макдональд М., Microsoft ASP.NET 4 с примерами на C# 2010 для профессионалов. — М.: Вильямс 2010. 1424 с.
3. Конахович Г.Ф., Пузыренко А. Ю., Компьютерная стеганография. Теория и практика. — М.: мк-Пресс, 2006. 288 с.
4. Lowy J., Programming WCF Services — O'Reilly Media, 2008, 784 с.
5. Brown P., Silverlight 4 in Action — Manning Publications, 2010, 800 с.

СОЗДАНИЕ И ВНЕДРЕНИЕ ЦИФРОВОГО ВОДЯНОГО ЗНАКА НА ОСНОВЕ СТЕГАНОГРАФИЧЕСКОГО АЛГОРИТМА ЧАЙА

Булгакова В.А., Захаров А.В.

Научный руководитель: к.т.н., доцент Волосатова Т.М.

МГТУ им. Н.Э.Баумана, кафедра РК6, Москва, Россия

CREATION AND EMBEDDING OF DIGITAL WATERMARK ON THE BASIS OF CHAE STEGANOGRAPHIC ALGORITHM

Bulgakova V.A., Zakharov A.V.

The scientific chief: Ph.D., docent Volosatova T.M.

BMSTU, RK6, Moscow, Russia

Аннотация

В работе рассмотрен один из методов внедрения цифрового водяного знака в изображение. В этом алгоритме внедряется черно-белое изображение (логотип), размером до 25 % от размеров исходного изображения. В соответствии с этим алгоритмом была написана программа, имеющая успешные результаты.

Abstract

This paper presents one of the methods of embedding a digital watermark in image. This algorithm includes an embedding of grayscale image (logo), that could be up to 25% of the host image. According to this algorithm was written program with successful results.

Введение

Сегодня существует два основных направления решения задачи защиты информации от несанкционированного доступа: криптография и стеганография. Цель криптографии – сокрытие содержимого сообщения за счет их шифрования. В отличие от этого, при стеганографии (от греч. «тайнопись») скрывается сам факт существования тайного сообщения.

Тайнопись осуществляется различными способами, общей чертой которых является то, что скрываемое сообщение встраивается в некоторый безобидный, не привлекающий внимания объект, который затем открыто транспортируется адресату.

Развитие средств вычислительной техники в последнее время дало новый толчок развитию компьютерной стеганографии. Сообщения встраивают в цифровые данные, как правило, имеющие аналоговую природу – речь, аудиозаписи, изображение, видео и даже текстовые файлы и исполняемые файлы программ, преобразованные в цифровую форму.

Основными причинами популярности исследований в области стеганографии в настоящее время являются:

1. ограничение на использование криптосредств в ряде стран мира, что влечет за собой большое количество исследований в духе классической стеганографии, то есть сокрытия факта передачи информации;
2. появление проблемы защиты прав собственности на информацию, представленную в цифровом виде, которое влечет работы в области так называемых водяных знаков.

Цифровой водяной знак (ЦВЗ) – специальная метка, незаметно внедряемая в изображение или другой сигнал с целью тем или иным образом контролировать его использование.

1 Стеганографические алгоритмы на основе слияния ЦВЗ и изображения-контейнера

ЦВЗ применяются в основном для защиты от копирования и несанкционированного использования. В отличие от обычных водяных знаков ЦВЗ могут быть не только видимыми, но и (как правило) невидимыми. Невидимые ЦВЗ анализируются специальным декодером, который выносит решение об их корректности.

ЦВЗ могут содержать некоторый аутентичный код, информацию о собственнике либо какую-нибудь управляющую информацию. Наиболее подходящими объектами защиты при помощи ЦВЗ являются неподвижные изображения, файлы аудио- и видеоданных.

На начальных этапах разработок предлагалось вкладывать информацию в незначимые биты для уменьшения визуальной заметности. Сейчас же ЦВЗ встраиваются в наиболее существенные области изображений, разрушение которых приведет к полной деградации самого изображения. Поэтому стеганографические алгоритмы учитывают свойства системы человеческого зрения аналогично алгоритмам сжатия изображений. В стегоалгоритмах зачастую используются те же преобразования, что и в современных алгоритмах сжатия (дискретное косинусное преобразование, вейвлет-преобразование).

Алгоритмы внедрения называются алгоритмами слияния, если вместо последовательности псевдослучайных чисел в изображение встраивается другое изображение (например, логотип фирмы). Размер внедряемого сообщения намного меньше размера исходного изображения. Перед встраиванием оно может быть зашифровано или преобразовано каким-нибудь иным образом.

Преимущества алгоритмов слияния:

1. можно допустить некоторое искажение скрытого сообщения, так как человек все равно сможет распознать его;
2. наличие внедренного логотипа является более убедительным доказательством прав собственности, чем наличие некоторого псевдослучайного числа.

2 Алгоритм Чайя (Chae)

В алгоритме предусматривается внедрение черно-белого изображения (ЦВЗ), размером до 25 % от размеров исходного изображения-контейнера.

Перед встраиванием выполняется одноуровневая декомпозиция как исходного изображения, так и ЦВЗ с применением фильтров Хаара. Вейвлет-коэффициенты исходного изображения обозначаются, как $f(m, n)$, а вейвлет-коэффициенты логотипа-ЦВЗ - $w(m, n)$. Модификации подвергаются все коэффициенты преобразования. Вначале коэффициенты каждого поддиапазона, как исходного изображения, так и ЦВЗ представляются 24 битами (из которых один бит отводится на знак). Размер ЦВЗ в 4 раза меньше исходного изображения, поэтому необходимо увеличить количество его коэффициентов.

После этого расширенный вчетверо блок коэффициентов ЦВЗ поэлементно складывается с 24-битной версией исходного изображения

$$f'(m, n) = \alpha f(m, n) + w(m, n) \quad (1)$$

Полученное значение отображается назад к исходной шкале на основе значений минимального и максимального коэффициента поддиапазона. После чего осуществляется обратное дискретное вейвлет-преобразование. Для извлечения ЦВЗ используется инверсная формула

$$w_i^* = \frac{f^*(m, n) - f(m, n)}{\alpha f(m, n)} \quad (2)$$

Достоинством данного алгоритма является то, что он позволяет скрыть довольно большой объем данных в исходном изображении: до четверти от размеров исходного изображения.

Заключение

Разработанная программа встраивает bmp, jpeg-изображения (рис.1) в изображения-контейнеры (рис.2), а также извлекает их описанным выше методом.



Рисунков 1 - Внедряемое изображение – ЦВЗ



Рисунок 2 - Изображение-контейнер
Программа разработана на языке C++ с использованием библиотеки Qt компании Nokia. Выбор данной среды обусловлен платформонезависимостью и удобством работы. Программа работает в операционных системах: Microsoft Windows (XP/Vista/7), Linux.



Рисунок 3 - Изображение со скрытым ЦВЗ

В результате получается скрытое изображение логотипа в изображении контейнера (рис.3), а при раскодировании – первоначальное изображение ЦВЗ.

Литература

1. Грибунин В.Г., Оков И.Н., Туринцев И.В. Цифровая стеганография. – М.: СОЛОН- Пресс, 2002.
2. Комбинированные методы защиты данных в САПР. - Волосатова Т.М., Денисов А.В., Чичварин Н.В. // «Информационные технологии», Приложение №5, 2012 – 32 с.
3. Защита проектной документации от несанкционированного доступа. - Волосатова Т.М., Денисов А.В., Чичварин Н.В. // 9 Международная конференция «Эффективные методы автоматизации подготовки и планирования производства», сборник трудов, с141-145. М.: 2012.
4. Метод сохранения данных с использованием искусственной дефокусировки. - Волосатова Т.М., Денисов А.В., Чичварин Н.В. //9 Международная конференция «Эффективные методы автоматизации подготовки и планирования производства», сборник трудов, с145-148. М.: 2012.
5. Анализ стеганографического алгоритма Чайя. – Булгакова В.А., Захаров А.В. // 3 Всероссийская научно-техническая конференция «Безопасные информационные технологии», сборник трудов, с.31-33. М.: 2012.
6. Chae J. J., Mukherjee D., Manjunath B. S. A robust embedded data from wavelet coefficients // Proceedings of SPIE, Electronic Imaging, Storage and Retrieval for Image and Video Database. 1998.

СТЕГАНОГРАФИЯ В ЦИФРОВЫХ ИЗОБРАЖЕНИЯХ. РЕАЛИЗАЦИЯ АЛГОРИТМА ПИТАСА

Чепурнов П.И.

Научный руководитель: к.т.н., доцент Волосатова Т.М.

МГТУ им. Н.Э.Баумана, кафедра РК6, Москва, Россия

STEGANOGRAPHY IN DIGITAL IMAGES. IMPLEMENTATION OF THE ALGORITHM PITASA

Chepurnov P.I.

Supervisor: Volosatova T.M.

BMSTU, Department of RK6, Moscow, Russia

Аннотация

В статье рассматриваются стеганографические алгоритмы, позволяющие внедрять текстовую информацию в цифровое изображение. Подробно исследован алгоритм Питаса, в примерах получены изображения, имеющие в себе скрытую информацию, и не отличающиеся от исходных для человеческого глаза. В заключении проанализированы искажения исходных изображений, используемых в качестве стегоконтейнеров.

Annotation

In the article, steganography algorithms to embed text information in a digital image. Studied in detail the algorithm of Pitasa, with examples obtained images with a hidden information, and did not differ from baseline to the human eye. In conclusion, analyzes the distortion of source images used as steshokonteynerov.

Введение

Стеганография - то наука о скрытой передаче [информации](#) путём сохранения в тайне самого факта передачи. В отличие от [криптографии](#), которая скрывает содержимое секретного сообщения, стеганография скрывает сам факт его существования. Как правило, сообщение будет выглядеть как что-либо иное, например, как изображение, статья, список покупок или письмо.

Из рамок цифровой стеганографии вышло наиболее востребованное легальное направление — встраивание цифровых водяных знаков (ЦВЗ) (watermarking), являющееся основой для [систем защиты авторских прав](#) и DRM (Digital rights management) систем. Методы этого направления настроены на встраивание скрытых маркеров, устойчивых к различным преобразованиям контейнера (атакам). [Цифровые водяные знаки \(ЦВЗ\)](#) используются для защиты от копирования, сохранения [авторских прав](#). Невидимые водяные знаки считываются специальным устройством, которое может подтвердить либо опровергнуть корректность. [ЦВЗ](#) могут содержать различные данные: авторские права, идентификационный номер, управляющую информацию. Наиболее удобными для защиты с помощью ЦВЗ являются неподвижные изображения, аудио и видео файлы.

В данной работе рассмотрим стеганографическую программу, в основе которой лежит алгоритм Питаса. Необходимо выяснить, как внедрять биты текстовой информации, а так же понять, будет ли конечное искажение заметно для человеческого глаза. Так же провести исследование, как будет искажаться изображение в зависимости от количества внедренной информации.

1 Теоретическая основа стеганографии метода

В качестве стегоконтейнеров используются цифровые изображения. Положительные стороны внедрения скрытой информации в изображение:

- относительно большой объемом цифрового представления изображений, который позволяет внедрять ЦВЗ большого объема либо повышать робастность внедрения;
- заранее известный размер контейнера;

- наличием в большинстве реальных изображений текстурных областей, имеющих шумовую структуру и хорошо подходящих для встраивания информации;
- слабой чувствительностью человеческого глаза к незначительным изменениям цветов изображения, его яркости, контрастности, содержанию в нем шума, искажениям вблизи контуров.

Из последнего пункта вытекает, что системы человеческого зрения (СЧЗ) обладают слабой чувствительностью к изменению яркости и контрастности. Рассмотрим этот диапазон восприятия:

Чувствительность к изменению яркости можно определить следующим образом. Пусть имеется начальная освещенность I , на которую настроился человеческий глаз. Начнем выделять пятно, изменение освещенности которого ΔI до тех пор, пока оно не будет обнаружено. На рис.1(б) показана зависимость минимального контраста $\Delta I / I$ от яркости I . Как видно из рисунка, для среднего диапазона изменения яркости, контраст (примерно), тогда как для малых и больших яркостей значение порога неразличимости возрастает. Было установлено, что $\Delta I \approx 0.01 - 0.03 I$ для средних значений яркости.

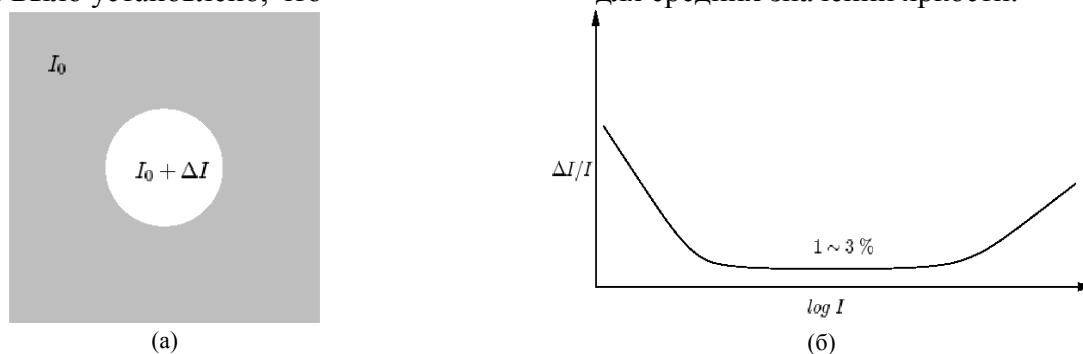


Рисунок 1 - Чувствительность к контрасту и порог неразличимости ΔI

Определения порога маскирования на основе известной яркостной чувствительности позволяет найти метрику искажения изображения, учитывающую свойства СЧЗ. Такого типа математические модели хорошо разработаны для случая квантования коэффициентов дискретного косинусного преобразования изображения, так как именно оно применяется в стандарте JPEG. Эффект маскирования в пространственной области может быть объяснен путем построения стохастических моделей изображения. При этом изображение представляется в виде марковского случайного поля, распределение вероятностей которого подчиняется, например, обобщенному гауссовскому закону.

Рассмотрим алгоритмы сжатия изображений, так как при встраивании ЦВЗ в основном используются те же подходы.

Стандарт сжатия JPEG является в настоящее время наиболее распространенным и своеобразным «benchmark» для алгоритмов ЦВЗ (то есть устойчивость системы ЦВЗ к сжатию JPEG проверяется обычно в первую очередь). В соответствии с этим стандартом изображение разбивается первоначально на блоки 8x8 элементов, к каждому из которых применяется дискретное косинусное преобразование (ДКП). Назначением ДКП является осуществление перераспределения энергии: значимые коэффициенты группируются в левом верхнем углу квадрата спектральных коэффициентов, так как соседние пиксели изображения коррелированы. Далее следуют равномерное табличное квантование коэффициентов, кодирование длин серий и кодирование Хаффмана.

2 Алгоритм Питаса

ЦВЗ представляет собой двумерный массив бит размером с изображение, причем число единиц в нем равно числу нулей. Существует несколько версий алгоритма, предложенного Питасом. Вначале предлагалось встраивать бит ЦВЗ в каждый пиксел изображения, но позже благоразумно было решено использовать для этой цели блоки

размером 2x2 или 3x3 пиксела, что делает алгоритм более робастным к сжатию или фильтрации. ЦВЗ складывается с изображением:

$$I(x, y) = I(x, y) + \alpha s(x, y)$$

В случае использования для внедрения блоков детектор ЦВЗ вычисляет среднее значение яркости этого блока. Отсюда появляется возможность неравномерного внедрения ЦВЗ в пиксели. Таким образом, можно получить ЦВЗ, оптимизированный по критерию робастности к процедуре сжатия алгоритмом JPEG. Для этого в блоке 8x8 элементов заранее вычисляют «емкость» каждого пиксела (с учетом ДКП и матрицы квантования JPEG). Затем ЦВЗ внедряют в соответствии с вычисленной емкостью. Эта оптимизация производится раз и навсегда, и найденная маска применяется для любого изображения. На рис.2 (а) и (б) показан ЦВЗ до и после оптимизации.

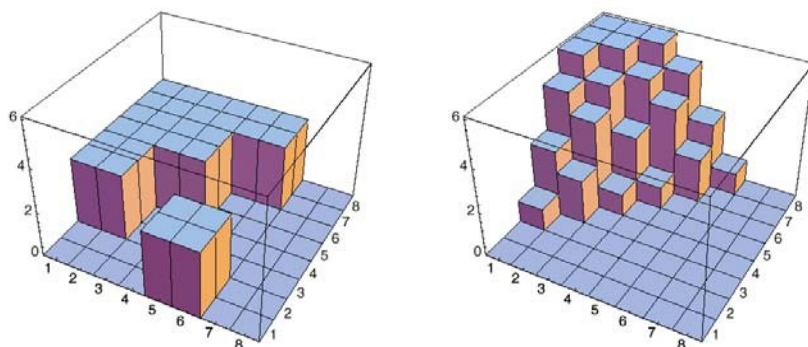


Рисунок 2 - Скрытие данных в пространственной области

Алгоритмы, описываемые в данном пункте, внедряют ЦВЗ в области исходного изображения. Их преимуществом является то, что для внедрения ЦВЗ нет необходимости выполнять вычислительно громоздкие линейные преобразования изображений. ЦВЗ внедряется за счет манипуляций яркостью $I(x, y) \in \{1, \dots, L\}$ или цветовыми составляющими $(r(x, y), b(x, y), g(x, y))$.

Пусть изображение имеет RGB-кодировку. Алгоритм Питаса позволяет определять области с большой робастностью и мы можем работать с битом наиболее подходящего цвета в определенной области. Рассмотрим алгоритм передачи одного бита секретной информации. Встраивание выполняется в канал красного цвета

Пусть s_i - встраиваемый бит, $I = \{R, G, B\}$ - контейнер, $p = (x, y)$ - псевдослучайная позиция, в которой выполняется вложение. Секретный бит встраивается в канал синего цвета путем модификации яркости по определенному закону. Хранение двоичной зашифрованной информации

$$b \mid p = \begin{cases} b \mid p + q \mid p, & \text{если } s_i = 0, \\ b \mid p - q \mid p, & \text{если } s_i = 1. \end{cases}$$

где q - константа, определяющая энергию встраиваемого сигнала. Ее величина зависит от предназначения схемы. Чем больше q , тем выше робастность вложения, но тем сильнее его заметность.

3 План реализации алгоритма

Пиксел изображения в 24-х битном .BMP формате занимает 3 Байта (24 бита) памяти (соответственно по 1 байту на каждый канал -Red, Green, Blue (RGB)) Один символ текста занимает 1 Байт (8 бит) дискового пространства. Если заменять Байт изображения Байтом текста, то мы получим абсолютно другой цвет пиксела, и, как следствие, сильное искажение изображения. Поэтому наиболее удобно заменять только 1 бит одного из каналов Пиксела на

1 бит Текста. Такая подмена будет незаметна для человеческого глаза. (Можно заменять последний бит каждого из каналов) Таким образом, один символ текста будет кодироваться в 8 пикселей изображения.

Весь процесс состоит из двух частей: шифрование и дешифровка. На этапе шифрования по линейным размерам изображения (в пикселях) мы сразу же оцениваем, какое количество текста мы можем поместить в данный стегоконтейнер. Искажений происходить не будет, т.к. если количество текста превышает максимальный размер контейнера – программа выдаст ошибку.

Заключение

Конечным продуктом является программа, осуществляющая ввод текстовой информации с последующим кодированием в цифровое изображение, а так же ее вывод. Эта программа реализована языке C++ с графическим интерфейсом, написанном в QT. Изменение цветовой палитры пикселя менее 0,3%. А человеческий глаз способен отследить изменение порядка (1-3)%. Следовательно, результатом работы программы будет изображение идентичное исходному изображению (используемого в качестве стегоконтейнера) для человеческого глаза, что позволит успешно хранить и передавать секретную информацию.

Литература

1. Жасмин Бланшет, Марк Саммерфилд - «QT4: Программирование GUI на C++» издательство «Кудиц пресс» Москва, 2007.
2. Грибунин В.Г., Оков И.Н., Туринцев И.В. - «Цифровая стеганография» издательство «Салон-пресс» 2009.
3. Коханович Г.Ф., Пузыренко А.Ю. «Компьютерная стеганография» издательство «МК-пресс» Киев, 2006.
4. Макс Шлее – «QT – профессиональное программирование на C++» издательство «БХВ - Петербург» 2005.
5. William K. Pratt – «Цифровая обработка изображений» издательство «Мир» Москва, 1982.
6. Научная статья Ioannis Pitas, Adrian G.Bors «Embedding parametric digital signatures in images» 1996.
7. Комбинированные методы защиты данных в САПР. - Волосатова Т.М., Денисов А.В., Чичварин Н.В. // «Информационные технологии», Приложение №5, 2012 – 32 с.: ил.
8. Защита проектной документации от несанкционированного доступа. - Волосатова Т.М., Денисов А.В., Чичварин Н.В. // 9 Международная конференция «Эффективные методы автоматизации подготовки и планирования производства», сборник трудов, с141-145. М.: 2012.
9. Метод сохранения данных с использованием искусственной дефокусировки. - Волосатова Т.М., Денисов А.В., Чичварин Н.В. //9 Международная конференция «Эффективные методы автоматизации подготовки и планирования производства», сборник трудов, с145-148. М.: 2012.

ИСПОЛЬЗОВАНИЕ ОНТОЛОГИЙ ДЛЯ УЛУЧШЕНИЯ ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА ПЕРСОНАЛЬНОГО КОМПЬЮТЕРА

Чернов М.М.

Научные руководители: к.т.н., доцент, Власов А.И., асс. Аверьянихин А.Е.

МГТУ им. Н.Э.Баумана, кафедра ИУ4, Москва, Россия

USING ONTOLOGIES TO IMPROVE THE USER PC INTERFACE

Chernov M.M.

Supervisors: Ph.D., associate professor, Vlasov A.I., ass. Averianixin A.E.

MSTU, Moscow, Russia

Аннотация

В статье рассматриваются вопросы разработки пользовательского интерфейса персонального компьютера на основе использования онтологий. Подробно исследовано создание онтологий предметной области на основе языка OWL. Представлена концепция извлечения информации через преобразования над узлами онтологии. В заключение представлены рекомендации по реализации разработанного интерфейса в виде системы с четырехслойной архитектурой.

Abstract

The article deals with the development of the user interface on the PC using ontologies. Detailed study of the creation of ontology-based language OWL. Finally, recommendations are designed to implement the interface in a system with four-layer architecture.

Введение

К настоящему времени в сети Интернет и на персональных компьютерах пользователей накоплен большой объем информации по разным областям знаний, хранящийся в файлах разных форматов или представленных на Интернет-ресурсах. Однако эти ресурсы достаточно хаотично распределены по отдельным сайтам или файловым системам персональных компьютеров, что значительно затрудняет поиск и использование содержащейся в них информации.

Меж тем как технические средства хранения информации достигли очень высокого уровня, методы организации информации на уровне файловой системы предлагают пользователю абстракцию файловых каталогов, формируя на устройствах долговременного хранения постоянную иерархическую структуру объектов для организации данных.

При этом пользователь вынужден использовать данную абстракцию для управления потоком информации исходящим от персонального компьютера или использовать систему поиска, которая, как правило, является частью любой современной операционной системы и путем постоянного индексирования содержимого файловой системы компьютера по ключевым словам позволяет быстро найти необходимую информацию. Но задача поиска осложняется тем, что некоторые форматы хранения информации (например, изображения, видео- и аудио-файлы), не позволяют осуществлять машинный поиск содержащейся в них информации, что делает актуальной задачу связывания файлов данных форматов с дополнительной информацией, по которой затем будет осуществляться поиск.

При создании системы поиска необходимо также учитывать специфику работы человеческого мышления, которое оперирует понятиями, множествами и связями между ними.

1 Использование онтологий в пользовательском интерфейсе персонального компьютера

Согласно [1-6] одним из подходов к созданию интеллектуальных систем является их разработка на основе онтологий. Это, с одной стороны, делает понятным для специалистов

интерфейс таких систем, с другой стороны, повышает доверие пользователей к ним. Онтологии также являются средством обеспечения повторного использования информации, хранимой в базах знаний интеллектуальных систем, а также методов, применяемых при решении задач.

Для описания модели знаний предметной области в виде онтологии необходимо выбрать язык описания онтологий. Согласно [2] в настоящее время широкое распространение получил язык OWL (web ontology language). OWL разработан консорциумом W3C в рамках инициативы Semantic Web. В 2004 году получил статус рекомендации [7–9]. OWL был создан в ответ на потребность в стандартизации способов представления знаний в Веб.

По выразительным способностям в рамках OWL выделяют три диалекта:

OWL Full обладает максимумом выразительных средств, но не гарантирует, что логический вывод в такой онтологии будет вычислимым. В частности в Full-диалекте классы могут одновременно выступать и в роли класса, и в роли экземпляра.

OWL DL гарантирует вычислительную полноту (логический вывод является вычислимым) и разрешимость (вычисления выполняются за конечное время). OWL DL содержит все языковые конструкции OWL, но их использование ограничено. DL отражает связь этого языка с дескриптивной логикой (разрешимой частью логики предикатов первого порядка).

OWL Lite обладает наименьшими выразительными способностями, но может быть промежуточным звеном при переходе от простых таксономий к онтологиям.

Разработчики приложений, использующие язык OWL, должны определить, какой из диалектов лучше подходит к их задачам.

2 Структуры данных при классификации файлов

Решая задачу классификации данных на своём персональном компьютере, пользователь создает иерархическую структуру каталогов файлов. При этом классификация получается противоречивой поскольку один и тот же файл не может находиться в 2-х каталогах одновременно, кроме того каталогу файлов часто дается название понятия к которому относятся содержащиеся в нем файлы (например «Микроэлектроника»), меж тем как тип файлов определяется названием родительского (например «Книги») или дочернего (например «Изображения») каталога. Такой подход вынуждает пользователя запоминать, по какому принципу был классифицирован тот или иной ресурс [6] (сначала по типу файлов, а затем – по понятиям, или наоборот). На рисунке 1 представлен пример классификации файлов с использованием абстракции каталогов.

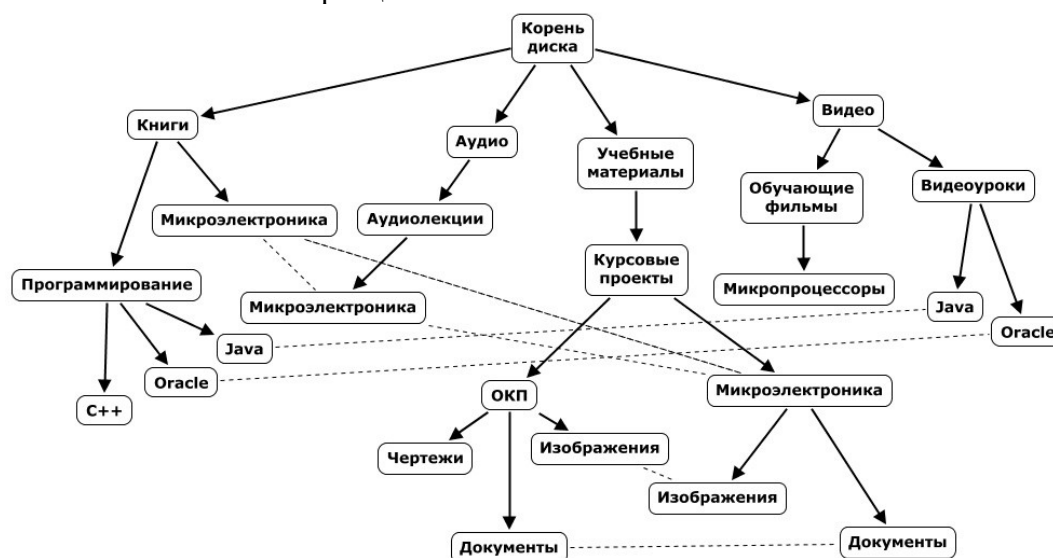


Рисунок 1 – Пример классификации файлов с использованием абстракции каталогов

Избавиться от неоднородности в классификациях можно разделив их на 2 части: дерево понятий и дерево множеств. Таким образом, в дереве множеств узлы оказываются упорядоченными отношением подмножества, а в дереве понятий - отношением подкласса. При этом для связи дерева понятий с деревом множеств задаются 2 отношения - представление и ассоциация. Определение таких отношений обусловлено тем, что любой ресурс (web-ссылка, множество файлов, файл, фрагмент изображения) может представлять какое-либо понятие («Человек», «Наука», «Строение»). При этом объем понятия всегда оказывается шире, чем объем представляющих его ресурсов. Например, понятие «Критика чистого разума» может быть представлено документом с текстом книги на русском языке, и в то же время представлено текстом на английском, либо множеством файлов с аудиозаписью текста книги или фрагментом изображения на котором представлен экземпляр книги. Отношение ассоциации может быть использовано, когда необходимо зафиксировать связь между 2-мя понятиями или ресурсами и в то же время нет необходимости в более точном определении характера связи.

На рисунке 2 представлено дерево множеств, упорядоченное отношением подмножества, дерево понятий, упорядоченное отношением подкласса и связи между узлами данных деревьев.

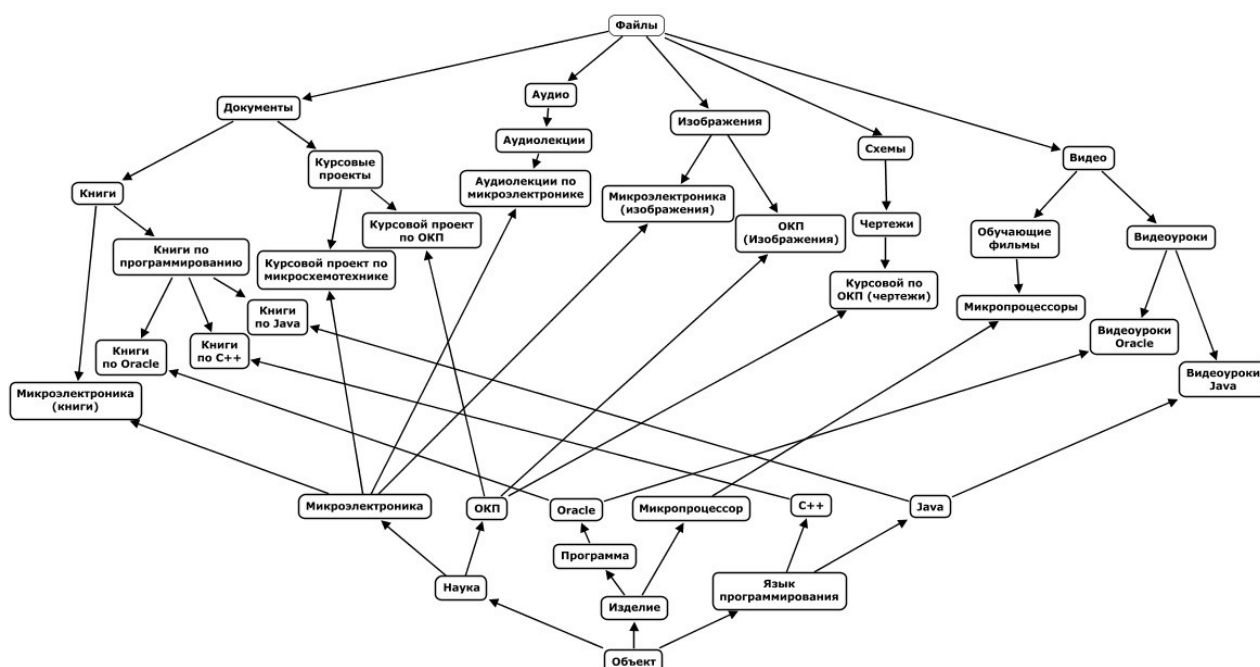


Рисунок 2 – Отношения между деревом ресурсов и деревом понятий

Определение таких структур данных позволит пользователю обращаться к файлам на своём персональном компьютере через дерево понятий, что для человека предпочтительнее использования абстракции каталогов [4].

3 Извлечение информации через преобразования узлов классификаций и операции над множествами

Поддержка языка запросов SPARQL даёт возможность пользователю делать сложные запросы к базе знаний. Однако написание запросов на SPARQL требует времени и определенного уровня подготовки пользователя. Наряду с этим большинство задач по выбору данных из системы можно легко решить с помощью определенных преобразований над узлами дерева понятий и дерева множеств и последующими операциями (объединение, пересечение, разность, симметрическая разность) над полученными множествами ресурсов.

На рисунке 3 представлен результат преобразования выделенного понятия «Композитор» к множеству представляющих ресурсов для отображения пользователю.

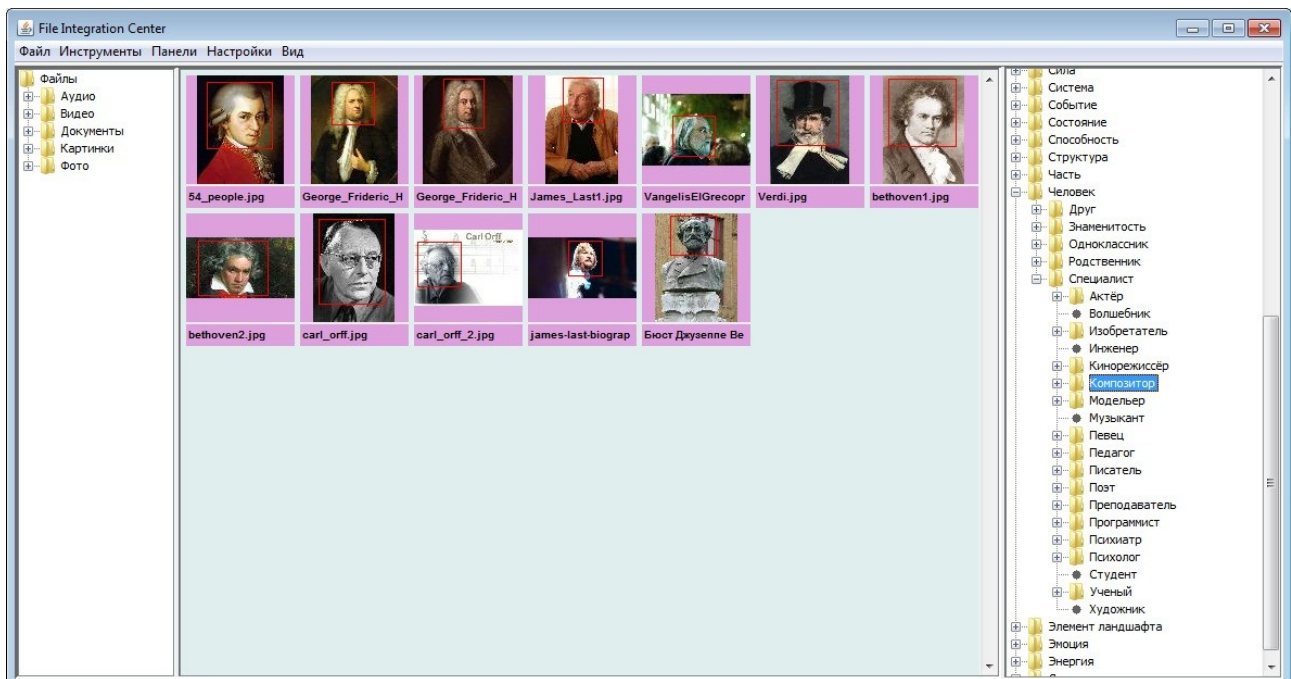


Рисунок 3 – Преобразование понятий к множеству представляющих ресурсов для отображения

Для удобной настройки механизма преобразований выбранных ресурсов для отображения может быть использован графический конструктор преобразований.

В данном конструкторе задается правило, согласно которому из базы знаний извлекаются ресурсы для отображения. Правила строятся на основании выделенных узлов и их отношений с другими ресурсами. На рисунке 4 представлено несколько правил созданных в графическом конструкторе трансформаций.

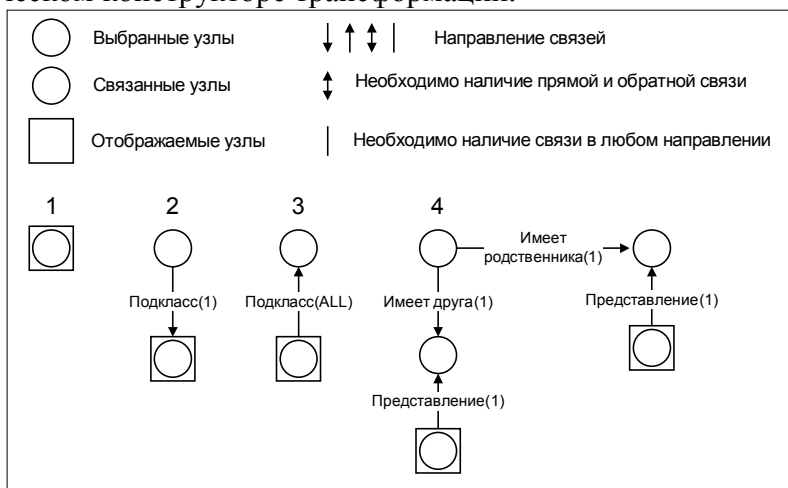


Рисунок 4 – Пример правил преобразования узлов созданных в конструкторе преобразований

В случае 1 (отображение без трансформаций) будут отображены выбранные узлы.

В случае 2 будут отображены непосредственные подклассы выбранных узлов (если такие найдутся).

В случае 3 будут отображены все родительские классы выбранного узла.

В случае 4 будут найдены все узлы, состоящие в отношении «Имеет друга» или «Имеет родственника» с выбранным узлом и отображены наборы ресурсов, представляющих найденные узлы.

К полученному на основе правил трансформации набору ресурсов могут быть применены операции (объединения, пересечения, разности, симметрической разности) с другими наборами ресурсов.

Операции манипулирования ресурсами как множеством осуществляется по уникальному идентификатору ресурса для всех видов ресурсов:

$$P_i^J \cap P_k^L = \begin{cases} P_i^J \cup P_k^L, J = L \\ \emptyset \end{cases} \quad (1)$$

Либо для ресурсов, являющихся частью других ресурсов (например, фрагментов изображений) при операциях пересечения и разности используется идентификатор родительского ресурса:

$$F_i^j \in P_j, \forall i$$

$$F_i^j \cap F_k^l = \begin{cases} F_i^j \cup F_k^l, j = l \\ \emptyset \end{cases} \quad (2)$$

В случае (2) операции над множествами позволяют быстро находить ресурсы, связанные не непосредственно, а через свои части с понятиями, между которыми требуется установить связь.

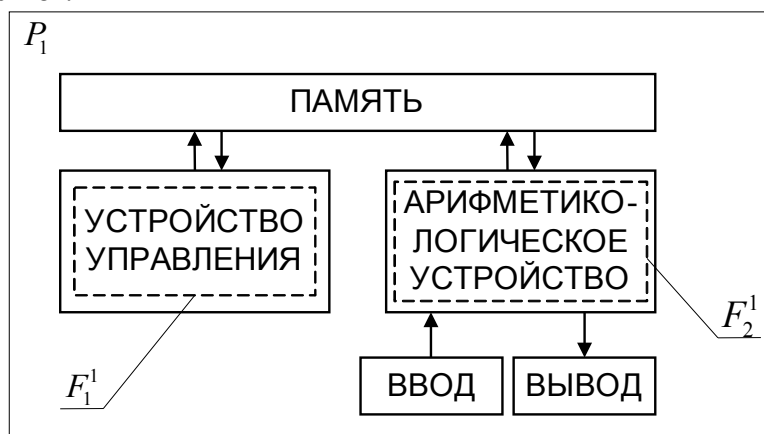


Рисунок 5 - Пример изображения (схемы) с фрагментами, представляющими понятия «Устройство управления» и «Арифметико-логическое устройство»

На рисунке 5 представлено изображение P_1 , на котором отмечены фрагменты F_2^1 и F_1^1 представляющие понятия «Устройство управления» и «Арифметико-логическое устройство» соответственно. Операция пересечения представляющих множеств понятий «Устройство управления» и «Арифметико-логическое устройство» в случае (1) даст пустое множество т.к. фрагменты изображения, на которых представлены понятия, имеют разные идентификаторы, а в случае (2) – объединение фрагментов т.к. они относятся к одному изображению. Таким образом, использование второго вида операций над ресурсами позволяет выявлять дополнительные связи между понятиями не заданные в явном виде.

4 Пример реализации модели понятийного интерфейса

При реализации разработанного интерфейса [3] необходимо учитывать, что для существующих операционных систем разработано множество программ для работы с разными типами файлов, поэтому разрабатываемый комплекс, должен быть создан как надстройка над существующей файловой системой, не ограничивая возможности пользователя по использованию установленной операционной системы. При этом инструменты нового интерфейса должны быть максимально интегрированы в интерфейс установленной операционной системы.

На рисунке 6 представлена архитектура системы, реализующий разработанный интерфейс.

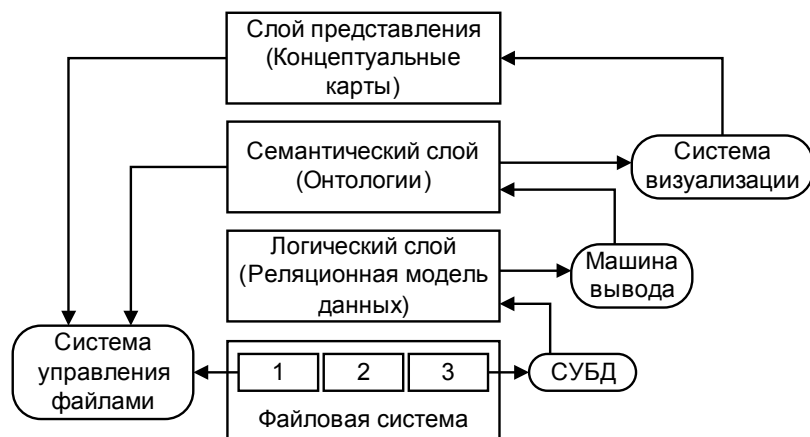


Рисунок 6 – Послойная структура разрабатываемого программного комплекса

Разделение архитектуры приложения на слои дает возможность использовать готовые решения для создания каждого слоя [5]. Так на физическом уровне используются средства файловой системы компьютера, предоставляемые установленной операционной системой, а для создания логического слоя может быть использована любая реализация СУБД, поддерживающая язык запросов SQL3.

Заключение

Результатом работы явилась разработка пользовательского интерфейса персонального компьютера, основанного на использовании онтологий. Был предложен подход к виртуализации файловой системы компьютера на основе модели RDF.

Использование языка описания онтологий OWL и модели RDF для описания знаний и виртуализации файловой системы компьютера позволило создать единый интерфейс для управления накопленными знаниями и файлами. Преимуществом языка OWL при описании метаданных системы явилось наличие диалектов, различающихся выразительностью и вычислительной сложностью, что дало возможность использовать более простой диалект (OWL Lite) в отношении файловой системы компьютера и более сложные диалекты (OWL DL, OWL Full) при описании онтологии предметной области.

В работе была представлена структура программного комплекса, представляющего из себя систему с 4-х слойной архитектурой, реализующего разработанный интерфейс. Преимуществом разделения архитектуры системы на слои явилась возможность использовать готовые библиотечные решения при создании каждого слоя.

Литература

1. Артемьева И.Л., Рештаненко Н.В. Интеллектуальная система, основанная на многоуровневой онтологии химии // Программные продукты и системы. – 2008. – № 1.-Режим доступа: <http://www.swsys.ru/index.php?page=article&id=113>.
2. Трофимов И.В. Эволюция выразительных способностей языка OWL // Программные системы: теория и приложения : электрон. научн. журн. 2011. № 4(8), с. 85–94. – Режим доступа: http://psta.psiras.ru/read/psta2011_4_85-94.pdf.
3. Аверьянихин Артур Евгеньевич, Верстов Владимир Александрович, Власов Андрей Игоревич, Фраерман Аркадий Игоревич Свидетельство о государственной регистрации программ для ЭВМ №2012610397 от 10 января 2012 года "Экспертная система по технологиям микро- и наносистем".
4. Власов А.И. Гексагональная понятийная модель визуального представления сложных производственных систем// Вестник МГТУ им. Н.Э. Баумана, сер. Приборостроение. 2012. С.157-169.
5. Власов А.И. Применение методов визуального моделирования для формализации конструкторско-технологической информации// Информатизация образования - 2012: материалы Международной научно-практической конференции, г. Орёл. - Орёл: ФГБОУ "ОГУ", 2012. - С.70-78.
6. Резчикова Е.В., Власов А.И. Перспективы применения концепт-карт для построения базы знаний ТРИЗ // Сборник трудов конференции "ТРИЗ. Практика применения методических инструментов". - Москва. 29 октября 2011. С.140-145.
7. OWL Web Ontology Language.Overview. W3C Recommendation : W3C, 27 October 2009. – Режим доступа: <http://www.w3.org/TR/owl-features/>.
8. OWL Web Ontology Language Guide. W3C Recommendation : W3C, 10 February 2004. – Режим доступа: <http://www.w3.org/TR/owl-guide/>.
9. OWL Web Ontology Language. Reference. W3C Recommendation : W3C, 10 February 2004. – Режим доступа: <http://www.w3.org/TR/owl-ref/>.

РАЗРАБОТКА ПРИБОРА ДЛЯ МНОГОЭЛЕКТРОДНОГО МОНИТОРИНГА СОСТАВА СРЕД КУЛЬТИВИРОВАНИЯ ЭМБРИОНОВ

Чупахин А.П.

Научные руководители: д.б.н., профессор, Семёнова М.Л.

МГУ им. М.В. Ломоносова, биологический факультет, кафедра эмбриологии,
Москва, Россия

д.т.н, профессор, РФ, Иванов В.Э.

Уральский федеральный университет имени первого Президента России Б.Н.Ельцина,
кафедра Технологии и Средств Связи.

DEVELOPMENT OF THE DEVICE OF THE MULTIELECTRODE MONITORING OF MEDIUM COMPOSITION OF EMBRIOS CULTIVATION

Chupakhin A.P.

Supervisors: Doctor of Biological Sciences., professor, Semenova M.I.

M.V. Lomonosov Moscow State University, faculty of biology, department of embryology,
Moscow, Russia

Doctor of Technical Sciences, professor, Ivanov V.E.

Ural Federal University, department of Technology and Communications,
Yekaterinburg, Russia.

Аннотация

В статье рассматривается один из вариантов осуществления мониторинга химических и биологических сред посредством многоэлектродной системы, её аппаратная и программная части, а так же финансовый аспект. Подробно исследовано осуществление вышеупомянутой системы для мониторинга состава сред культивирования эмбрионов. Кратко представлены результаты испытания разработанной системы в лабораторных условиях. В заключении представлены рекомендации по дальнейшему развитию разработки.

Abstract

The report presents the method of the monitoring of chemical and biological environment by means of multi-electrodes system, its hardware and software as well as the financial aspect. The using of the method to control the composition of culture medium of the embryos was studied in detail. The results of testing of developed system in laboratory are summarized. The recommendations for the further development of the project are provided in the conclusion.

Введение

Основной проблемой при изучении живых биологических объектов является их сохранность в процессе получения информации. К тому же, малый размер биологических объектов затрудняет использование традиционных методов регистрации их состояния.

Использование системы микроэлектродов позволит решить возникающие перед исследователем проблемы. Размеры микроэлектродов сопоставимы с размерами клеток, поэтому их можно помещать в межклеточное пространство и следить за изменениями, происходящими непосредственно в ткани, а не в окружающей среде вблизи ткани. Микроэлектроды не токсичны, поэтому их использование не влияет на процессы, происходящие в биологических объектах. Система микроэлектродов позволяет одновременно производить измерения различных параметров: концентрации кислорода, углекислого газа, продуктов азотистого обмена, различных ионов. Такое количество получаемых данных позволит создать адекватную систему мониторинга состояния исследуемого образца. В результате создания такой системы можно значительно повысить процент выживаемости эмбрионов и тканей.

Использование системы микроэлектродов способно облегчить и модернизировать рабочий процесс в ряде медицинских центров, больниц, а также научно-исследовательских институтов и биологических лабораторий.

Однако, на сегодняшний день столь перспективная тема практически не затрагивается. Публикации на данную тему единичны [1-2].

Значимой проблемой является выбор оборудования, благодаря которому возможно снятие и обработка сигналов с электродов. Анализ современного рынка показал, что большинство используемой для данной цели аппаратуры является одноканальными универсальными рН-метрами/ионометрами и позволяет снимать показания только с определённых типов электродов. В то же время, анализ литературы показывает возможность создания электродов для измерения концентрации таких веществ, как 17- β -эстрадиол [3], что является очень важным для анализа сред культивирования эмбрионов. Более серьёзное многоканальное оборудование, позволяющее использовать любые электроды, имеет высокую стоимость.

В данной публикации приведены результаты разработки малобюджетного универсального многоканального оборудования и программного обеспечения для работы с системами электродов.

1 Конструкция прибора

Измерительная система (ИС): Совокупность измерительных, связующих, вычислительных компонентов, образующих измерительные каналы, и вспомогательных устройств (компонентов измерительной системы), функционирующих как единое целое, предназначенная для:

- получения информации о состоянии объекта с помощью измерительных преобразований в общем случае множества изменяющихся во времени и распределённых в пространстве величин, характеризующих это состояние;
- машинной обработки результатов измерений;
- регистрации и индикации результатов измерений и результатов их машинной обработки;
- преобразования этих данных в выходные сигналы системы в разных целях.[4]

На основе этого определения построена следующая схема:

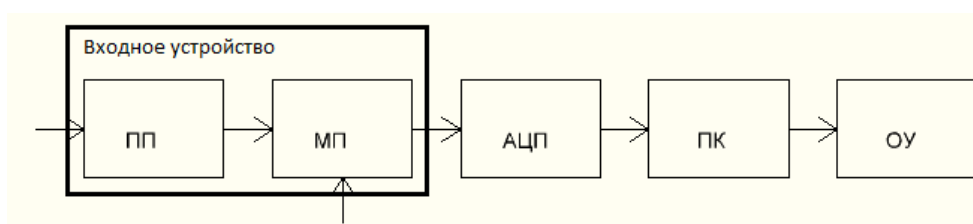


Рисунок 1 - Структурная схема цифрового измерительного прибора

Реализация каждого из блоков прибора:

1) Первичный измерительный преобразователь.

В качестве первичного измерительного преобразователя используется соответствующий электрод. Например, лабораторный комбинированный рН электрод К-10601/7. Такой электрод обладает электрическим сопротивлением от 10 до 80 Мом. Важно отметить, что в зависимости от величины рН такой электрод может иметь как положительный, так и отрицательный потенциал относительно встроенного электрода сравнения.

2) Масштабный преобразователь.

В связи с достаточно малыми входными токами (10^{-10} А) использован высокоомный операционный усилитель, включенный в качестве повторителя. Инвертирован входной

сигнал, а также реализовано смещение на величину равную 1,5 В. Это значение обусловлено напряжением питания используемого АЦП, а также конструкцией электрода.

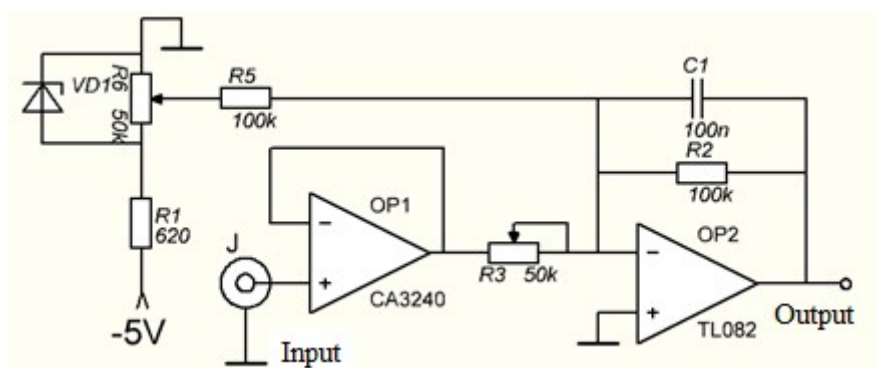


Рисунок 2 - Схема электрическая принципиальная МП

Разработанная схема усилительного каскада позволяет производить регулировку коэффициента усиления и подстраивать схему под параметры используемых электродов. Кроме того, многократное дублирование такой схемы дает возможность увеличения числа измерительных каналов без внесения коррекции в прочие структурные элементы устройства. Простота конструкции позволяет обеспечить невысокую стоимость.

3) АЦП.

Для измерения потенциала на выходе усилительной схемы, используется 12 – разрядный 16 – каналный АЦП микроконтроллера STM32F100R8. Напряжение питания АЦП - 3,3В.

Передача данных на ПК производится по протоколу RS - 232, при помощи универсального асинхронного передатчика, который также входит в состав STM32F100R8.

Для возможности использования с ПК, не имеющими последовательного порта, используется микросхема – конвертер FT232RL.

4) ПУ и ОУ.

Роль ПУ и ОУ выполняет персональный компьютер. Используется специально разработанная программа. В ее задачи входит обработка пришедших данных для отображения их в удобной для пользователя форме. Программа позволяет в реальном времени отслеживать колебания рН. Все полученные данные сохраняются в текстовый файл с указанием даты измерения. Для управления работой программы используется MDI интерфейс. Работа велась в среде разработки Microsoft Visual Studio 2010 [5].

3 Калибровка и результаты измерений

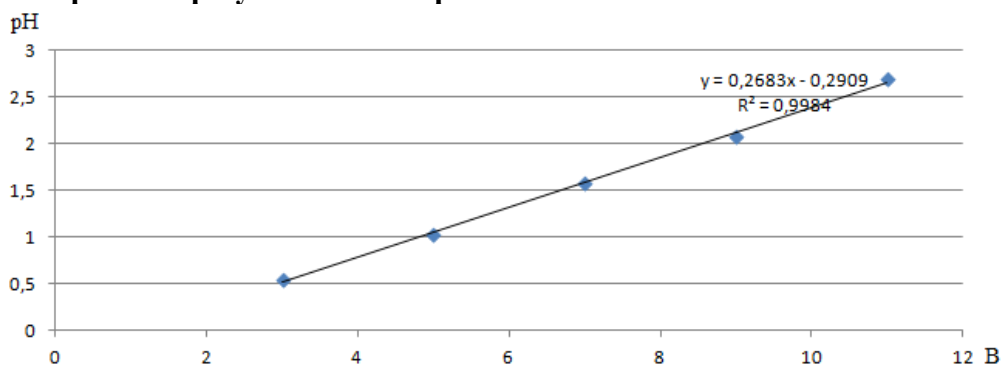


График калибровки для электрода К-10601/7

Таблица 1 - Данные по калибровке для электрода К-10601/7

№ п.п.	Фиксанал, состав	рН	Измерения	
			Полученные значения, В	Среднее значение, В
1	82,0мл 0,1М лимонной кислоты; 18,0мл 0,1М тринатриевой соли лимонной кислоты	3,0	0,542	0,545
			0,544	
			0,546	
			0,548	
			0,547	
2	35,0мл 0,1М лимонной кислоты; 65,0мл 0,1М тринатриевой соли лимонной кислоты	5,0	1,023	1,023
			1,023	
			1,022	
			1,022	
			1,022	
3	50,0мл 0,1М KH_2PO_4 ; 29,1мл 0,1М NaOH	7,0	1,592	1,586
			1,589	
			1,585	
			1,582	
			1,581	
4	25мл 0,2М глицина; 4,4мл 0,2М NaOH	9,0	2,090	2,085
			2,087	
			2,085	
			2,082	
			2,080	
5	50,0мл 0,05М Na_2HPO_4 ; 4,1мл 0,1М NaOH	11,0	2,703	2,697
			2,699	
			2,696	
			2,694	
			2,692	

Заключение

Разработано устройство мониторинга химических и биологических сред. Проведена апробация устройства с использованием комбинированного рН электрода. Полностью реализованы следующие требования: многоканальность, отображение результатов в реальном времени, невысокая стоимость устройства. Следующим этапом работы будет испытание разработки с использованием специализированных электродов, увеличение числа доступных пользователю функций путем доработки программного обеспечения, увеличение числа измерительных каналов.

Литература

1. Date Y, Takano S, Shiku H, Ino K, Ito-Sasaki T, Yokoo M, Abe H, Matsue T. Monitoring oxygen consumption of single mouse embryos using an integrated electrochemical microdevice. //Biosens Bioelectron. 2011. № 30(1) P. 100-106.
2. Trimarchi J.R., Liu L, Porterfield D.M., Smith P.J., Keefe D.L. A non-invasive method for measuring preimplantation embryo physiology // Zygote. 2000. №8(1). P. 15-24.
3. Kim Y.S., Jung H.S., Matsuura T, Lee H.Y., Kawai T., Gu M.B. Electrochemical detection of 17beta-estradiol using DNA aptamer immobilized gold electrode chip. // Biosens Bioelectron. 2007. № 22(11) P. 2525-2531.
4. ГОСТ Р 8.596 – 2002. Метрологическое обеспечение измерительных систем. Основные положения. Введ. 2002-30-09.
5. Майо Дж. Microsoft Visual Studio 2010. Самоучитель. - СПб.:БХВ-Петербург, 2010. - 450с.

АЛГОРИТМ ДВИЖЕНИЯ ДОДЕКАПОДА В ПРЯМОЛИНЕЙНОЙ ТРУБЕ ПЕРЕМЕННОГО СЕЧЕНИЯ

Данг Суан Хиен

Научный руководитель: д.ф.-м.н., профессор, Карпенко А.П.

МГТУ им. Н.Э. Баумана, кафедра РК6, Москва, Россия

MOVING ALGORITHM OF DODEKAPOD IN STRAIGHT TUBE WITH VARIABLE SECTION

Dang Xuan Hiep

Supervisor: Dr., Prof., Karpenko A.P.

BMSTU, Moscow, Russia

Аннотация

В статье исследуются алгоритмы движения додекапода в прямолинейной трубе. Рассмотрены цилиндрические трубы постоянного и переменного сечений. Исследовано необходимое и достаточное условия для преодоления додекаподом участка трубы переменного сечения. Выполнено моделирование движения додекапода в средах *Solidworks* и *MatLab*. В заключении представлены рекомендации по использованию алгоритма и перспективы применения додекапода.

Annotation

The article studies the movement algorithm of dodekapod in a straight tube. The tube is considered a cylindrical tube of constant and variable cross-sections. Studied the necessary and sufficient conditions for dodekapod passing through the tube. The algorithm is modeled in *Solidworks* and *MATLAB*. In conclusion, provided recommendations for the use of the algorithm and application prospects of dodekapod.

Введение

Додекапод представляет собой односекционный параллельный манипулятор, обладающий некоторыми преимуществами по сравнению с последовательными манипуляторами: высокая жесткость, точность позиционирования и большое рабочее пространство [1]. Параллельная структура позволяет додекаподу гибко менять свою форму, чтобы увеличивать рабочее пространство, сохраняя при этом точность позиционирования. Эта возможность используется для применения додекапода в разных отраслях промышленности.

В работе [2] представлена идея и рассмотрены возможности использования додекапода как мобильного пространственного робот-манипулятора, перемещающегося по сложной поверхности или в трубе переменного профиля.

В статье [3] выполнено исследование кинематики додекапода. Результаты исследования позволяют определить основные параметры додекапода: расстояние между платформой и основанием, положения и ориентация платформы.

В данной работе впервые выполнено исследование додекапода при движении в цилиндрической трубе постоянного и переменного сечений.

1 Основные параметры додекапода

Додекапод состоит из шести шарнирных узлов (A, B, C, D, E, F) и 12 штанг, соединяющих этих узлов. Узлы (A, B, C) и (D, E, F) образуют две параллельные грани (рисунок 1). Эти тройки узлов (A, B, C) и (D, E, F) играют роль упоров при движении додекапода. Когда одна грань фиксирована, другая грань движется за счет одновременного изменения длин штанг AD, AE, BD, BF, CE, CF (рисунок 2). Предлагается что в процессе движения додекапод отсутствует проскальзывание шарниров.

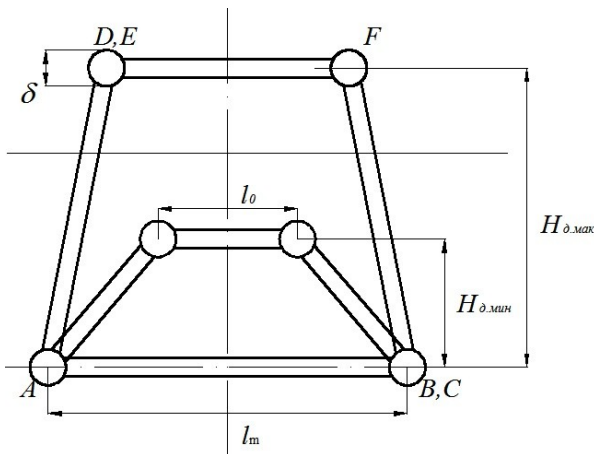


Рисунок 1 - Схема додекапода и основные обозначения

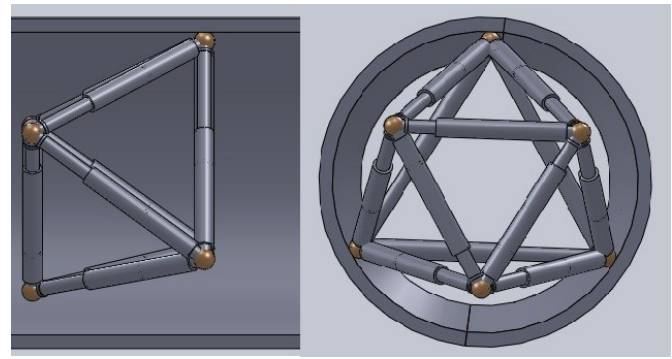


Рисунок 2 - Додекапод в цилиндрической трубе постоянного сечения:
а) передний вид; б) боковой вид

Введем следующие основные обозначения:

l_0, l_m - минимальная и максимальная длины штанги додекапода;

δ - диаметр шарнира додекапода;

l_s - длины штанг на гранях ABC, DEF , когда соответствующие соприкасаются с внутренней поверхностью трубы;

$H_{д.мин.}$ - минимальное расстояние между гранями ABC и DEF (когда длины боковых штанг достигают минимальной длины);

$H_{д.макс.}$ - максимальное расстояние между гранями ABC и DEF (когда длины боковых штанг достигают максимальной длины);

h_s - шаг движения додекапода, $l_0 < h_s < l_m$.

2 Алгоритм движения додекапода в трубе постоянного сечения

Пусть цилиндрическая труба имеет внутренний диаметр D и длину L . Для того, чтобы в процессе движения додекапод прочно фиксировался в трубе, грани ABC и DEF должны быть равносторонними треугольниками, перпендикулярными к оси симметрии трубы (рисунок 3). Отсюда следует, что условия перемещения додекапода в указанной трубе имеют вид: величина l_s находится в диапазоне возможных длин штанг, то есть $l_0 < l_s < l_m$; из рисунка 3 имеется $l_s = \frac{\sqrt{3}}{2} D$. Тогда получает условие: $\frac{2}{\sqrt{3}} l_0 \leq D \leq \frac{2}{\sqrt{3}} l_m$ (1)

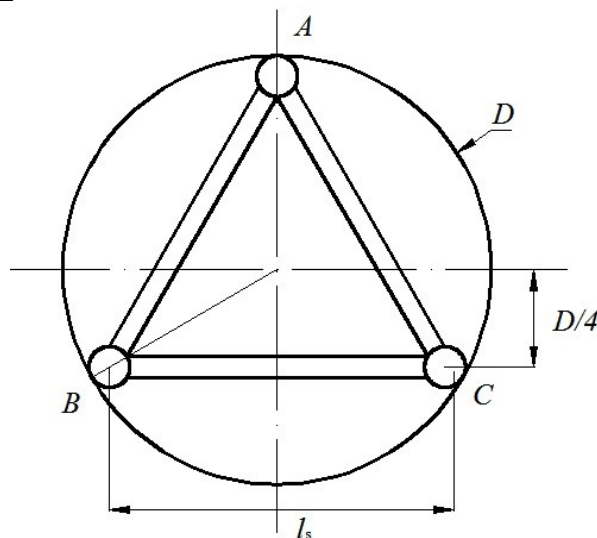


Рисунок 3 - Определение длины штанг додекапода
Алгоритм движения додекапода имеет следующий вид (рисунок 4).

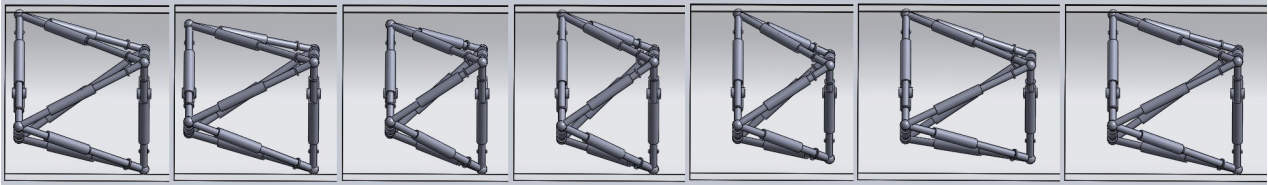


Рисунок 4 - Движения додекапода в трубе постоянного сечения

1) Задаем начальные положения длин штанг и начальные положения шарниров:

$$l_{AB} = l_{BC} = l_{CA} = l_{DE} = l_{EF} = l_{DF} = l_s = \frac{\sqrt{3}}{2} D; \quad l_{AF} = l_{BF} = l_{BD} = l_{CD} = l_{CE} = l_{AE} = l_m.$$

2) Уменьшаем длин штанг AB, BC, CA с l_s до l_0 .

3) Уменьшаем длин штанг AD, AE, BD, BF, CE, CF с l_m до l_0 .

4) Повышаем длин штанг AB, BC, CA с l_0 до l_s .

5) Уменьшаем длин штанг DE, EF, DF с l_s до l_0 .

6) Повышаем длин штанг AD, AE, BD, BF, CE, CF с l_0 до l_m .

7) Повышаем длин штанг DE, EF, DF с l_0 до l_s . После этого шага додекапод

двигается вперед на шаг $h_{\text{со макс}} = \frac{H}{\delta \cdot \text{мин}}$.

8) Повторяем шаги 2-7 до достижения додекаподом конца трубы.

3 Алгоритм движения додекапода в трубе с переменным профилем

Рассматриваемая труба имеет три части: две цилиндрические части с диаметрами соответственно D (длина L_1) и d (длина L_2); переходной конической участком длиной Δ (рисунок 5). Сложность задачи заключается в том, что в процессе преодоления конического участка додекапод может столкнуться с трубой. Чтобы исключить столкновение, нужно обеспечить выполнение ряда ограничений на параметры додекапода и величины его шагов.

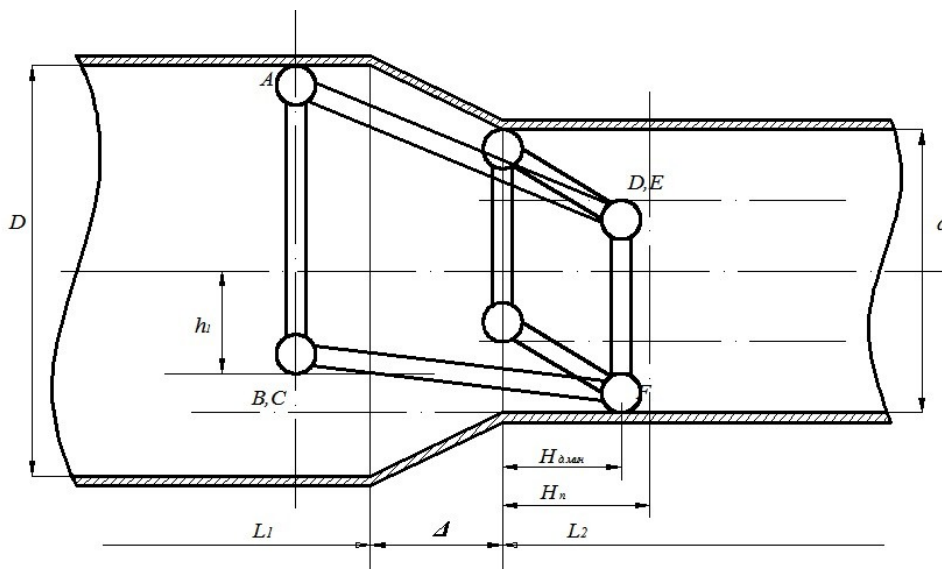


Рисунок 5 - Состояние додекапода на переходе трубы

Введем следующие обозначения:

$$l_{AB} = l_{DE} = \frac{\sqrt{3}(D - \delta)}{2} \text{ - длина штанг граней } ABC \text{ и } DEF;$$

$$l_{AF} = l_m \text{ - длина боковых штанг.}$$

Алгоритм движения додекапода представляется таким образом (рисунок 6).

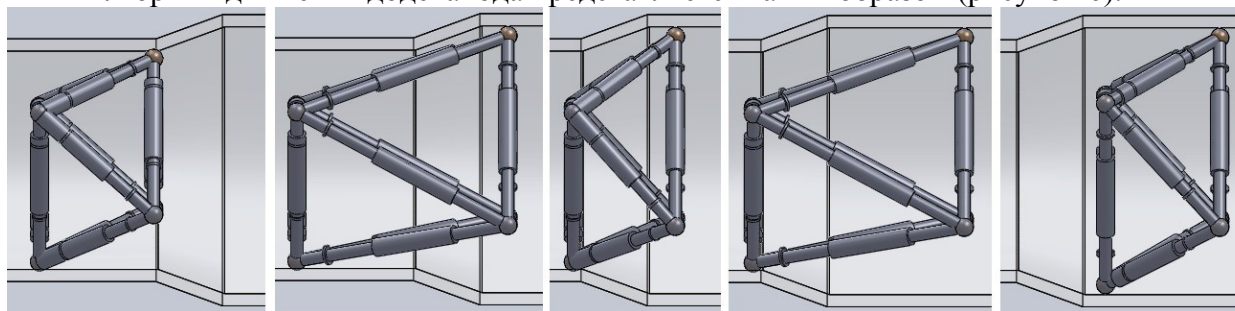


Рисунок 6 - Движение додекапода в трубе переменного сечения

- 1) Движение с шагом $h_{s\phi} = \frac{H}{\phi} \cdot \frac{H}{\phi}$ до достижения начала конического участка трубы (за $N = [L_1 / h_s]$ шагов).
- 2) Движение с шагом $h_{s1} = L_1 - Nh_s$ до достижения гранью DEF левого края конического перехода.
- 3) Движение через конический переход с шагом $h_{s\phi} = \Delta + H$.
- 4) Продолжение движения до конца трубы с шагом $h_{s\phi} = \frac{H}{\phi} \cdot \frac{H}{\phi}$.

Заключение

В работе рассматриваются условия и алгоритмы движения додекапода в прямолинейной трубе постоянного и переменного сечений. В развитие работы планируется изучение алгоритмов движения додекапода в непрямолинейных трубах, трубах эллиптического и других профилей, в трубах с отводами и т.д.

Литература

1. Merlet J.P. Parallel Robots // Kluwer Academic Publishers, 2000, p. 31-65.
2. Саяпин С.Н., Синев А. В. Адаптивный мобильный пространственный робот-манипулятор и способ организации движений и контроля физико-механических свойств и геометрической формы контактируемой поверхности и траектории перемещения с его помощью // Патент на изобретение № 2424893, 2011г.
3. Данг С.Х. Кинематика односекционного параллельного манипулятора типа «додекапод» // Наука и образование: электронное научно-техническое издание, www.technomag.edu.ru, 02.2013г.

МОДУЛЬ НАНЕСЕНИЯ ЦИФРОВОГО ВОДЯНОГО ЗНАКА НА ОСНОВЕ ТОНКОГО КЛИЕНТА ДЛЯ СОЗДАНИЯ СХЕМ РАЗЛИЧНОЙ ФИЗИЧЕСКОЙ ПРИРОДЫ

Друк А.И.

Научный руководитель: доцент, кандидат наук, Волосатова Т.М.

МГТУ им. Н.Э.Баумана, кафедра РК6, Москва, Россия

DIGITAL WATERMARKING MODULE BASED ON THIN-CLIENT TO CREATE A CIRCUIT OF DIFFERENT PHYSICAL NATURE

Druk A.I.

Supervisor: Associate Professor, Ph.D., Volosatova T.M.

MSTU, Moscow, Russia

Аннотация

В статье рассматриваются принципы модульного построения тонкого клиента для создания и редактирования схем различной физической природы. Подробно рассмотрен метод нанесения цифрового водяного знака. Кратко представлено сравнение возможностей современных технологий создания тонкого клиента. В заключении представлены рекомендации по выбору технологий.

Abstract

The article describes module structure principles of the thin client to create and edit circuits of different physical nature. Detailed digital watermarking technology described. Summarizes the comparison of the capabilities of modern technology to create a thin client. In conclusion, presented recommendations for the choice of technology.

Введение

Целью работы является создание приложения для нанесения цифрового водяного знака (ЦВЗ) на основе редактора схем различной физической природы. Модульность приложения WebCAD позволяет реализовать данный метод без существенной переработки исходного приложения. Тем самым, расширяется исходный функционал приложения, появляются более совершенные методы защиты чертежей.

Построение схем является неотъемлемой частью инженерных задач. Автоматизация данного процесса позволяет ускорить их решение, снизить затраты и уменьшить количество ошибок. Разработка приложения велась с использованием систем командной разработки, таких как система контроля версий.

Инновационным подходом в создании приложений подобной направленности является клиент-серверная архитектура. Большинство существующих аналогов требуют наличия дополнительного программного обеспечения для корректной работы.

Цифровой водяной знак

Любой чертеж является интеллектуальной собственностью автора. В подавляющей большинстве случаев автор захочет защитить свою собственность от нежелательного распространения. Одним из самых распространенных способов защиты визуальной информации являются водяные знаки. Их наносят на ценные бумаги, документы, денежные купюры и т.д. Нанеся водяной знак на чертеж, можно гарантировать защиту авторских прав. Цифровой водяной знак (ЦВЗ) - это специальная метка, встраиваемая в цифровой контент (называемый контейнером) с целью защиты авторских прав и подтверждения целостности самого документа. ЦВЗ можно встраивать в электронные документы любого типа.

Следующий код реализует технологию нанесения текстового цифрового водяного знака на чертеж.

```

public void Watermark()
{
    WriteableBitmap input = bPic.ToBitmap();

    TextBlock txt = new TextBlock();
    txt.Text = "* МПТУ им. Н.Э. Баумана *\\nWebCAD v1.2";
    txt.FontSize = 25;
    txt.TextAlignment = TextAlignment.Center;
    txt.FontWeight = FontWeights.Bold;
    txt.Foreground = new SolidColorBrush(Colors.Blue);
    txt.Effect = new DropShadowEffect();
    txt.Opacity = 0.3;

    WriteableBitmap watermark = new WriteableBitmap(400, 100);
    watermark.Render(txt, null);

    watermark.Invalidate();
    var width = watermark.PixelWidth;
    var height = watermark.PixelHeight;

    var result = input.Clone();

    var position = new Rect(input.PixelWidth - width,
        input.PixelHeight - height, width, height);
    result.Blit(position, watermark, new Rect(0, 0, width, height));
    bPic = result.ToImage();
    SaveFile(stream, SaveFormat.JPEG);
}

```

Невидимые ЦВЗ внедряются в цифровые данные, но не могут быть восприняты как таковые. Важнейшее применение цифровые водяные знаки нашли в системах защиты от копирования, которые стремятся предотвратить или удержать от несанкционированного копирования цифровых данных. Стеганография применяет ЦВЗ, когда стороны обмениваются секретными сообщениями, внедрёнными в цифровой сигнал. Используется как средство защиты документов с фотографиями — паспортов, водительских удостоверений, кредитных карт с фотографиями. Комментарии к цифровым фотографиям с описательной информацией — ещё один пример невидимых ЦВЗ. Хотя некоторые форматы цифровых данных могут также нести в себе дополнительную информацию, называемую метаданными, ЦВЗ отличаются тем, что информация «защита» прямо в сигнал. Объекты мультимедиа в этом случае будут представлять собой контейнеры (носители) данных.

Основное преимущество состоит в наличии условной зависимости между событием подмены объекта идентификации и наличии элемента защиты — скрытого водяного знака. Подмена объекта идентификации приведет к выводу о подделке всего документа. Цифровые водяные знаки получили своё название от старого понятия водяных знаков на бумаге (деньгах, документах).

Так называемый, жизненный цикл ЦВЗ может быть описан следующим образом. Сначала в сигнал-источник S в доверенной среде внедряются водяные знаки при помощи функции E . В результате получается сигнал S_E . Следующий этап — распространение S_E через сеть или любым другим способом. Во время распространения на сигнал может быть совершена атака. У получившегося сигнала S_{EA} водяные знаки могут быть потенциально уничтожены или изменены. На следующем этапе функция обнаружения D пытается обнаружить водяные знаки w , а функция R вытащить из сигнала внедрённое сообщение. Этот процесс потенциально может совершать злоумышленник.

Одной из особенностей приложения является его модульная архитектура. Использование данной архитектуры позволяет добавлять новую функциональность без существенной переработки приложения. На базе приложения WebCAD разработано несколько модулей с реализацией стеганографических методов и методов нанесения ЦВЗ.

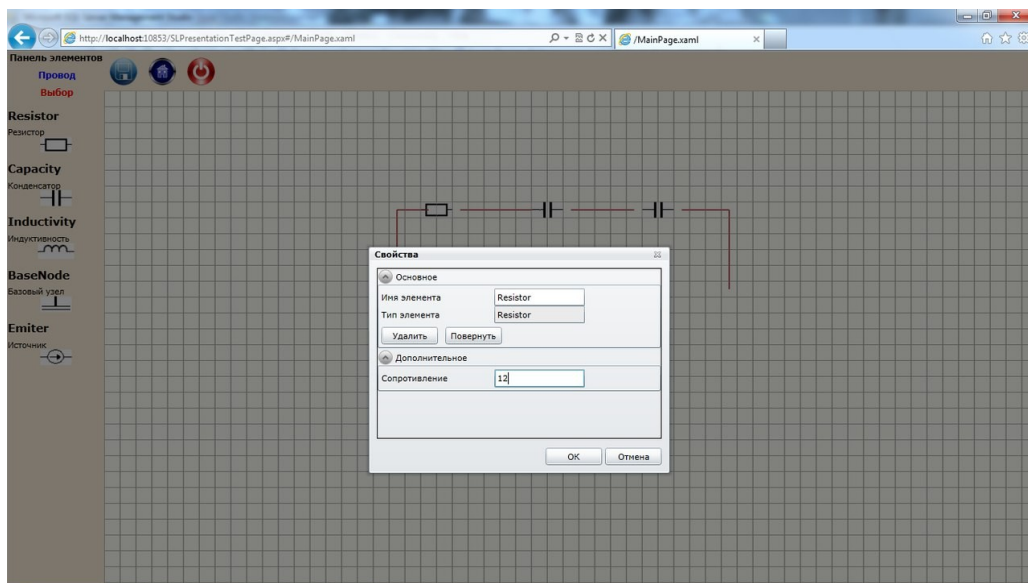


Рисунок 1 - Окно свойств элемента

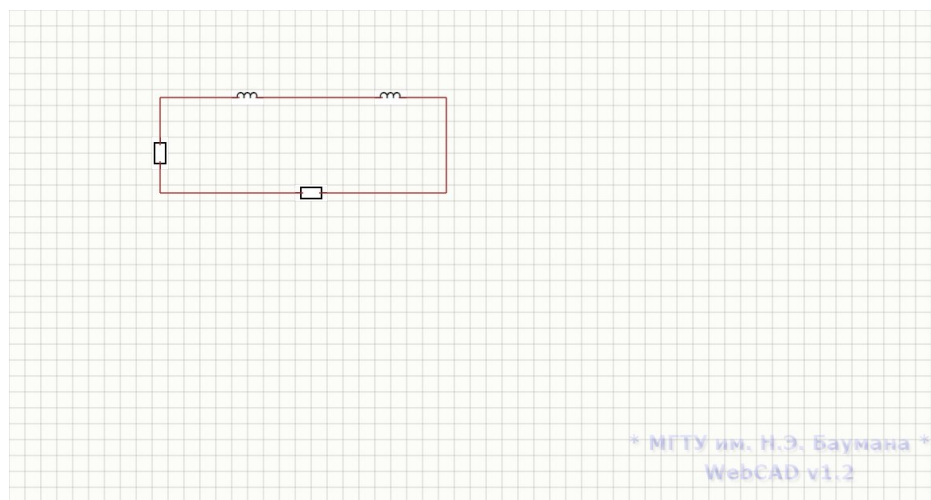


Рисунок 2 – Цифровой водяной знак в виде текста

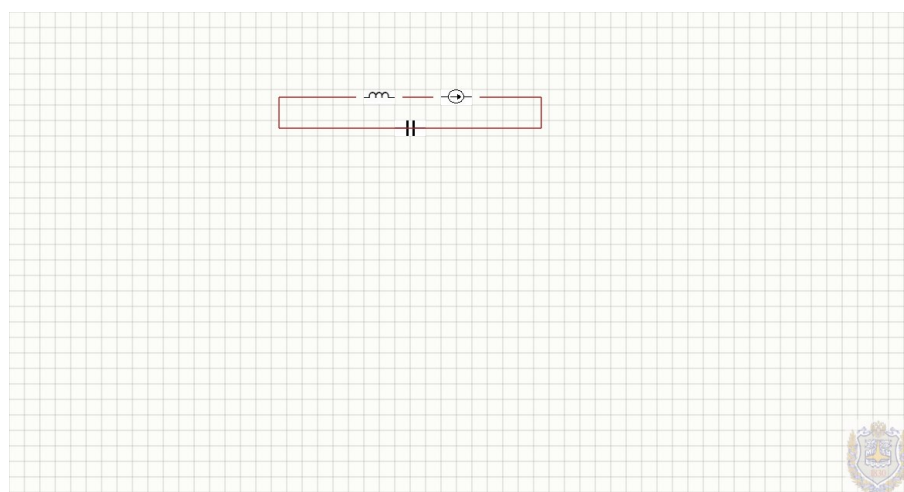


Рисунок 3 – Цифровой водяной знак в виде изображения

Функциональность данной реализации методики нанесения водяного знака не ограничивается возможностью нанести авторский логотип или текстовую информацию. Главным преимуществом является возможность варьировать параметры наносимого водяного знака, такие как его цветность и прозрачность. Задание высокой степени прозрачности позволит сделать водяной знак неразличимым для человеческого глаза, но не для компьютера. Таким образом, водяной знак может являться защитой чертежа в случае попытки открыть его на компьютере, не имеющем ПО для открытия изображений с конкретным водяным знаком. В случае попытки открыть чертеж в системе, не предназначенную для его воспроизведения или модификации, чертеж будет недоступен для пользователя.

Заключение

В результате данной работы был реализован модуль системы WebCAD для нанесения ЦВЗ. В отличие от другого программного обеспечения с подобным функционалом, пользователю не требуется сторонних приложений для работы, кроме веб-браузера с установленным плагином Silverlight. Достоинством данной реализации является простота ее использования при богатом функционале. К недостаткам можно отнести отсутствие обратной системы-системы валидации чертежей на основе соответствия ЦВЗ. В рамках данного проекта поставленные задачи были выполнены. Более того, имеется техническая возможность для расширения функционала за счет модульности приложения.

Литература

1. Троелсен Э., Язык программирования C# 2010 и платформа .NET 4.0. — М.: Вильямс, 2011. 1344 с.
2. Макдональд М., Microsoft ASP.NET 4 с примерами на C# 2010 для профессионалов. — М.: Вильямс 2010. 1424 с.
3. Конахович Г.Ф., Пузыренко А. Ю., Компьютерная стеганография. Теория и практика. — М.: мк-Пресс, 2006. 288 с.
4. Lowy J., Programming WCF Services — O'Reilly Media, 2008, 784 с.
5. Brown P., Silverlight 4 in Action — Manning Publications, 2010, 800 с.

ИЗОБРЕТАЮЩИЕ МАШИНЫ: ИСТОЧНИК ИННОВАЦИЙ

Гаврилов А.Е.

Научный руководитель: д.т.н., доцент, Резчикова Е.В.

МГТУ им. Н.Э.Баумана, кафедра ИУ4, Москва, Россия

INVENTING MACHINE: SOURCE OF INNOVATIONS

Gavrilov A.E.

Supervisor: Docent, Rezchikova E.V.

MSTU, Moscow, Russia

Аннотация

В статье рассматривается класс систем автоматизированного изобретения системы САПР, позволяющих автоматизировать изобретательскую деятельность. Кратко представлен перечень существующих на рынке программ, описаны принципы их работы и недостатки.

Abstract

In article are considered a class Computer Aided Invention of CAD systems, allowing to automate inventive activity. The list of programs existing in the market is briefly submitted, the principle of their work and shortcomings is described.

Введение

Быстрое развитие научно-технического прогресса, требует совершенствование имеющихся или создания новых продуктов и технологий. В условиях растущей открытости российского рынка, новые продукты (технологии) должны соответствовать мировому уровню качества, в противном случае они не будут востребованы рынком.

Тем временем, число специалистов, которых можно было бы привлечь к разработке новых продуктов, сокращается. Так, по сравнению с 1990г. в России число исследователей и разработчиков сократилось в 2,6 раз, а число научно-исследовательских организаций за последние 10 лет снизилось на 30% [1].

В таких условиях разработка новых продуктов высокого качества возможна только при условии существенного повышения производительности труда в области конструирования и проектирования. Одним из эффективных решений данной задачи является разработка и внедрение в деятельность организации систем автоматизированного проектирования (САПР).

Современные САПР позволяют автоматизировать оформление документации, процесс принятия решений, проводить математическое моделирование и оптимизацию параметров устройства или технологии. Однако они не позволяют разрабатывать главного – новые концепции продуктов. Известно, что резкое повышение качества продуктов возможно лишь за счет использования в них большого числа различных новаций [2]. Таким образом, можно предположить, что дальнейшее развитие САПР будет связано с автоматизацией начальных стадий проектирования и, в первую очередь, изобретательской деятельности. В настоящее время сформировался новый класс САПР Computer Aided Invention (CAI) – система автоматизированного изобретения.

Важно отметить решающее значение стадии разработки концепции на будущее разрабатываемого продукта, поскольку в этот период закладываются основы продукта, а смена концепции в дальнейшем требует дополнительных затрат. Нередко также бывает, когда одному конкуренту удается найти наиболее эффективную концепцию продукта, что позволяет ему завладеть основной долей рынка в своем сегменте.

1 Изобретающие машины

Автоматизация изобретательской деятельности – непростая задача, ведь изобретательство отождествляется с высшим проявлением творчества. Тем не менее, в последнее время разработан ряд изобретающих программ, который в той или иной мере автоматизируют разработку концепции продуктов.

К таким программам можно отнести:

- Goldfire Innovator (<http://inventionmachine.com>);
- Innovation WorkBench (<http://www.ideationtriz.com>);
- TriSolver "Idea Generator & Manager" (<http://www.trisolver.eu>);
- Creax CreationSuite (<http://www.creationsuite.com>).

К первой попытке, автоматизировать изобретательскую деятельность, можно отнести программу "Изобретающая машина" (ИМ). Она была разработана в Минске, в лаборатории изобретающих машин, руководимой В. Цуриковым. Идеи создания такого комплекса программ были высказаны еще в 1975–76 гг., но только в 1989 году была выпущена первая версия. В основу программы был положен алгоритм решения изобретательских задач (АРИЗ) предназначенный для решения нестандартных, с точки зрения теории решения изобретательских задач (ТРИЗ), задач повышенной сложности. Взаимодействие пользователя с ИМ осуществляется в виде диалога, в ходе которого строится прототип, а затем новая обобщенная модель системы, приближающая решателя к конечному оптимальному варианту.

Современные системы намного сложнее – позволяют совместно работать над одним проектом команде специалистов, позволяют проводить патентный поиск по базам различных стран, поиск по внутренним знаниям компании, по базам эффектов, и предоставляют набор инструментов для решения задач, основанных на различных методах.

Поскольку методов решения изобретательских задач существует более 300, например, методы случайного поиска, структурно-функциональные методы, методы логического поиска, проблемно-ориентированные методы и др., то каждая из перечисленных систем предоставляет свой набор инструментов для поиска инновационных решений, опирающиеся на соответствующие методы.

Алгоритмом всех выше перечисленных программ являются эвристические методы решения технических проблем, т.е. последовательность предписаний и процедур обработки информации, выполняемая с целью поиска более рациональных и новых конструктивных решений. Это приводит к двум существенным недостаткам: необходимость предварительного обучения пользователя и низкая конкретность решения технических проблем.

Таким образом, выше перечисленные методы можно условно отнести к изобретающим программам 1-го поколения. Они не решают поставленную пользователем проблему, а только подсказывают направления ее решения. Опираясь на данную подсказку, пользователь должен самостоятельно решить проблему. Хотя данные программы и освобождают пользователя от ряда рутинных операций, но все творческие операции, направленные на решение проблемы, пользователь выполняет самостоятельно.

Следовательно, данное поколение программ, хотя позволяет системно подходить к решению задач, но имеет ряд существенных ограничений, что ограничивает их применение в инженерной практике.

Ко 2-му поколению изобретающих программ можно отнести систему "Новатор 4.02" (<http://www.method.ru/>) от российской компании «Метод». Основа Новатора – формальные методы решения технических проблем и большая база конкретных знаний. С помощью данной программы даже неподготовленный пользователь сможет решать сложные технические проблемы [3, 4].

Разработку концепции продукта (технологии) Новатор рассматривает как решение технической проблемы. Этот процесс разбивается на 4-е стадии:

1. Выявление проблемы – на данной стадии пользователь составляет модель проблемной ситуации в виде блок-схемы с помощью специального графического

редактора, после чего предлагается расширить проблемную ситуацию, включив новые объекты, отношения или свойства объектов. В результате анализа расширенной проблемной ситуации формулируются ряд проблем, подлежащих решению. Важно отметить, что данный этап не является обязательным, поскольку пользователь может сразу указать проблему, подлежащую решению.

2. Анализ проблемы – для увеличения числа решений Новатор, используя базы знаний формулирует перечень проблем, содержательно похожих на исходную. Как правило, это синонимичные, общие, частные над- и подпроблемы. Также в случае сложной проблемы на данном этапе производится ее разбиение над более простыми проблемами, которые решаются поочередно.

3. Решение проблемы – для решения поставленной выше проблемы формируется последовательность запросов к базе знаний, в результате чего, находятся решения, представляющие собой описания устройств, и оформляются в виде структурированного списка. На данном же этапе система может предложить способы улучшения показателей устройств за счет изменения свойств или замены материала элементов устройств, изменения конструкции и режима работы устройства.

4. Выбор решения – на данном этапе пользователь может сравнить полученные решения не только по функциональному, но и по другим показателям устройства, для чего используется метод экспертных оценок, позволяющий получить относительный показатель качества для каждого устройства.

Таким образом, хотя Новатор и находит новые решения технических проблем, но они ограничиваются комбинацией знаний, заложенных в базе знаний системы. Начальная база знаний включает словарь терминов с более чем 2000 научно-технических терминов, базу проблем, базу эффектов с приблизительно 4500-ми эффектами и базу решений с более 1000-ю обобщенных технических решений.

Можно сделать вывод, что, хотя данная система и позволяет эффективно решать технические проблемы, но она требует постоянного пополнения базы знаний с помощью специальных редакторов, чтобы "идти в ногу со временем", а данная задача не из простых.

Заключение

В ходе работы исследованы особенности работы существующих на рынке систем, позволяющих автоматизировать изобретательскую деятельность. Несмотря на то, что приведенные системы превосходно справляются с рядом задач, были выявлены существенные недостатки, которые ограничивают их применение в инженерной практике.

В результате работы можно сделать вывод, что автоматизация изобретательской деятельности непростая задача, однако появление в последнее время большого числа САИ систем свидетельствует о высокой потребности в таких системах.

Отдельные результаты работы получены в рамках Гранта № 14.В37.21.0453 по ФЦП «Научные и научно-педагогические кадры инновационной России» на 2009 – 2013 годы.

Литература

1. Наука России в цифрах: 2011. Статистический сборник — М.: ЦИСН, 2011. – 220 с.
2. Альтшуллер Г.С. Творчество как точная наука. 2 изд., дополн. — Петрозаводск: Скандинавия, 204. – 208 с.
3. В.А. Шахнов, А.И. Власов, Л.А. Зинченко. О методическом обеспечении инженерного образования в современных условиях// Высшее образование в России. 2012. №3. С.104-108.
4. Резикова Е.В., Власов А.И. Перспективы применения концепт-карт для построения базы знаний ТРИЗ // Сборник трудов конференции "ТРИЗ. Практика применения методических инструментов". - Москва. 29 октября 2011. С.140-145.

СРАВНЕНИЕ AUTOCAD REVIT И BENTLEY AECOSIM BUILDING DESIGNER

Григорьев В.Ю., Бутаков А.С., Усенко Е.С.

Научные руководители: к.т.н., доцент, Маничев В.Б., к.т.н., доцент, Жук Д.М.

МГТУ им. Н.Э.Баумана, кафедра РК-6, Москва, Россия

COMPARISON AUTOCAD REVIT AND BENTLEY AECOSIM BUILDING DESIGNER

Grigorev V.Y., Butakov A.S., Usenko E.S.

Supervisors: Ph.D., Associate professor, Manichev V.B., Ph.D., Associate professor, Zhuk D.M.

BMSTU, RK6 Department, Moscow, Russia

Аннотация

В статье рассматривается краткий обзор САПР двух крупнейших компаний в области проектирования инженерных систем и строительных конструкций зданий. Первый программный комплекс информационного моделирования зданий – Autodesk Revit Architecture, второй комплекс – Bentley AECOSim Building Designer. Также кратко представлена история развития этих двух инструментов проектирования. В заключении представлены сильные и слабые стороны обеих систем.

Abstract

A brief overview of two major CAD companies in the area of engineering systems designing and building constructions discusses in this article. The first software complex of building information modeling - Autodesk Revit Architecture, the second set - Bentley AECOSim Building Designer. Also, briefly presented the history of these two design tools. In conclusion, submitted the strong suit and weak side of both systems.

Введение

Эра применения компьютеров в проектировании началась в 60-х годах прошлого столетия. Первым большим достижением было создание в США графического устройства SketchPad в Lincoln Laboratory. В 1965 году совместными усилиями автогиганта General Motors, фирмы ИТЕК и крупнейшего производителя компьютеров корпорации DEC была создана 1-ая Электронная Чертежная Машина. В 1975 году по заказу крупнейших производителей автомобилей (GM, Ford, Chrysler) и самолетов (Lockheed) был разработан мощный пакет автоматизированного проектирования UNIGRAPHICS.

В конце 80-ых годов появились разработки, кардинально изменившие подход к автоматизированному проектированию зданий. В его основе – переход от графического (чертежного) проектирования к компоновке зданий из трехмерных параметрических объектов. Архитектор Фил Бернштейн (Phil Bernstein) – стратег автоматизации проектирования зданий и вице-президент фирмы AutoDesk – назвал этот подход информационным моделированием зданий – Building Information Modeling (BIM).

Делая выбор в пользу той или иной САПР, следует, прежде всего, убедиться, удовлетворяет ли она следующим требованиям:

- Надёжность. Приобретая систему с нестандартным форматом хранения информации, инженеры совершают большую ошибку.
- Доступность. Если после приобретения САПР проектировщик не сможет пройти необходимого обучения или не на высоте окажется техническая поддержка, то нормальная работа любой САПР маловероятна.
- Открытость. Если система «сама в себе» и её нельзя настроить или доработать под потребности проектировщика, то выбор такой системы – ошибка.
- Стабильность. Система не должна постоянно менять основные принципы работы. Работа в САПР – это обычаи и своды правил. Поэтому только постоянство стиля САПР позволит вовремя и качественно выполнять обязательства.

- Масштабируемость. Хорошая САПР удовлетворит любой кошелёк. Она позволит вам начать с «малого решения» и со временем расширить систему до желаемого уровня.

В этой работе последовательно рассмотрены две крупнейшие системы в области проектирования инженерных систем - Autodesk Revit Architecture и Bentley AECOsim Building Designer.

1 Обзор Autodesk Revit Architecture

При подготовке обзора были использованы источники: [1], <http://www.iesoft.ru>, <http://www.cad.ru>.

В Autodesk Revit Architecture существуют инструменты моделирования строительства, которые позволяют координировать проектную и строительную модель в пределах одного файла. Новый инструмент “Create Parts/Divide Model” позволяет разделять геометрию модели на части (рис. 1). Данная функция очень удобна, когда необходимо создать спецификации или ведомости материалов для определенной части модели, выполнить экспорт или сделать визуализацию определенной части проекта.

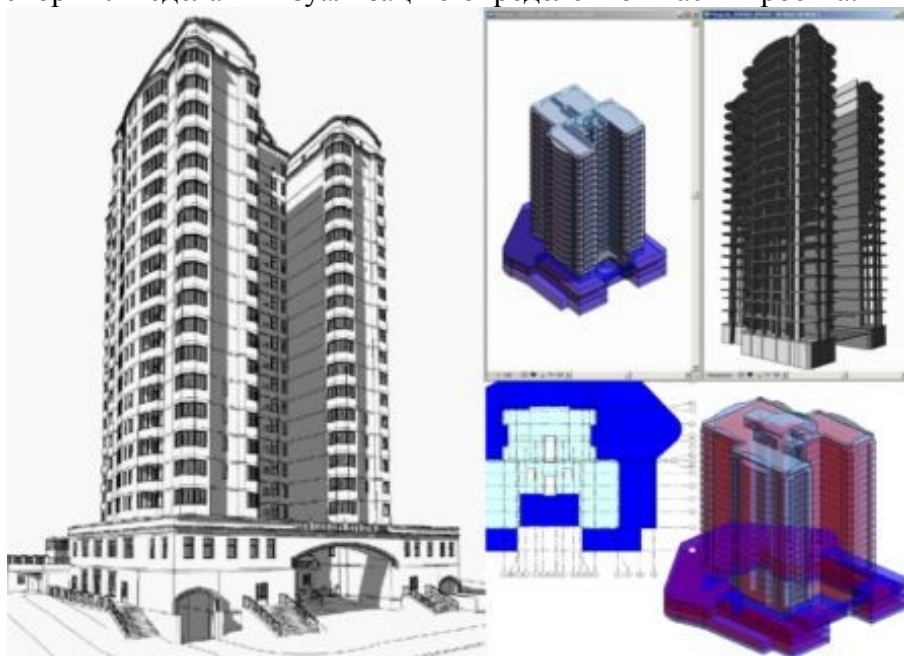


Рисунок 1 – Трехмерная типовая модель, открытая в интерфейсе Autodesk Revit Architecture

В этой системе существует возможность выделения сборок объектов, таких как сборные стены, колонны, балки и перекрытия. Со сборками можно выполнять следующие действия: выбирать, создавать спецификации, изолировать, маркировать и отслеживать в проекте. Возможности новой версии продукта позволят создавать четырехмерные графики строительных работ. Данная функция впервые стала доступна как дополнительный модуль в рамках программы подписки Autodesk Revit Architecture 2011. В Autodesk Revit Architecture 2012 функция доступна в рамках основных возможностей продукта.

В системе существует глобальная база климатических данных, содержащая свыше 1,5 миллионов наборов метеоданных. Проектировщики могут выполнять расчеты с использованием технологии облачных вычислений от [Green Building Studio](http://www.greenbuildingstudio.com) в ходе рабочего процесса. Для более удобного анализа полученных данных результаты расчета выводятся в графическом формате. Предусмотрена возможность сравнения полученных данных для нескольких вариантов проекта.

Очень важным фактором в Autodesk Revit Architecture 2012 является инструмент работы с облаками точек, который позволяет использовать данные лазерного сканирования при работе с моделью. Данная функция позволит упростить и ускорить работу с проектами

реконструкции и модернизации. В программе присутствуют функции перезагрузки, выгрузки и удаления данных объектов. Облако точек в программе ведет себя как отдельный объект, что позволяет привязываться стандартными инструментами Autodesk Revit Architecture к поверхностям, которые динамично выделяются в облаке точек.

Увеличение производительности работы Autodesk Revit Architecture 2012 положительно сказалось на обработке больших проектов в Revit. Присутствует возможность очистки чертежей от неиспользуемых категорий импорта и стилей объектов. Загрузка элементов в память осуществляется многопоточно, что позволяет сократить время открытия видов при первом отображении элементов в сеансе работы.

Revit Server является основой коллективной работы над проектами Revit с использованием сервера.

Для наиболее оптимальной организации совместной работы по глобальной сети Revit Server использует конфигурацию: Центральный сервер и нескольких локальных серверов. Объединенные модели Revit хранятся на центральном сервере, доступом к которому с локальных серверов могут пользоваться все специалисты проектной группы.

2 Обзор Bentley AECOSim Building Designer

При подготовке обзора были использованы источники: [3], [4], <http://www.cadalyst.com>.

Несколько месяцев назад, Bentley выпустила новую версию своего приложения для проектирования зданий, под именем AECOSim Building Designer. Новая версия сочетает в себе все разнообразные приложения Bentley BIM по различным дисциплинам в одном продукте. Под дисциплинами понимаются четыре различных приложения для архитекторов, инженеров-строителей, инженеров-механиков и инженеров-электриков. Открывая общее приложение, AECOSim Building Designer открывает все наборы инструментов, которые ранее содержались в этих отдельных дисциплинарных инструментах, как показано на рисунке 2. Здесь можно увидеть все индивидуальные наборы инструментов в левой части окна интерфейса.

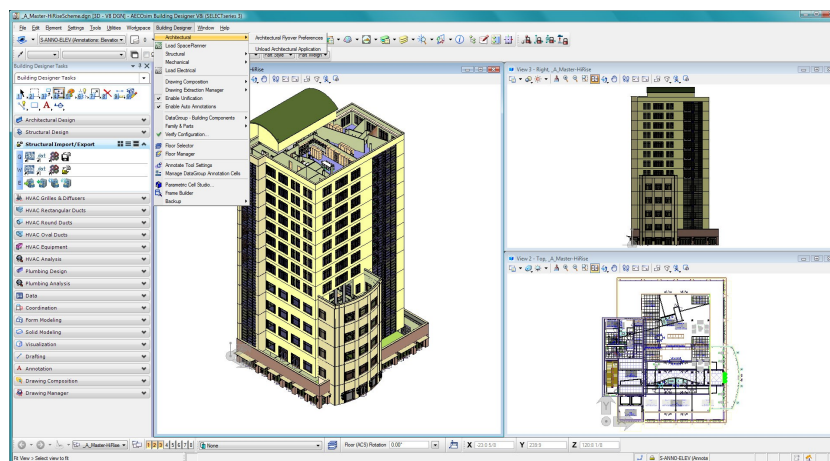


Рисунок 2 – Образец модели в интерфейсе AECOSim Building Designer, который включает в себя все дисциплинарные наборы инструментов.

Помимо интеграции различных дисциплин в одно приложение BIM, не так много было добавлено Bentley в уже созданные ранее обширные возможности, которые включают в себя полный набор инструментов рисования и средств аннотирования, твердотельного и поверхностного моделирования, визуализаций и анимаций и прочее. Большинство инструментов в отдельных наборах такие же, за исключением нескольких, таких как возможность создания различных видов крыш, водосточных труб и поручней в архитектурном наборе инструментов.

Учитывая, что AECOsim Building Designer представляет собой интегрированное приложение BIM, оно также создает возможности для междисциплинарной координации. Это дает возможность пользователю определить набор бизнес и графических элементов и обнаружить геометрическое столкновение между множеством элементов (рис. 3). Проектировщики могут интерактивно и наглядно рассмотреть столкновения, комментировать и помечать их. Результаты пересечения могут быть сгруппированы, размечены и экспортированы в таблицу Excel или системы GPS.

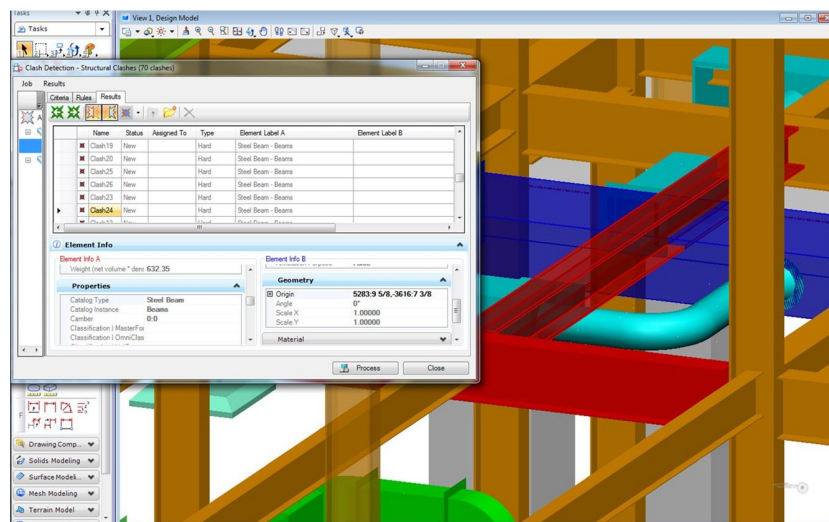


Рисунок 3 – Новая утилита обнаружения столкновений (пересечений) в AECOsim Building Designer

Еще одной новой особенностью является то, что в иностранной литературе известно как hypermodeling. По существу это система, которая автоматически устанавливает маркер в модель, где чертежи носят не только дополнительный информационный характер, но и существует ссылка на соответствующий чертеж или лист. Эти hypermodels также можно увидеть, когда эта модель экспортируется в Bentley Navigator, которая представляет собой приложение для навигации, просмотра и разметки модели, похожие на Autodesk Navisworks.

Несмотря на все улучшения, эта система является очень трудной для интуитивного пользования и научиться эффективно использовать этот инструмент проектирования без серьезных затрат времени и усилий практически не представляется возможным. Но для тех, кто уже хорошо знает решения Bentley, открываются возможности в полной мере воспользоваться всеми преимуществами этой глобальной системы для того, чтобы сделать выдающуюся работу.

Заключение

Bentley AECOsим Building Designer, по существу построенный на вершине системы MicroStation, который имеет как плюсы, так и минусы. С положительной стороны, он наследует весь спектр мощных возможностей MicroStation, и продолжает удовлетворить тех, кто вложил годы обучения и работы в MicroStation. Но отрицательной стороной является то, что он вынужден нести много багажа САД приложения, таких как слои, ссылки, несколько файлов для разных конструкций, элементов пользовательского интерфейса, и так далее. Он не может начать с чистого листа, и быть проворным и интуитивно понятным. Напомним, что с начала движения BIM, Bentley всегда решительно выступал за "эволюционный", а не "революционный" подход к BIM, и это именно то, что мы получаем с AECOsим Building Designer.

Десятки САПР были предложены архитекторами за последние 20 лет. Почти все они приняли, по крайней мере, 2D САД, а растущие из года в год дизайнерские фирмы сделали выбор в пользу более продвинутой 3D-технологии проектирования зданий, из которых Autodesk, Microstation Bentley Systems, ArchiCAD Graphisoft.

Проблема выбора комплексов является актуальной по настоящее время. На предприятиях, а также на частных фирмах изо дня в день повышаются требования к качеству проектного решения. Даже незначительное уменьшение трудозатрат в одном технологическом процессе дает большой экономический эффект.

Литература

1. Лисяк В.В., Лисяк Н.К., Лисяк М.В. Обзор распространенных в России САПР фирмы AUTODESK // Известия ЮФУ. Технические науки. 2010. С. 70-76.
2. Удлер Е.М., Пекерман Э.Е. Проблемы обучения дисциплине САПР в архитектурно-строительном вузе // Теория и методика профессионального образования. 2012. С. 535-539.
3. Khemlani L. Autodesk and Bentley's Unprecedented Interoperability Agreement // AECbytes "Building the Future" Article. 2008. [PDF] (<http://www.aecbytes.com>).

ПОСТРОЕНИЕ КАСКАДА МЕТОДОВ ДЛЯ РАСПОЗНАВАНИЯ ЭМОЦИОНАЛЬНОГО СОСТОЯНИЯ ПО МИМИКЕ ЛИЦА ЧЕЛОВЕКА

Мокров А.М.

Научный руководитель: к.т.н., доцент, Каганов Ю.Т.
МГТУ им. Н.Э.Баумана, кафедра ИУ9, Москва, Россия

DESIGN CASCADE METHODS FOR RECOGNITION OF SHOW EMOTION BY FACIAL EXPRESSION OF PERSON

Mokrov A.M.

Supervisor: Dr., As. Prof., Kaganov Y.T.
MSTU, Moscow, Russia

Аннотация

В статье рассматриваются вопросы разработки программного продукта для осуществления видеоконтроля и распознавания лица человека. Разработанная программная система основана на использовании последовательности алгоритмов, позволяющих выделить наиболее характерные черты человеческого лица и идентифицировать это лицо, как принадлежащее определённому индивиду. Подробно исследовано взаимодействие алгоритмов распознавания и выявлены их достоинства и недостатки в контексте программной системы. В качестве заключительной подсистемы использованы алгоритмы искусственных нейронных сетей типа карты Кохонена и звезды Гроссберга, что позволяет значительно повысить эффективность работы системы по сравнению с традиционными сетями типа персептрона.

Abstract

In this article have been described methods and approaches for designing program system of video monitoring and recognition of person's face. This program system is based on using algorithms for pick out specific personality traits and identification human. Interaction algorithms of recognition was investigated in detail and merits and demerits were detected in the context of program system. The Kohonen's maps and Grosberg's stars have been used as artificial neural nets algorithms. These algorithms are very efficient in comparison with traditional neural nets as perceptrons

Введение

В настоящее время системы распознавания лиц и мимики широко используется в различных сферах деятельности человека. Особенно в медицине, финансовой деятельности, криминалистике, психологии, нейрофизиологии, в информационных и компьютерных технологиях. Исходя из анализа аспектов применения системы распознавания лиц нетрудно сделать вывод о том, что основным критерием проектирования подобных алгоритмов и систем распознавания является оптимальность с точки зрения соотношения скорости и эффективности работы программного обеспечения, выполнения анализа изображения и последующей обработки данных. Это в свою очередь должно быть сравнимо с корректностью получаемого решения поставленной задачи. В последнее время в сфере решения проблемы скорости обработки и категоризации данных подают надежду нейронные сети в силу их схожести со строением нейронов головного мозга и их связей, а также с достаточно широкими возможностями самообучения и последующего корректирования выходных данных [1,2].

Целью работы является разработка эффективного алгоритма и программного обеспечения распознавания лица человека и выявления особенностей эмоционального состояния, соответствующего мимическому образу человека.

1 Постановка задачи

Приведём основные задачи данной работы:

- выбор некоторого набора выражений лица;

- определение необходимых реперных точек, совокупность которых однозначно характеризует выбранные мимические жесты;
- реализация процесса определения данных точек;
- создание внутреннего представления и обработка рассматриваемых точек;
- выбор подходящей нейронной сети для решения данной задачи;
- выполнение категоризации мимического жеста на основе полученных данных посредством нейронной сети;
- анализ полученных результатов.

Выделенные пункты постановки задачи последовательно проанализированы и предложены процедуры алгоритма распознавания.

2 Этапы решения задачи

Для детального анализа проблема распознавания образов разбита на отдельные этапы для которых могут использоваться разные алгоритмы, например, с использованием нейронных сетей и других методов искусственного интеллекта.

На первом этапе необходимо осуществить поиск лица на изображении, затем, рассматривая конкретную область, найти и обработать реперные точки, по которым будет впоследствии распознаваться изображение. Последним шагом является процесс классификации образов на основе нейронной сети.

Первый этап

В качестве алгоритма для обнаружения лиц был использован широко распространённый и достаточно оптимальный [4,5] метод поиска, предложенный по результатам исследований Виолы и Джонса (Viola&Jones).

Перечислим основные концепции данного алгоритма.

1. Использование интегрального представления изображений.
2. Использование признаков (функций) Хаара.
3. Использование алгоритма адаптивного усиления (AdaBoost).
4. Использование каскадного классификатора.

Алгоритм Виолы-Джонса реализован в готовой библиотеке OpenCV.

В основе алгоритма Виолы-Джонса лежит принцип сканирующего окна, согласно которому на вход поступает представление изображения в виде двумерной матрицы пикселей следующего размера «ширина_изображения * высота_изображения». При этом каждый элемент матрицы хранит информацию о цвете соответствующей единицы изображения. На выходе получаем множество прямоугольников вида:

$$r_i = \{x, y, w, h, a\}, \quad (1)$$

где (x, y) — координаты левой нижней вершины i -го прямоугольника, содержащего искомую область, w — ширина, h — высота, а параметр a — угол наклона прямоугольника. Таким образом, сканирование всего изображения осуществляется посредством окна сканирования, применяемого ко всем участкам изображения и к разным масштабам анализируемого изображения с последующим использованием функции Хаара, а также классификатора на каждой итерации. Стоит заметить, что данный алгоритм может быть применён для обнаружения не только лиц, но и других объектов, что достигается корректированием соответствующих функций и созданием новых примитивов Хаара.

Интегральное представление изображения позволяет быстро рассчитывать среднюю интенсивность (яркость) произвольного прямоугольника, это свойство потребуется при использовании функций Хаара. Известно, что интегральное представление изображения является матрицей размера «ширина_изображения * высота_изображения», каждый элемент матрицы $R_{a,b}$ которой можно представить в виде:

$$R_{a,b} = \sum_{i=0, j=0}^{i \leq a, j \leq b} I_{i,j} \quad (2)$$

где $I_{i,j}$ — яркость (i, j) пикселя анализируемого изображения.

Подсчёт значений элементов данной матрицы занимает линейное время и осуществляется в один проход по анализируемой матрице изображения. Подсчёт значения интегрального представления изображения на каждом следующем пикселе проводится по формуле:

$$R_{a,b} = I_{a,b} - R_{a-1,b-1} + R_{a,b-1} + R_{a-1,b} \quad (3)$$

Для того, чтобы вычислить суммарную интенсивность на выбранном прямоугольнике достаточно воспользоваться формулой:

$$S = R_{x1,y1} - R_{x2,y2} - R_{x3,y3} + R_{x4,y4} \quad (4)$$

где $A=(x1, y1)$, $B=(x2, y2)$, $C=(x3, y3)$, $D=(x4, y4)$.

Для определения искомого среднего значения яркости необходимо поделить суммарную интенсивность на площадь прямоугольника. Для анализа изображений в методе Виолы-Джонса используются функции Хаара. Задача классификации представляет собой процесс обработки значений функции Хаара и получения ответа о принадлежности к определённому классу. В данном случае существует всего два класса «есть лицо» и «нет лица». Для получения верного ответа о принадлежности классу требуется построить и обучить классификатор, то есть выбрать верные пороговые значения для каждого из примитивов Хаара. Для выбора требуемых функций Хаара и установления пороговых значений перепада по яркости использован метод машинного обучения AdaBoost [6]. Алгоритм данного метода представляет собой процедуру усиления «слабых» моделей классификации, то есть таких моделей, которые делают большое количество ошибок во время выполнения, посредством их композиции и объединения в более сильные классы. Фактически данный алгоритм заключается в построение цепочки классификатора (каскада), каждый элемент которого обучается на ошибках предыдущего. Результат же определяется путём голосования: пример считается принадлежащим тому классу, который был получен в качестве результата большинства моделей каскада.

Второй этап

Алгоритм процесса адаптации активной модели внешнего вида к анализируемому изображению может быть представлен в следующем виде:

- вычисляются все параметры модели, а также преобразования формы на основе начального приближения;
 - вычисляется вектор ошибки δt ;
 - производится построение текстуры на основе рассматриваемого изображения путём кусочных деформаций;
 - вычисляется вектор возмущений δp ;
 - модифицируется вектор комбинированных параметров и преобразований путём суммирования его с вектором возмущений;
 - производится обновления параметров формы и текстуры;
- повторение всех действий, начиная с пункта , до достижения сходимости.

Данный алгоритм довольно точно определяет точки активной модели внешнего вида, однако, как показывают практические расчёты, процесс долго сходится и для его выполнения требуется большое количество итеративных вычислений, в следствие чего был предложен новый алгоритм с использованием обратной композиции [6, 7].

Третий этап

Так как задачей данной работы является строгая категоризация данных, то в качестве сети для разработки алгоритма определения выражения лица была выбрана сеть Кохонена-Гроссберга. Нейронная сеть Кохонена-Гроссберга или сеть встречного распространения представляет собой комбинацию двух алгоритмов: самоорганизующейся карты Кохонена и звезды Гроссберга. В основе создания подобных сетей и результатов их анализа заложено предположение о том, что человеческий мозг построен именно на каскадных связях разных сетей по причине более внушительных результатов при использовании сетей с такими связями по сравнению с сетями, основанными на классической структуре [8, 9].

Заключение

В работе проведён анализ готовых библиотек для работы с изображениями, приведено детальное исследование отдельных этапов реализации алгоритмов в них реализованных, в частности, проанализированы процедуры необходимые для обнаружения лица, процедуры поиска реперных точек, а также рассмотрены этапы обучения алгоритма поиска реперных точек. В результате выполнения данной работы получен вывод о возможности обеспечения безотказной работы системы распознавания образов при использовании самообучающихся нейронных сетей для решения задач распознавания выражения лица. Построенная нейронная сеть, в отличие от описанных стандартных методов, решает проблемы выбора аспектов поведения системы при возникновении незапрограммированных ситуаций. При применении предложенных алгоритмов в системах видеоконтроля, можно гораздо эффективнее выявлять чрезвычайные ситуации, анализ и распознавание которых невозможно при использовании статических систем. Нейронные сети на основе карты Кохонена позволяют развивать и модифицировать систему распознавания образов благодаря возможности комбинации и взаимосвязи различных сетей. По сравнению с классической нейронной сетью, основанной на персептроне, сеть Кохонена работает значительно эффективнее, однако замечено, что в некоторых случаях скорость обучения заметно уступает скорости обучения обычной сети, особенно при схожести входных значений эталонных наборов.

Литература

1. Lakhmi C. Jain, N.M. Martin «Fusion of Neural Networks, Fuzzy Systems and Genetic Algorithms: Industrial Applications». — CRC Press, CRC Press LLC, 1998.
2. Классификация известных нейросетей по основным категориям применения [Электронный ресурс] — Режим доступа: <http://www.masters.donntu.edu.ua/2008/fvti/shakhovaya/library/index8.htm>
3. Власов А.И., Шахнов В.А. НЕЙРОКОМПЬЮТЕРЫ — АРХИТЕКТУРА И РЕАЛИЗАЦИЯ [Электронный ресурс]. - Режим доступа: http://citforum.ru/hardware/neurocomp/neurocomp_02.shtml
4. Finding Faces in Images [Электронный ресурс] — Режим доступа: http://www.cognotics.com/opencv/servo_2007_series/part_2/index.html
5. P. Viola and M.J. Jones Rapid Object Detection using a Boosted Cascade of Simple Features // proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR 2001), 2001
6. S. Baker, R. Gross, and I. Matthews. Lucas-Kanade 20 years on: A unifying framework: Part 3. Technical Report. — CMU-RI-TR-03-35, Carnegie Mellon University Robotics Institute, 2003.
7. Iain Matthews and Simon Baker Active Appearance Models Revisited. — International Journal of Computer Vision, Vol. 60, No. 2, November, 2004, pp.135–164.
8. T. Cootes, G. Edwards, and C. Taylor Active appearance models», Proceedings of the European Conference on Computer Vision, volume 2, 1998, pp. 484–498.
9. Хайкин С. Нейронные сети: полный курс. — М.:Вильямс, 2006.

ИССЛЕДОВАНИЕ АЛГОРИТМА МНОГОЦЕЛЕВОЙ ОПТИМИЗАЦИИ ХИЩНИК-ЖЕРТВА

Бушневский А.В.

Научный руководитель: д.ф.-м.н. профессор Карпенко А.П.

МГТУ им. Баумана, кафедра РК6, Москва, Россия

STUDY OF PREY-PREDATOR GENETIC ALGORITHM FOR MULTI-OBJECTIVE OPTIMIZATION PROBLEMS

Bushnevskiy A.V.

Supervisor: Dr., Prof., Karpenko A.P.

MSTU, Moscow, Russia

Аннотация

Для решения задач многоцелевой оптимизации часто применяются эволюционные алгоритмы. Основным преимуществом этих алгоритмов является возможность нахождения набора решений, оптимальных по Парето, позволяя выбрать лучшее из компромиссных решений. Статья описывает модель генетического алгоритма, вдохновленного взаимодействием хищников и жертв в естественной природной среде. Рассматривается базовая модель алгоритма, возможные его модификации, а так же их влияние на поведение алгоритма. Эффективность алгоритма проверяется на тестовых задачах ZDT.

Abstract

Evolutionary algorithms are often used for finding a solution of multi-objective optimization problems. The main advantage of evolutionary methods of evolutionary multi-objective optimization methods is that they allow finding a set of Pareto-optimal solutions, allowing to choose the best one of the compromise solutions. The article describes the genetic algorithm model, inspired by predator and prey interaction in the natural environment. The basic algorithm model, several modifications of the algorithm and the influence on its performance are considered. The performance of the modified prey-predator model is tested on a standard ZDT1 problem.

Введение

Для решения задач многоцелевой оптимизации обычно прибегают к расширению стандартных методов эволюционных алгоритмов (EA), однако профессор *Hans-Paul Schwefel* из университета Dortmund, Германия представил несколько иную и более естественную концепцию. В 1998 году он и его соавтор представили модель хищник-жертва для нахождения множественных решений, оптимальных по Парето. В предложенном ими алгоритме (который называется базовым) использовался новый метод выборки решений, навеянный отношениями хищник-жертва. В этом методе каждая жертва представляет один пространственный вектор решений, а каждый хищник представляет одну целевую функцию. Алгоритм имитирует естественный феномен поглощения хищником наиболее слабой жертвы, что означает, что хищник ликвидирует худшую жертву в своей окрестности, которая соотносится с худшим значением целевой функции. Для осуществления этой идеи авторы предложили использовать тороидальную сетку, в узлах которой случайным образом инициализируются жертвы и один (или более) хищников. После этого каждый хищник рассматривает все жертвы (решения) в окрестности и удаляет жертву, соответствующую худшему значению целевой функции. Затем в окрестности хищника выбирается и изменяется случайная жертва. Измененное дочернее решение помещается на место удаленной жертвы. Хищник делает случайное перемещение в один из соседних с ним узлов. Шаг повторяется для каждого из хищников. Вместе с удалением худших решений алгоритм выделяет лучшие решения. Результаты предварительной симуляции показали, что начальная случайная популяция стремится к фронту решений, оптимальных по Парето.

1 Постановка задачи и схема алгоритма

Рассмотрим задачу многокритериальной оптимизации (МКО-задачу) вида

$$\min_{X \in R^n} \Phi(X) = \Phi(X^*) = \Phi^*, X \in D_x, \quad (1)$$

где X – вектор варьируемых параметров задачи; D_x – ограниченное и замкнутое множество допустимых значений вектора X ; $\Phi(X) = (\phi_1(X), \dots, \phi_n(X))$ – вектор целевых функций. Лицо, принимающее решение (ЛПР), стремится найти такой вектор X^* – искомое решение МКО-задачи, которое минимизирует на множестве D_x каждую из целевых функций.

Рассмотрим базовую модель хищник-жертва шаг за шагом.

- Инициализировать набор жертв случайным образом в пределах допустимых значений переменных.
- Поместить жертв в вершины ненаправленного связного графа.
- Поместить хищников случайным образом в вершины графа.
- Сопоставить каждого хищника с целевой функцией таким образом, что каждая целевая функция отождествлена, по крайней мере, с одним хищником.
- Оценить жертвы вокруг каждого хищника и выбрать худшую жертву. Например, в задаче минимизации целевых функций худшей жертвой будет та, для которой целевая функция, сопоставленная с этим хищником, имеет наибольшее значение.
- Выбранные жертвы поглощаются хищниками, т.е. худшие жертвы удаляются и заменяются их потомками.
- Создать потомка путем изменения случайно выбранной жертвы в окрестности худшей жертвы, которая была выбрана хищником.
- Хищники совершают случайное перемещение в соседнюю вершину от текущего положения хищника.
- Завершить одно поколение алгоритма хищник-жертва. Повторить шаги 5 – 8 для следующих поколений.

С ростом числа поколений популяция жертв стремится к фронту решений, оптимальных по Парето.

2 Модификации алгоритма хищник-жертва

Исследования авторов показывают, что скорость сходимости базового алгоритма была достаточно высока. Однако алгоритм не обеспечивает сохранение лучших решений, то есть с ростом числа хищников, некоторые решения, близкие к фронту решений, оптимальных по Парето, могут быть удалены, если они были помещены в окрестности некоторых других хищников.

Для улучшения работы алгоритма *Kalyanmoy Deb* предложил следующую модификацию. Хищники сопоставляются с весовыми векторами, вместо отдельных целевых функций. Например, для двухцелевой задачи создается n хищников, каждый из которых сопоставляется с одним из весовых векторов, равномерно распределенных между $[0; 1]$ и $[1; 0]$. Таким образом для случая с двадцатью одним хищником, первому хищнику соответствует вектор $[0; 1]$, второму хищнику – вектор $[0,05; 0,95]$ и так далее до двадцать первого хищника, которому соответствует вектор $[1; 0]$.

Для того, чтобы предотвратить ситуацию, когда хищники полностью уничтожают популяцию жертв и для поддержания числа жертв на допустимом уровне может использоваться формула

$$iterations = \left\lfloor \frac{numPrey_{actual} - numPrey_{preferred}}{num_{predators}} \right\rfloor, \quad (5)$$

где $iterations$ – число перемещений, которые могут сделать хищники перед тем, как жертвы могут сделать свои перемещения; $numPrey_{actual}$ – текущее число жертв; $numPrey_{preferred}$ – желаемое число жертв; $num_{predators}$ – число хищников. По существу, формула (1) используется для того, чтобы текущее число жертв не слишком

отличалось от желаемого числа жертв. Хищники стремятся минимизировать разницу между этими двумя значениями. Оператор округления в меньшую сторону $\lfloor \cdot \rfloor$ используется для того, чтобы хищники полностью не уничтожили популяцию жертв. Это соотносится с тем фактом, что в природе хищники зачастую быстрее своих жертв. Другое достоинство использования формулы (1) – то, что, как только достигается минимальное значение числа жертв, или другими словами, хищники становятся 'медленнее', новые жертвы имеют лучший шанс на выживание. В результате число жертв начнет расти, вместо того, чтобы продолжать уменьшаться. Рост будет продолжаться до того момента, когда эффект от применения формулы опять не будет в пользу хищников. Таким образом, формула (1) реализует механизм динамического изменения размера популяции жертв.

3 Вычислительный эксперимент

В качестве целевых функций использовались следующие стандартные тестовые *ZDT* функции для n переменных.

Zitzler-Deb-Thiele 1:

$$\begin{aligned} \Phi_1 &= x_1, \\ \Phi_2 &= g \cdot \left(1.0 - \sqrt{\frac{\Phi_1}{g}} \right), \\ g(x_2, \dots, x_n) &= 1.0 + \frac{9}{n-1} \sum_{i=2}^n x_i, \\ 0 &\leq x_i \leq 1, i = 1, \dots, n. \end{aligned}$$

Zitzler-Deb-Thiele 2 [2].

Рисунок 1 показывает аналитический фронт Парето и финальную популяцию для задачи *ZDT1*.

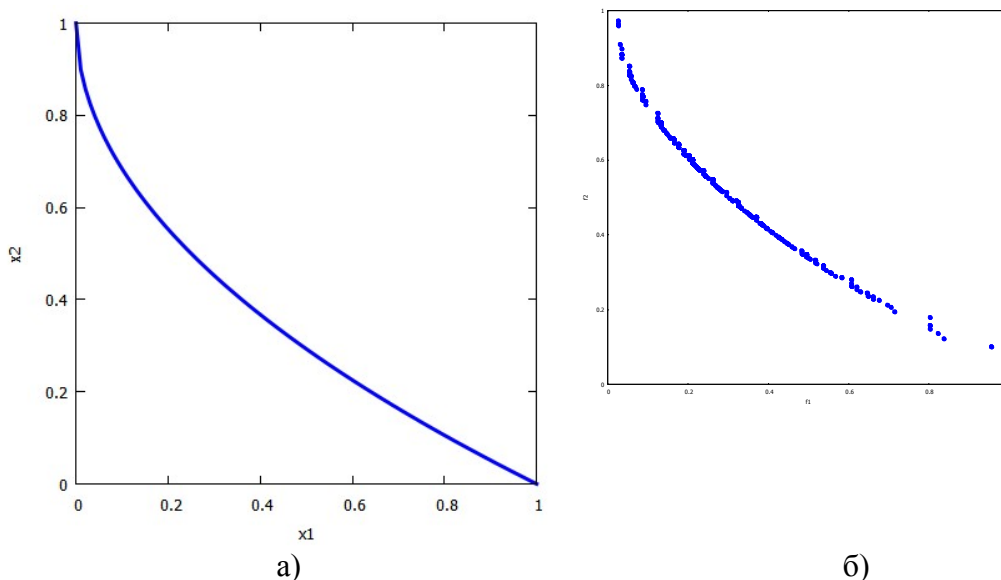


Рисунок 1 – а) Аналитический фронт Парето для задачи *ZDT1*; б) Аппроксимация того же фронта решений с помощью модифицированного алгоритма хищник-жертва: размер сетки 50 \times 50

Базовый алгоритм с описанными модификациями находит набор решений для задач *ZDT1* и *ZDT2*. Выпуклость фронта решений, оптимального по Парето, по все видимости, не влияет на предложенный метод. С увеличением размерности сетки, решения сходятся к фронту решений, оптимальному по Парето.

Заключение

В работе рассмотрен и реализован базовый алгоритм хищник-жертва. Идея использования различных хищников, охотящихся за жертвами, была успешно применена для решения задач многоцелевой оптимизации. Были реализованы модификации алгоритма для сохранения лучших решений и поддержания числа жертв. Несмотря на то, что модифицированный метод не имеет простой и естественной схемы, как базовая модель, он имеет высокую скорость схождения и находит равномерно распределенный набор решений в области решений, оптимальных по Парето.

Литература

1. Kalyanmoy Deb, Udaya Bhaskara Rao Investigating Predator-Prey Algorithms for Multi-Objective Optimization / In: KanGAL Report No. 2005010, 2005, India.
2. Xiaodong Li A real-coded predator-prey genetic algorithm for multiobjective optimization / In Lecture Notes in Computer Science, 2003, pp. 207-221.

РАЗРАБОТКА ПО АРМ ДЛЯ ЗАЩИТЫ ДАННЫХ В САПР

Круглая Е.И., Пилипенко А.В.

Научный руководитель к.т.н. доцент Волосатова Т.М.

МГТУ им. Н.Э.Баумана, кафедра РК-6, Москва, Россия

SOFTWARE DEVELOPMENT FOR AUTOMATED DATA PROTECTION IN CAD SYSTEMS

Kruglaya E.I., Pilipenko A.V.

Supervisor: Ph.D. assistant professor Volosatova T.M.

BMSTU, Moscow, Russia

Аннотация

Данная статья посвящена исследованию алгоритмов стеганографии. В работе описаны алгоритмы Коча и Бенхама, представлена их программная реализация. Результаты работы созданного программного средства также приведены в статье. На основании результатов работы программы проведено исследование возможности использования данных методов для защиты данных в САПР. Даны рекомендации по использованию данных алгоритмов.

Abstract

This article deals with the investigation of steganographic algorithms. Koch and Benham algorithms were described in the article along with their software implementation. Examples of software's output were also presented. The possibility of applying these techniques to data protection in CAD systems was analyzed on the basis of these examples. Recommendations on the use of the specified algorithms were given.

Введение

Данная работа посвящена проблеме защиты данных, созданных с помощью различных систем САПР. Специфической особенностью данной задачи является то, что значительная часть проектной документации – чертежи и схемы объектов проектирования хранятся в форматах *CDW, *DWG, *DXF и др. Однако, подавляющее большинство стеганографических алгоритмов, реализующих известные методы, ориентированы на работу с файлами следующих форматов: *.BMP, *.JPG, *.PNG, *.TIFF, *.GIF, *.AVI.

В рамках работы выполнены исследование двух алгоритмов стеганографии на основе дискретно-косинусного преобразования (ДКП) и их программная реализация. Разработано программное средство с графическим интерфейсом, позволяющее пользователю скрывать и извлекать информацию из изображений с использованием рассмотренных алгоритмов. Результаты тестирования разработанного ПО, а также их анализ представлены в данной статье.

1 Постановка задачи

Рассмотрен один из наиболее простых способов решения проблемы защиты расчетно-конструкторской документации (РКД) от несанкционированного доступа. Данный подход описывает следующая последовательность действий.

1) РКД искусственно делится на графическую и текстовую части.

2) Текстовая часть формируется как стегосообщение. В качестве контейнера для встраиваемой информации может выступать «нейтральное» изображение, либо изображение объекта проектирования.

Для решения поставленных задач проведен анализ алгоритмов стеганографии на основе ДКП. Преимуществом данного вида алгоритмов является возможность их применения для сокрытия данных как в видео, так и в аудиофайлах. Данный подход позволяет передавать РКД по открытым каналам. Известно, что эффективность защиты данных в алгоритме зависит от числа выбранных коэффициентов ДКП и пригодности блоков для встраивания информации. Поэтому для исследования были выбраны два алгоритма: Коча (*Koch*) и Бенхама (*Benham*). В алгоритме Коча используется два коэффициента ДКП, а в

алгоритме Бенхама – три. В алгоритме Бенхама также присутствует операция отсеечения непригодных для встраивания информации блоков.

2 Алгоритм Коча

В исследуемом алгоритме в блок размером 8×8 пикселей осуществляется встраивание одного бита стегосообщения. Два коэффициента ДКП выбираются случайным образом. Вычисление матрицы ДКП и последующее восстановление изображения (обратное ДКП) производятся по формулам:

$$\Omega(u, v) = \frac{\zeta(u) * \zeta(v)}{\sqrt{2N}} * \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} C(x, y) * \cos \left[\frac{\pi * u * (2x + 1)}{2N} \right] * \cos \left[\frac{\pi * v * (2y + 1)}{2N} \right]; \quad (1)$$

$$C_M(x, y) = \frac{1}{\sqrt{2N}} * \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} \zeta(u) * \zeta(v) * \Omega(u, v) * \cos \left[\frac{\pi * u * (2x + 1)}{2N} \right] * \cos \left[\frac{\pi * v * (2y + 1)}{2N} \right]. \quad (2)$$

Здесь $C(x, y)$ - элемент оригинального изображения размерностью $N \times N$, $C_M(x, y)$ - элемент восстановленного изображения размерностью $N \times N$, x и y – пространственные координаты пикселей изображения, $\Omega(u, v)$ - элемент матрицы коэффициентов ДКП, а u и v - координаты в частотной области. Коэффициент $\zeta(u)$ определяется следующим образом:

$$\zeta(u) = \begin{cases} \frac{1}{\sqrt{2N}}, & \text{если } u = 0, \\ 1, & \text{в противном случае.} \end{cases} \quad (3)$$

Встраивание информации осуществляется за счет изменения коэффициентов ДКП так, чтобы разность их абсолютных значений была больше некоторой положительной величины при передаче бита 0, а для передачи бита 1 — меньше некоторой отрицательной величины.

Таким образом, исходное изображение искажается за счет внесения изменений в коэффициенты ДКП. Степень искажения полученного изображения зависит от зашумленности исходного изображения — чем больше шумов оно содержит, тем менее заметны искажения при встраивании информации. Искажения больше всего заметны на монотонных изображениях (рис. 1).



а) исходное изображение

б) $P = 50$ (увеличено)

в) $P = 250$ (увеличено)

Рисунок 1 – Результаты встраивания в цветовой канал контейнера некоторого сообщения при различных значениях порога встраивания P

3 Алгоритм Бенхама

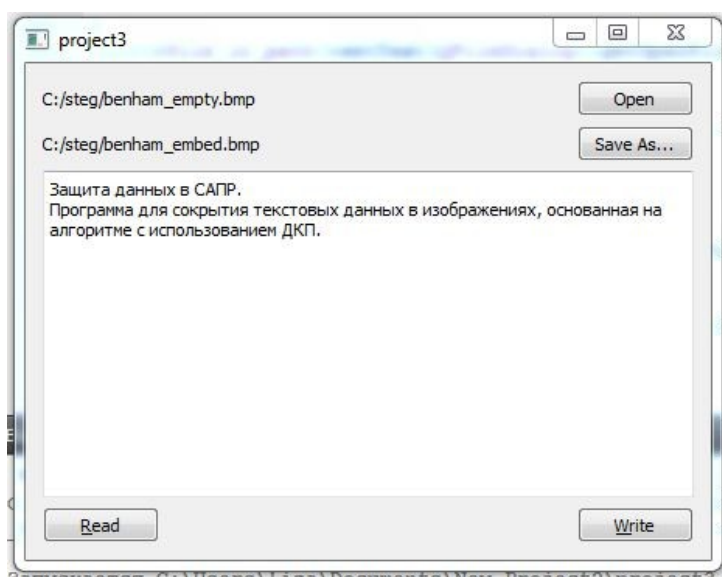
Алгоритм Бенхама является усовершенствованной версией предыдущего алгоритма. Для встраивания используются лишь «пригодные» блоки и внутри блока для встраивания выбираются не два, а три коэффициента, что существенно уменьшает визуальные искажения контейнера. Пригодными для встраивания информации считаются блоки изображения, не имеющие резких переходов яркости, а также не являющиеся слишком монотонными. Для первого типа блоков характерны очень большие значения нескольких низкочастотных коэффициентов, сопоставимых по своей величине с ДС-коэффициентом, для второго типа –

равенство нулю большинства высокочастотных коэффициентов. Эти особенности и являются критерием отсеивания непригодных блоков.

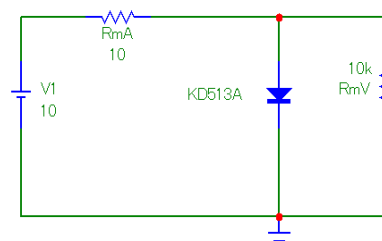
При встраивании бита цифрового водяного знака (ЦВЗ) выбираются три коэффициента (также случайным образом) ДКП блока. Если нужно передать бит 1, коэффициенты изменяются (если требуется) так, чтобы третий коэффициент стал меньше каждого из первых двух; если нужно встроить 0 он делается больше других. В том случае, если такая модификация приведет к слишком большой деградации изображения, коэффициенты не изменяют, и этот блок просто не используется. Изменение трех коэффициентов вместо двух, а тем более отказ от изменений в случае неприемлемых искажений уменьшает вносимые ЦВЗ погрешности. Декодер всегда сможет определить блоки, в которые ЦВЗ не встроен, повторив анализ, выполненный в кодере.

4 Описание программной реализации

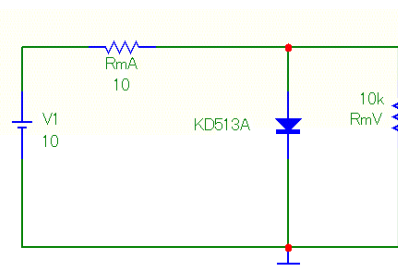
Программа создана на языке С++ в среде Qt Creator. На рис. 2 представлен графический интерфейс системы, позволяющий пользователю записать необходимую информацию в контейнер или считать уже закодированный текст.



а) исходное изображение



б) исходное изображение



в) модифицированное изображение

Рисунок 2 – Графический интерфейс результаты работы созданного программного средства

После загрузки рабочего изображения его необходимо сохранить под новым именем (“Save as...”). В текстовом окне вводится любой набор символов (русского или латинского алфавита) и при нажатии кнопки *Write* происходит встраивание сообщения в контейнер. Для восстановления информации необходимо открыть файл и нажать кнопку *Read*.

5 Сравнение результатов

На рис. 3 представлены результаты кодирования информации с использованием алгоритмов Бенхама и Коча. Алгоритм Бенхама, являющийся усовершенствованной версией алгоритма Коча, доказывает эффективность использования трех коэффициентов ДКП вместо двух и отсеивания «непригодных» блоков. При встраивании информации по методу Бенхама, искажения на изображениях менее заметны.

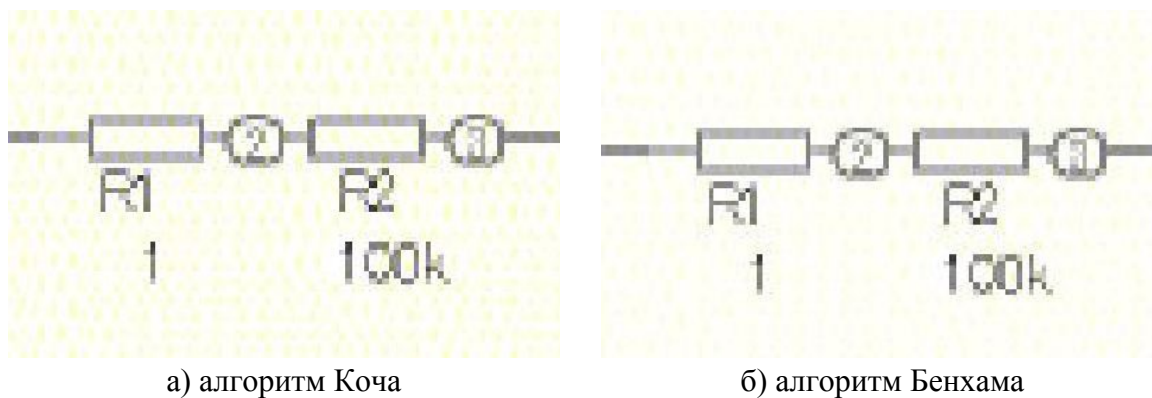


Рисунок 3 – Кодирование информации по алгоритмам Коча и Бенхама

Следует отметить, что стеганографические алгоритмы, такие как метод Бенхама и Коча, дают хорошие результаты при сокрытии данных в изображениях формата *BMP, *JPEG и др. Однако, непосредственное использование этих методов в проектной документации САПР приводит к порче файлов и потере данных (проект может быть восстановлен, но его структура будет нарушена). Для защиты данных в РКД необходимо использование других комбинированных методов, в которых документация разделяется на графическую и текстовую части.

Заключение

Проведенные исследования и полученные результаты позволяют сделать следующие выводы: стеганографические методы не являются надежными для защиты данных в САПР, поэтому необходимо работать над практической реализацией теоретически обоснованных комбинированных методов, а также искать новые и эффективные решения.

Литература

1. Конахович Г. Ф., Пузыренко А. Ю. Компьютерная стеганография. Теория и практика. — К.: МК-Пресс, 2006. — 126 с.
2. Грибунин В. Г., Оков И. Н., Туринцев И. В. Цифровая стеганография. — М.: Солон-Пресс, 2002.
3. Комбинированные методы защиты данных в САПР. - Волосатова Т.М., Денисов А.В., Чичварин Н.В. // «Информационные технологии», Приложение №5, 2012 – 32 с.: ил.
4. Защита проектной документации от несанкционированного доступа. - Волосатова Т.М., Денисов А.В., Чичварин Н.В. // 9 Международная конференция «Эффективные методы автоматизации подготовки и планирования производства», сборник трудов, с141-145. М.: 2012.
5. Метод сохранения данных с использованием искусственной дефокусировки. - Волосатова Т.М., Денисов А.В., Чичварин Н.В. //9 Международная конференция «Эффективные методы автоматизации подготовки и планирования производства», сборник трудов, с145-148. М.: 2012.
6. Защита данных в САПР: анализ стеганографических алгоритмов Коча (Koch) и Бенхама (Benham). – Круглая Е.И., Пилипенко А.В. // 3 Всероссийская научно-техническая конференция «Безопасные информационные технологии», сборник трудов, с.87-89. М.: 2012.

ПРИМЕНЕНИЕ ИМПУЛЬСНОГО ВОЛОКОННОГО ЛАЗЕРА ДЛЯ КОРРЕКТИРОВКИ МАСС РОТОРА

Кузнецов В.С.

*Научные руководители: к.т.н., Камалдинов А.М.,
д.т.н., профессор, Карпенко А.П.*

МГТУ им. Н.Э. Баумана, кафедра РК6, Москва, Россия

ON THE APPLICATION OF PULSED FIBER LASER FOR ROTOR MASS CORRECTION

Kuznetsov V.S.

Supervisors: Dr. Kamaldinov A.M., Prof., Karpenko A.P.

MSTU, Moscow, Russia

Аннотация

В работе рассмотрен вопрос применения импульсного волоконного лазера как средства корректировки масс ротора. Приведены описание экспериментальной установки, в которой был применен данный тип лазера, и некоторые результаты, полученные при ее эксплуатации. Уделено внимание геометрическим параметрам следа, формирующегося под действием лазерного луча. В заключении сделаны выводы о возможных применениях данного способа корректировки масс.

Annotation

The article discusses the possibility of using pulsed fiber laser as an instrument to provide rotor mass correction. The description of experimental set-up equipped with pulsed fiber laser and some operational data are presented. Some attention is paid to geometric parameters of the trace that have been formed on the surface of the rotor as a result of laser processing. In the conclusion possible applications of the method are discussed.

Введение

Балансировка роторов является одним из основных средств уменьшения вибраций машин и приборов. Процесс балансировки состоит из двух операций: определения значений и углов дисбалансов ротора и их уменьшения корректировкой масс [1].

В настоящее время распространены методы корректировки масс, использующие для удаления материала из тела ротора механическое воздействие (например, сверление). В сравнении с ними, лазерная эрозия имеет явное преимущество: корректировку масс этим методом можно осуществлять непосредственно в процессе вращения ротора. Кроме того, метод обладает большей точностью, а его применение открывает широкие возможности для автоматизации процесса балансировки. Известны технические решения [1, 2], в которых для корректировки масс вращающихся роторов используются преимущественно мощные лазеры, работающие в режимах свободной генерации и модуляции добротности.

В отдельных случаях под воздействием лазерного излучения на поверхности ротора формируется след конической, или каплевидной формы, длина которого зависит от радиуса ротора в плоскости коррекции, частоты его вращения и длительности импульсов лазера [1, 3]. Геометрические параметры следа при этом не всегда удовлетворяют техническим требованиям.

В упомянутых решениях за один импульс лазера, как правило, удаляется объем металла массой до нескольких мг. Часть металла в жидкой фазе выносится за пределы зоны взаимодействия, другая может застывать на поверхности ротора в виде облоя. На ротор оказывается термическое воздействие, которое может привести к нежелательным изменениям в структуре металла и к деформации ротора. При воздействии на ротор импульса лазера, работающего в режиме модуляции добротности, в области взаимодействия могут возникать процессы, оказывающие ударное воздействие на подшипники ротора.

Чтобы избежать негативных последствий, связанных с действием мощных импульсов лазера на ротор, нами была предпринята попытка использовать для корректировки масс ротора импульсный волоконный лазер [3].

1 Структура экспериментальной установки

На рисунке 1 приведена упрощенная схема экспериментальной установки для динамической балансировки тонкостенных роторов массой 20...200 г с собственным приводом и частотами вращения, достигающими 15000 мин^{-1} , разработанной коллективом отделения 260 ФГУП «НПО «Техномаш».

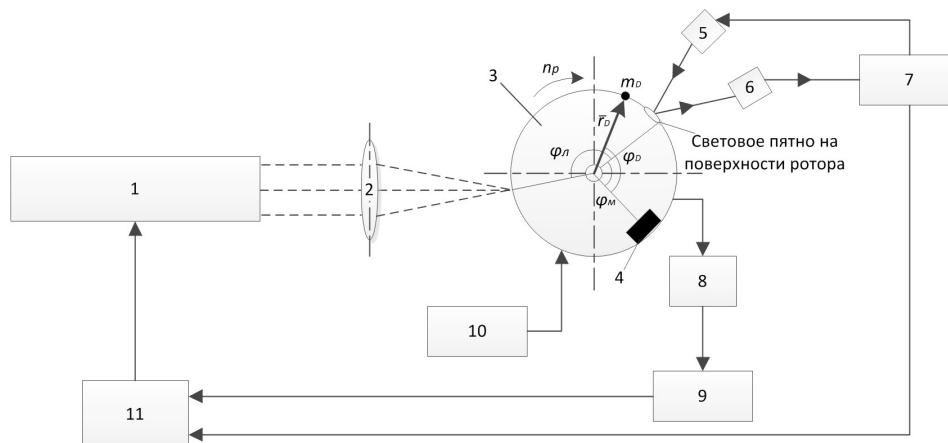


Рисунок 1 – Схема экспериментальной установки

На схеме использованы следующие обозначения: 1 – импульсный волоконный лазер, 2 – качающаяся оптическая система, 3 – балансируемый ротор, 4 – контрастная метка, нанесенная на поверхность ротора, 5 – источник света, 6 – фотоприемник, 7 – усилитель сигнала фотоприемника, одновременно служащий источником питания для источника света, 8 – вибродатчики, 9 – усилитель сигналов с вибродатчиков, 10 – источник питания собственного привода ротора, 11 – электронная аппаратура.

Одной из особенностей установки является наличие в ней качающейся оптической системы. Совершая колебательные движения вдоль линии, параллельной оси вращения ротора, оптическая система «разбрасывает» импульсы лазера по его поверхности.

Экспериментальная установка оснащена импульсным волоконным лазером со средней мощностью излучения 20 Вт (производитель лазера – НТО «ИРЭ Полюс»). Данная модель лазера построена по схеме со ступенчатым усилением [3], изображенной на рисунке 2.

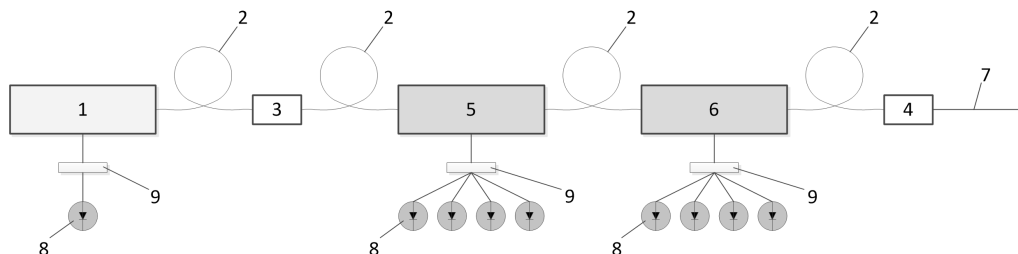


Рисунок 2 – К принципу построения импульсного волоконного лазера

На рисунке 2 обозначены: 1 – задающий лазер, 2 – соединительное оптоволокно, 3, 4 – оптические изоляторы, 5, 6 – оптоволоконные усилители первой и второй ступеней, соответственно, 7 – лазерный луч, 8 – светодиоды, 9 – фокусирующие системы светодиодов. Задающий лазер 1 играет роль генератора импульсов. Для их усиления используется двухступенчатая схема, включающая оптоволоконные усилители 5 и 6.

Существенно, что возможен режим работы импульсного волоконного лазера, при котором задающий лазер генерирует импульсы с заданными частотой и длительностью, а двухступенчатый усилитель включается только в определенные моменты времени. В таком режиме импульсы на выходе лазера излучаются лишь при включенном усилителе.

2 Принцип функционирования экспериментальной установки

При вращении ротора контрастная метка один раз за период обращения ротора проходит под источником света. Будем считать, что коэффициент отражения контрастной метки меньше коэффициента отражения материала, из которого изготовлен ротор. Тогда каждый раз при прохождении контрастной метки под световым пятном, на выходе фотоприемника будет устанавливаться низкий уровень сигнала. Сигнал с фотоприемника усиливается и подается на вход электронной аппаратуры. Электронная аппаратура определяет период вращения ротора T_p как временной интервал между соседними падениями уровня сигнала на фотоприемнике и рассчитывает частоту вращения ротора

$$n_p = \frac{1}{T_p} \cdot 60. \quad (1)$$

За начало периода принимается момент времени, когда на электронную аппаратуру приходит сигнал фотоприемника с низким уровнем.

Отсчитывая время τ_m от начала периода, можно определить угол между текущим положением контрастной метки и световым пятном источника света на поверхности ротора,

$$\varphi_m = \frac{\tau_m}{T_p} \cdot 360^\circ. \quad (2)$$

Преобразуя сигналы, получаемые с вибродатчиков и фотоприемника, электронная аппаратура определяет значения и углы дисбалансов ротора в полярной системе координат, связанной с осью вращения ротора. Угол дисбаланса φ_D отсчитывается от текущего положения контрастной метки против направления вращения ротора.

На рисунке 1 условно изображены точечная неуравновешенная масса m_D и радиус-вектор \vec{r}_D , определяющий положение этой массы относительно оси вращения, которые характеризуют дисбаланс в некоторой плоскости балансируемого ротора [1],

$$\vec{D} = m_D \cdot \vec{r}_D. \quad (3)$$

В полярной системе координат дисбаланс \vec{D} определяется парой значений $(m_D \cdot |\vec{r}_D|, \varphi_D)$. Для упрощения будем считать, что дисбаланс \vec{D} лежит в плоскости коррекции, а радиус ротора в этой плоскости $R_k = |\vec{r}_D|$.

При корректировке масс ротора электронная аппаратура в начале каждого периода генерирует задержку, длительность которой τ_z определяет выражение

$$\tau_z = \begin{cases} \frac{360^\circ - \varphi_n + \varphi_D}{360^\circ} \cdot T_p - \frac{\tau_n}{2}, & \text{если } \varphi_D < \varphi_n + \frac{\tau_n}{2} \cdot 360^\circ, \\ \frac{\varphi_D - \varphi_n}{360^\circ} \cdot T_p - \frac{\tau_n}{2}, & \text{если } \varphi_D \geq \varphi_n + \frac{\tau_n}{2} \cdot 360^\circ. \end{cases} \quad (4)$$

По истечению времени τ_z электронная аппаратура подает сигнал включения усилителя лазера и удерживает его в течение временного интервала τ_u .

В результате воздействия большого числа импульсов на поверхности ротора формируется след, длину которого можно оценить соотношением

$$l_c = \frac{2 \cdot \pi \cdot R_k \cdot n_p}{60} \cdot \tau_u. \quad (5)$$

3 Результаты эксперимента

Действие установки было опробовано на роторе массой около 66 г, выполненном из стали и вращающегося с частотой $n_p = 15000 \text{ мин}^{-1}$. Радиус ротора в плоскости коррекции $R_k \approx 17 \text{ мм}$. При корректировке масс ротора лазер работал в режиме с длительностью импульса 200 нс и частотой следования импульсов задающего лазера 20 кГц. Для интервала τ_u было установлено значение 45 мкс.

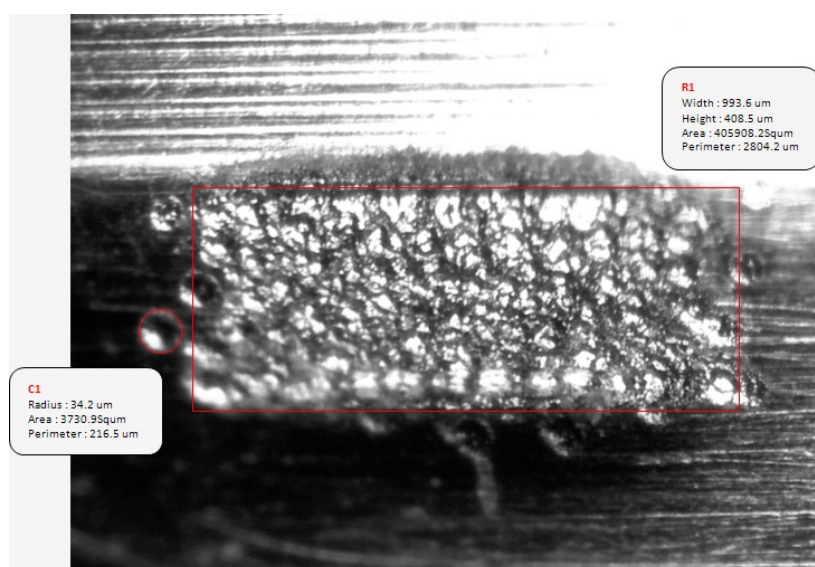


Рисунок 3 – След, образовавшийся на поверхности ротора под воздействием большого числа импульсов лазера

Под воздействием большого числа импульсов на поверхности ротора сформировался след, показанный на рисунке 3. Большая часть импульсов лазера попала в область, длина которой $l_c \approx 1 \text{ мм}$, а ширина $w_c \approx 0,4 \text{ мм}$. Полученное значение длины следа неплохо согласуется с оценкой $l_c \approx 1,2 \text{ мм}$, полученной из соотношения (5).

Заключение

Производительность процесса балансировки с использованием импульсного волоконного лазера для корректировки масс ротора невелика: на балансировку одного ротора приходится затрачивать несколько часов. Однако при этом дисбалансы ротора можно уменьшить до значений, не превышающих $10^{-4} \text{ г}\cdot\text{см}$.

Низкая производительность ограничивает область применения балансировочного оборудования, использующего для корректировки масс роторов импульсные волоконные лазеры. Однако в приборостроении, где важна точность балансировки, подобные решения могут найти применение при балансировке тонкостенных роторов, масса которых не превышает 200 г, либо для доводки дисбалансов роторов до предельно малых значений.

Литература

1. Справочник по балансировке/М.Е. Левит, Ю.А. Агафонов, Л.Д. Вайнгортин и др.; Под общ. ред. М.Е. Левита. – М.: Машиностроение, 1992. – 464 с.
2. LIA Handbook of Laser Materials Processing. Ed. Ready J.F. Orlando: LIA Magnolia Publishing Inc., 2001. 715 p.
3. Григорьянц А.Г., Шиганов И.Н., Мисюров А.И. Технологические процессы лазерной обработки: Учеб. пособие для вузов / Под ред. А.Г. Григорьянца. – 2-е изд., стереотип. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2008. - 664 с.

ВОЗМОЖНОСТИ И ПРЕИМУЩЕСТВА СИНХРОННОЙ ТЕХНОЛОГИИ ТРЕХМЕРНОГО МОДЕЛИРОВАНИЯ

Голиков С.А., Капитонов К.Д., Кошман А.А.
Научный руководитель: к.т.н., доцент, Маничев В.Б.
МГТУ им. Н.Э.Баумана, кафедра РК6, Москва, Россия

FEATURES AND ADVANTAGES OF THREE-DIMENSIONAL MODELING SYNCHRONOUS TECHNOLOGY

Golikov S.A., Kapitonov K.D., Koshman A.A.
Supervisor: Ph.D., Associate professor, Manichev V.B.
BMSTU, RK6 Department, Moscow, Russia

Аннотация

В статье рассматривается синхронная технология трехмерного моделирования. Кратко представлена история развития данного метода. Приводятся основные возможности синхронной технологии. В заключении представлены преимущества синхронной технологии перед другими методами трехмерного моделирования.

Abstract

In this article we observed synchronous technology of three-dimensional modeling. The evolution of the method was briefly provided. The main features of the synchronous technology were also given. As a conclusion we provided the advantages of synchronous technology over other methods of three-dimensional modeling.

Введение

Синхронная технология — метод трехмерного параметрического моделирования, представленный компанией Siemens PLM Software в 2008 г. Данный метод проектирования реализован в CAD-системе среднего уровня Solid Edge. Синхронная технология совмещает средства параметрического моделирования на основе конструктивных элементов с возможностями прямого редактирования элементов геометрической формы. Синхронная технология возможна благодаря использованию вариационного подхода к проектированию изделия, состоящего в связывании граничных элементов геометрической модели логическими и параметрическими ограничениями (как заданными пользователем, так и автоматически распознанными системой CAD).

1 История создания

Синхронная технология является развитием так называемого метода прямого проектирования, который был представлен в предыдущих версиях продуктов компании Siemens. Прямое моделирование позволяло быстро и гибко изменять части модели, поскольку ее пересчет происходил только в месте редактирования. Но прямое моделирование все равно основывалось на дереве построений и могло привести к изменению или разрушению задумки проектировщика, так как ранее созданные элементы также могли быть изменены при редактировании методом прямого моделирования.

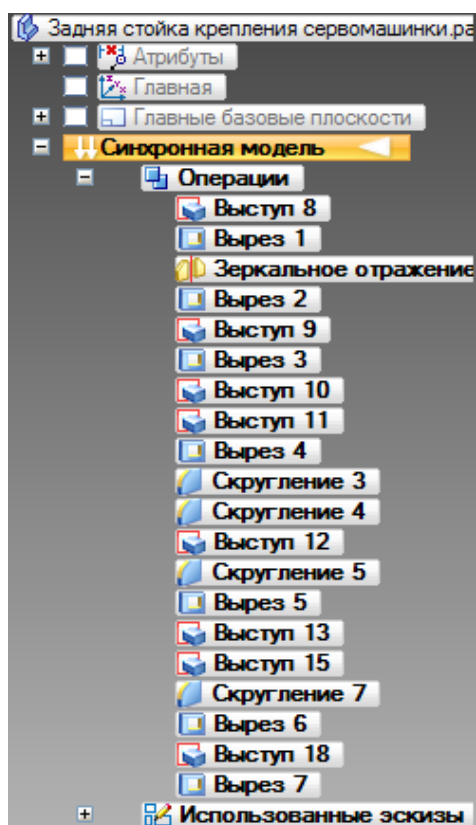


Рисунок 1 - История создания модели

Синхронная технология в свою очередь была избавлена от этого за счет заданных размеров модели и геометрических и конструктивных взаимосвязей. История создания модели ведется, но созданные элементы, при их возможном изменении, уже не влияют друг на друга.

2 Возможности синхронной технологии

Синхронная технология реализует поэлементное моделирование детали без дерева построения. Это позволяет изменять размеры, правила проектирования и какие-либо параметры, создавая или редактируя модель, при этом не отталкиваясь от того как она была создана. Данная технология позволяет менять параметры отверстий, скруглений, массивов без наложения взаимосвязей на отдельные элементы модели. Элементы детали можно изменять, не опираясь на то, когда они были созданы. Например, для удаления какого-либо отверстия или изменения положения плоскости пользователю не обязательно удалять все шаги которые были сделаны после построения этих элементов, он просто может удалить элемент из истории построения (если на него не наложены какие-либо ограничения) или динамически изменить какую-то часть модели.

Для большего контроля при изменении каких-либо размеров введены опции их смещения в ту или иную сторону. При этом данная технология динамически показывает как изменяется тот или иной конструктивный элемент при его редактировании, то есть пользователь может видеть к чему придет его изменение, и как оно затронет остальные части модели. Также при невозможности изменения геометрии из-за имеющихся ограничений, пользователь будет проинформирован программой об этом.

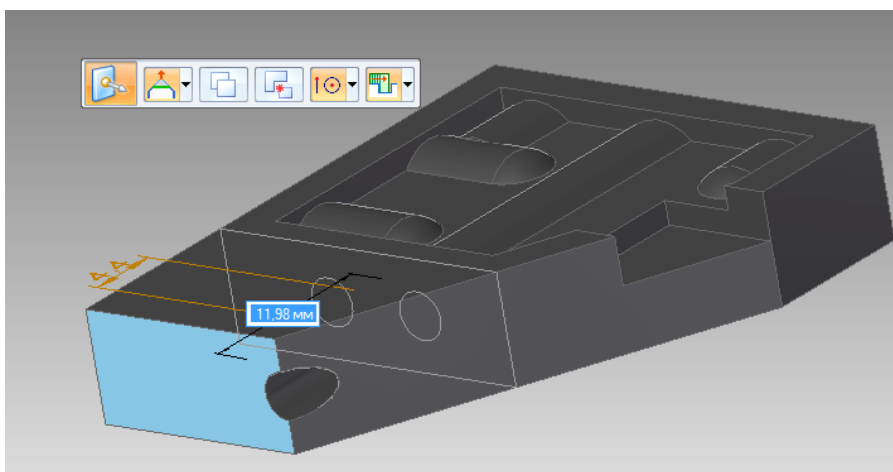


Рисунок 2 - Пример динамического изменения элементов модели

Если пользователю необходимо, чтобы какие-либо части модели не изменялись при редактировании других, то для этого существует набор правил, с помощью которых определяется вид возможных изменений тех или иных элементов. Отключая, или наоборот включая, те или иные зависимости пользователь может влиять на изменения геометрии какой-либо части модели.

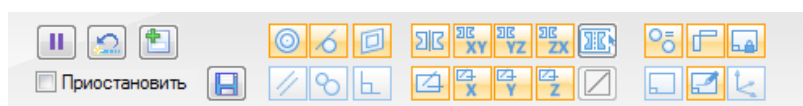


Рисунок 3 - Настройка текущих правил при изменении элементов модели

Стоит отметить, что использование синхронной технологии позволяет создавать различные варианты деталей с похожей геометрией путем прямого изменения исходного изделия, а также удобно исправлять деталь по месту в сборке, что является одним из ключевых моментов при проектировании сборок с большим числом деталей.

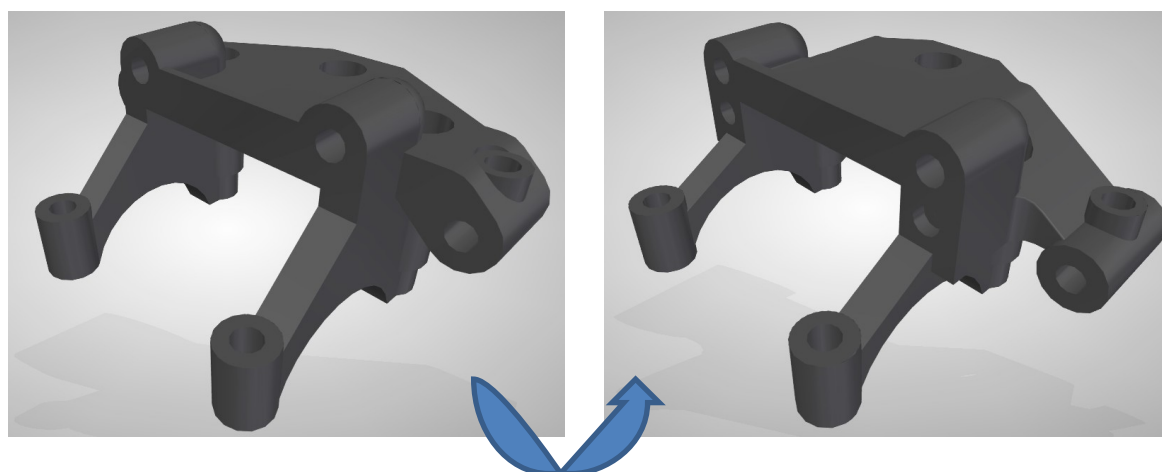


Рисунок 4 - Пример создания передней стойки крепления амортизаторов путем прямого изменения существующей задней стойки

Все это дает проектировщику возможность быстро найти оптимальное решение при возникновении различных проблем, связанных с взаимным расположением деталей в сборке, а также их возможным пересечением.

Заключение

Синхронная технология — это новая ветвь развития моделирования, которая меняет не только методику моделирования деталей, но и весь образ мышления при создании модели. Это не просто прямое редактирование, а объединение свободы конструирования с параметрической точностью систем с деревом построения.

При синхронном проектировании пользователь имеет полный контроль над параметрами и элементами импортированной, а затем сохраненной модели. Благодаря этому он может вернуться к ее редактированию в любое время и менять элементы в любом порядке. Поэтому одним из преимуществ новинки является упрощение работы с моделями, полученными из других САПР.

Таким образом, синхронная технология направлена на сокращение времени и трудоемкости процесса проектирования и, как следствие, на повышение производительности труда проектировщиков, а также на многократное использование моделей, созданных в разных САПР.

Литература

1. Хохленков Р.В. Solid Edge с синхронной технологией. М: ДМК Пресс, 2010. 372 с.
2. Брук Р. Переход к Solid Edge с синхронной технологией // Ж. САПР и графика. 2009. 5. С. 4–6. [HTML] (<http://www.sapr.ru/Article.aspx?id=20367>).
3. Свобода с синхронной технологией // Ж. САПР и графика. 2010. 8. С. 58–59. [HTML] (<http://www.sapr.ru/article.aspx?id=21586&iid=985>).

МОДИФИКАЦИЯ АЛГОРИТМОВ КОМПЬЮТЕРНОГО ЗРЕНИЯ АНАЛИЗА ДИНАМИЧЕСКИХ ОБЪЕКТОВ ДЛЯ СЛУЧАЯ ПЕРЕМЕННОГО ОСВЕЩЕНИЯ

Марченков А.М.

Научный руководитель к.т.н., доцент Волосатова Т.М.

МГТУ им. Н.Э. Баумана, кафедра РК6, Москва, Россия

MODIFICATION OF COMPUTER VISION ALGORITHMS FOR DYNAMIC OBJECTS ANALYSIS IN CASE OF VARIABLE LIGHTING

Marchenkov A.M.

The scientific chief: Ph.D., associate professor Volosatova T.M.

Department RK6 BMSTU, Moscow, Russia

Аннотация

Приведен краткий обзор методов распознавания динамических объектов. Показана инвариантность голографических методов к переменной освещенности. Предложен вариант модификации существующих алгоритмов слежения.

Abstract

A brief review of methods for identification of dynamic objects is given. The invariance of the holographic methods for variable light conditions is shown. A variant of the existing tracking algorithm modification is suggested.

Введение

Под анализом динамических объектов в системе технического зрения понимается анализ перемещения объекта в пределах сцены, определение направления движения и поворота.

В настоящее время эта область компьютерного зрения является одной из наиболее популярных и динамически развивающихся. Одним из направлений развития является модификация существующих алгоритмов для уменьшения вычислительных затрат, например, анализ возможности встраивания алгоритмов непосредственно в камеру.

Традиционные алгоритмы имеют высокую вычислительную сложность, а ресурсы однокристалльной платформы камеры (набор процессорных команд, тактовая частота и размер памяти) лимитированы из-за ограничений по тепловыделению и стоимости. Таким образом, необходима разработка принципиально новых алгоритмов с большей вычислительной эффективностью [1].

Для определения перемещения объекта в пределах сцены обычно анализируется не вся сцена, а только особые точки. Особая точка сцены или точечная особенность – это такая точка сцены, изображение которой можно отличить от изображений всех соседних с ней точек сцены.

Модификация алгоритмов компьютерного зрения анализа динамических объектов для случая переменного освещения

Большинство современных алгоритмов слежения за особенностями опираются на работы Лукаса и Канаде. В трекаре Лукас-Канаде особенность считается только смещающейся, без искажений. Этот алгоритм в принципе применим для функций любой размерности n . Пусть x - особенность первой функции F , необходимо найти такую точку $x + h$ функции G , что разность окрестностей этих точек по мере - минимальна.

Расстояние между окрестностями записывается в виде:

$$E = \sum_{x \in R} [F(x + h) - G(x)],$$

где $F(x)$, $G(x)$ – две функции.

Функцию $F(x)$ с помощью разложения в ряд Тейлора можно приближенно представить в виде:

$$F(x+h) = F(x) + h \frac{\partial F(x)}{\partial x},$$

где $\frac{\partial}{\partial x} = [\frac{\partial}{\partial x_1} \frac{\partial}{\partial x_2} \dots \frac{\partial}{\partial x_n}]^T$ – градиент.

Используя это приближение, ищется минимум E путем дифференцирования и приравнивания производной к нулю:

$$0 = \frac{\partial}{\partial h} E \approx \frac{\partial}{\partial h} \sum_x [F(x) + h \frac{\partial F}{\partial x} - G(x)]^2 = \sum_x 2 \frac{\partial F}{\partial x} [F(x) + h \frac{\partial F}{\partial x} - G(x)]$$

Отсюда смещение h можно получить как

$$h = \left[\sum_x \left(\frac{\partial F}{\partial x} \right)^T [G(x)] - F(x) \right] \left[\sum_x \left(\frac{\partial F}{\partial x} \right)^T \left(\frac{\partial F}{\partial x} \right) \right]^{-1}$$

Как было указано ранее, задача слежения за особенностями без учета аффинных искажений является поиском величины освещенности в наборе точек. Поэтому метод Лукас-Канаде часто применяется для поиска величины освещенности во всем изображении.

Развитием этой работы являются следующие вариации трекеров:

- Томаси-Канаде – переформулирование Лукас-Канаде. Движение считается смещением, и рассчитывается путем итеративного решения построенной системы линейных уравнений;
- Ши-Томаси-Канаде - учитывает аффинные искажения особенности;
- Джин-Фаваро-Соатто - модификация Ши-Томаси-Канаде с учетом аффинных изменений освещенности особенности.

Усложнение задачи слежения ведет к увеличению размерности системы линейных уравнений, к решению которой сводится эта задача. Вследствие этого увеличивается вычислительная сложность задачи. Подробнее задачи поиска точечных особенностей и варианты трекеров рассмотрены в работе [2].

Для более простых случаев, например, обнаружение объектов движущихся на постоянном фоне (лента конвейера), достаточно применить метод вычитания в совокупности с вычислением векторов перемещений. Более подробно они описаны в [3].

Одним из вариантов уменьшения сложности задачи слежения является применение голографических методов.

Экспериментальные исследования

Чтобы определить, насколько голографические методы чувствительны к освещенности, получим цифровые голограммы от пяти изображений с интенсивностями объекта $I_1 = 255, I_2 = 170, I_3 = 100, I_4 = 60, I_5 = 30$. Тестовые изображения показаны на рисунке 1.



Рисунок 1 - Тестовые изображения с интенсивностями объекта $I_1 = 255, I_2 = 170, I_3 = 100, I_4 = 60, I_5 = 30$ соответственно

На рисунке 2 приведены линии равного уровня голограмм, синтезированных для тестовых изображений. Для определения качества синтезированных голограмм проведем процедуру восстановления исходных изображений, ее результаты показаны на рисунке 3. Появление засвеченной области на восстановленном изображении связано с ошибками дискретного преобразования Фурье на конечном интервале значений.

Из результатов эксперимента видно, что полученные голограммы схожи между собой. От интенсивности пикселей объекта зависит только значение амплитуды голограммы,

которая изменяется пропорционально с яркостью пикселей. При этом фаза голограммы практически не изменяется.

Теперь проведем эксперимент на изображениях, на которых объект имеет не однородную интенсивность. Возьмем три тестовых изображения с различной однородностью интенсивностей пикселей изображения, они представлены на рисунке 4. Все три изображения имеют различную неоднородность яркостей пикселей. На первом изображении границы объектов и их основная часть имеют разные интенсивности ($I_{\text{тела}} = 195, I_{\text{границы}} = 125$), аналогично на втором изображении, но помимо этого яркость границы буквы «С» меньше, чем буквы «А» ($I_{\text{тела}} = 195, I_{\text{границы А}} = 125, I_{\text{границы С}} = 100$). На третьем изображении неоднородность наиболее ярко выражена, помимо описанных выше, некоторая часть тела объекта «А» имеет интенсивность более низкую интенсивность ($I_{\text{части объекта А}} = 80$).

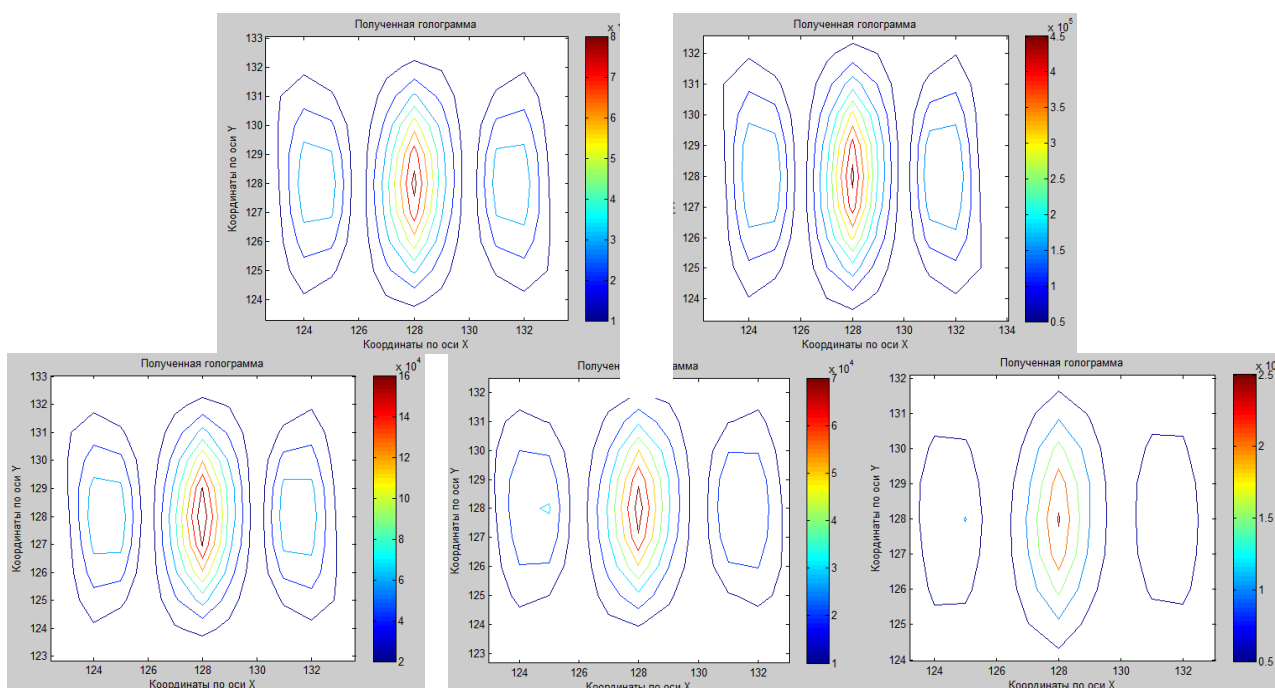


Рисунок 2 - Линии уровня для изображений с интенсивностями

$I_1 = 255, I_2 = 170, I_3 = 100, I_4 = 60, I_5 = 30$ соответственно

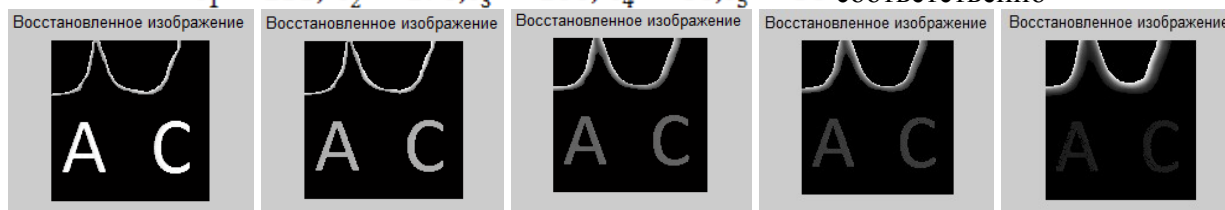


Рисунок 3 - Восстановленные изображения по голограммам тестовых изображений с интенсивностями $I_1 = 255, I_2 = 170, I_3 = 100, I_4 = 60, I_5 = 30$ соответственно

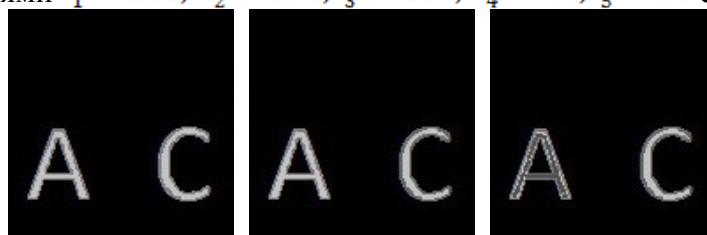


Рисунок 4 - Тестовые изображения с неоднородными интенсивностями пикселей объекта

На рисунках 5 и 6 приведены линии уровня полученных цифровых голограмм и восстановленные изображения соответственно.

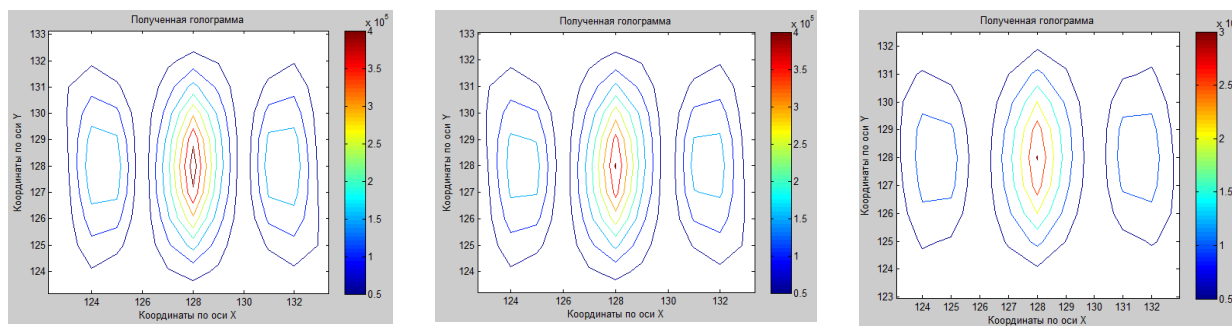


Рисунок 5 - Линии уровня цифровых голограмм от тестовых изображений с неоднородными интенсивностями пикселей объекта

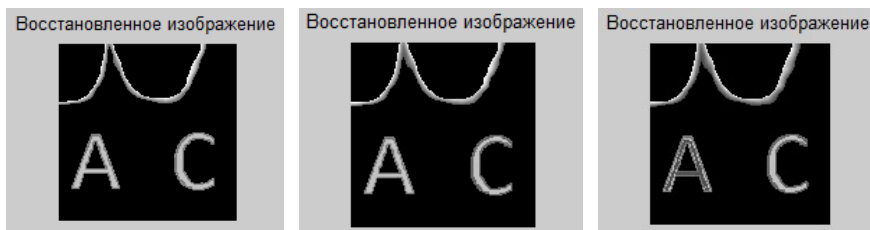


Рисунок 6 - Восстановленные изображения по голограммам тестовых изображений с неоднородными интенсивностями пикселей объекта

Результаты эксперимента с неоднородными изображениями полностью совпадают с полученными ранее. Из этого можно сделать вывод, что голографические методы не чувствительны к изменению освещенности объекта и на них не влияет неоднородность освещения объекта.

Заключение

Использование голографических методов совместно с трекарами позволяет уменьшить размерность решаемой системы линейной уровней, за счет того, что в ней не нет необходимости учитывать переменную освещенность сцены. Таким образом ресурсоемкость решаемой задачи уменьшится при сохранении исходных требований к алгоритмы.

Литература

1. Птицын Н. Встроенная видеоаналитика для детектирования и сопровождения объектов при помощи многомасштабных признаков [Электронный ресурс] / Птицын Н. – Режим доступа: <http://synesis.ru/ru/surveillance/articles/multiscale-va> (21.01.2013)
2. Конушин. А. Слежение за точечными особенностями сцены [Электронный ресурс] // Компьютерная графика и мультимедиа. –2003-№1 - Режим доступа: <http://cgm.computergraphics.ru/content/view/54> (21.01.2013)
3. Марченков А.М. Анализ методов распознавания динамических объектов. Сборник трудов 14 молодежной международной научно-технической конференции «Наукоемкие технологии и интеллектуальные системы», с. 122-125.

РЕАЛИЗАЦИЯ КАНОНИЧЕСКОГО МЕТОДА ОПТИМИЗАЦИИ РОЕМ ЧАСТИЦ В ПРОГРАММЕ МОЛЕКУЛЯРНОГО ДОКИНГА

Матвеева К.О.

Научный руководитель: д.т.н., профессор, Карпенко А.П.

МГТУ им. Н.Э.Баумана, кафедра РК-6, Москва, Россия

PARTICLE SWARM OPTIMISATION IN DOCKING PROGRAM

Matveeva K.O.

Supervisor: Dr., Prof., Karpenko A.P.

MSTU n.a. Bauman, Moscow, Russia

Аннотация

В данной работе описан канонический метод роя частиц и его модификация в применении к программе жесткого молекулярного докинга. Приведены результаты экспериментов, которые показывают, что канонический метод роя частиц неэффективен для рассматриваемой задачи из-за высокой сложности, однако его модификация может значительно улучшить результат работы программ докинга.

Abstract

In this paper is described a canonical method of a swarm of particles and its modification in the application to the hard molecular docking program. Presents the results of experiments that show that the canonical method of a swarm of particles is inefficient for the solving this task due to the high complexity, though, the modification of a swarm of particles method can significantly improve the result.

Введение

Молекулярный докинг (или молекулярная стыковка) – это метод молекулярного моделирования, который позволяет предсказать наиболее выгодную для образования устойчивого комплекса ориентацию и положение одной молекулы по отношению к другой. Молекулярный докинг позволяет моделировать взаимодействие между большей молекулой – рецептором и меньшей молекулой – лигандом. Молекулярный докинг – всегда вычислительно сложная процедура, так как существует большое число вариантов расположения двух молекул относительно друг друга, и с ростом количества атомов в молекулах количество вариантов расположения, которое должно быть проанализировано, растет экспоненциально [1].

Одной из наиболее важных задач для молекулярного моделирования является определение возможных конфигураций супермолекул, а также оценка устойчивости соответствующих структур [2]. Знания о предсказанной ориентации могут быть использованы для предсказания прочности комплекса или средства связей между двумя молекулами.

Существует много программ для моделирования стыковки белков и оценки устойчивости супермолекулы. Большинство из этих программ работает по следующему принципу: один белок фиксируется в пространстве, а второй поворачивается вокруг него разнообразными способами. При этом для каждой конфигурации поворотов производятся расчеты оценочной функции. Оценочная функция основана на электростатических взаимодействиях, Ван-дер-Ваальсовском отталкивании и так далее. Проблема при этом заключается в том, что вычисления по всему конфигурационному пространству требуют много времени, редко приводя к единственному решению.

1 Постановка задачи докинга

С математической точки зрения задача докинга состоит из трех подзадач [3]. Во-первых, необходимо определить, какими степенями свободы будут наделены лиганд и белок-мишень. Во-вторых, необходимо определить алгоритм поиска в этом многомерном

пространстве наилучшего способа связывания лиганда с белком-мишенью. В-третьих, необходимо составить скоринг-функцию, которая определяет энергию связывания лиганда с белком-мишенью и которая служит объектом глобальной оптимизации для алгоритма поиска. В данной работе рассматривается одно из возможных решений второй подзадачи – решение на основе алгоритма поиска методом роя частиц и его модификации.

2 Канонический метод роя частиц

Алгоритм оптимизации роем частиц (*Particle Swarm Optimization, PSO*) является эвристическим поведенческим алгоритмом безусловной оптимизации. Впервые был предложен для описания поведения стаи птиц либо других скоплений живой природы, образующих роевой интеллект. Впоследствии изучение метода показало его пригодность к решению задач оптимизации, в которых вектор решения стали называть «частицей». Метод широко развивался и модифицировался и нашел применения во многих инженерных, экономических, информационных и других науках. Канонический метод роя частиц представлен в работе [4].

Рассматриваем задачу глобальной минимизации целевой функции в D -мерном арифметическом пространстве \mathbf{R}^D

$$\min_{X \in \mathbf{R}^D} F(X) = F(X^*) = F^* . \quad (1)$$

Множество векторов решений называется множеством частиц. Каждая частица имеет вектор начальных случайных положений в пространстве поиска и вектор начальных случайных скоростей

$$X_{i,0} = X^0, |X_{i,t}| = D , \quad (2)$$

$$V_{i,0} = V^0, |V_{i,t}| = D , \quad (3)$$

где t – номер итерации; n – число частиц в рое, $i \in [1: n]$.

Итерационные формулы метода роя частиц имеют вид

$$X_{i,t+1} = X_{i,t} + V_{i,t+1} , \quad (4)$$

$$V_{i,t+1} = \omega V_{i,t} + \alpha R_{1,t} \otimes (X_{i,t}^b - X_{i,t}) + \beta R_{2,t} \otimes (X_{i,t}^g - X_{i,t}) . \quad (5)$$

Здесь $X_{i,t}^b$ – вектор лучшего решения, найденного частицей i за всё время поиска; $X_{i,t}^g$ – вектор лучшего из решений, найденных за всё время поиска соседними с i -й частицами, включая частицу i ; $R_{1,t}, R_{2,t}$ – D -мерные векторы псевдослучайных чисел в интервале $[0;1]$; ω, α, β – весовые коэффициенты, свободные параметры алгоритма.

Множество частиц может иметь различную топологию соседства, от которой в значительной мере зависят скорость сходимости алгоритма и вероятность локализации глобального экстремума. Так, например, топология полного графа обеспечивает наибольшую скорость сходимости по сравнению с другими топологиями, но склонна к локализации локального экстремума.

Схема метода имеет следующий вид.

- 1) Инициализируем рой частиц, задаем значения свободных параметров, полагаем счетчик числа итераций .
- 2) Находим лучшее решение для каждой частицы, обновляем вектора $X_{i,t}^b$.
- 3) Находим лучшее решение среди всех соседних с i -й частиц (при полной связности – поиск лучшего решения среди всех частиц) $\forall i \in [1: n]$, обновляем вектора $X_{i,t}^g$.
- 4) Вычисляем $X_{i,t+1}, V_{i,t+1} \forall i \in [1: n]$ по формулам (1), (2).
- 5) Проверяем выполнение условия окончания итераций. Если это условие выполнено, то завершаем итерации, в противном случае полагаем $t = t + 1$ и переходим к шагу 2.

3 Модификация канонического метода роя частиц

Положим, что набор соседей, по которому вычисляется изменение скорости частицы, может включать в себя более двух соседей. Тогда, для случая четырех наборов соседей итерационные формулы будут выглядеть следующим образом:

$$X_{i,t+1} = X_{i,t} + V_{i,t+1}, \quad (6)$$

$$V_{i,t+1} = \omega V_{i,t} + \alpha R_{1,t} \otimes (X_{i,t}^b - X_{i,t}) + \beta R_{2,t} \otimes (X_{i,t}^g - X_{i,t}) + \gamma R_{3,t} \otimes (X_{i,t}^p - X_{i,t}) + \delta R_{4,t} \otimes (X_{i,t}^q - X_{i,t}). \quad (7)$$

Здесь $R_{1,t}, R_{2,t}, R_{3,t}, R_{4,t}$ – D -мерные вектора псевдослучайных чисел в интервале $[0;1]$; $\omega, \alpha, \beta, \gamma, \delta$ – весовые коэффициенты, свободные параметры алгоритма.

Положим, что в третий и четвертый набор соседей частицы выбираются случайно.

4 Реализация и эксперимент

Программа докинга была реализована так, что меняя параметры, можно запускать программу докинга, как с базовым методом роя частиц, так и с модифицированным. В работе рассматривается комплекс *3U7S* (протеаза *HIV1* и дарунавир в качестве лиганда), размерность вектора решений $D = 28$.

В первом эксперименте для роя в 32 частицы с двумя наборами соседей (канонический метод роя частиц) алгоритм либо не сходится, либо сходится чрезвычайно медленно. Полученная на одной из итераций структура молекулярного комплекса представлена на рисунке 1.

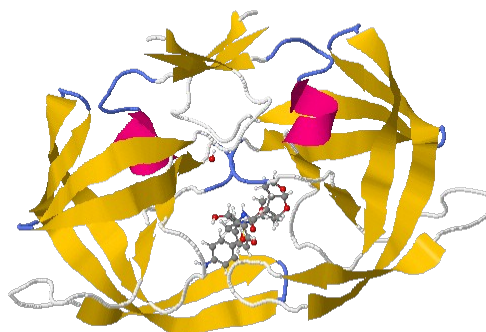


Рисунок 1 – Пример молекулярного комплекса, полученного каноническим методом роя частиц

Во втором эксперименте для роя в 32 частицы с четырьмя наборами соседей алгоритм сошелся за 3939 секунд. Значение энергии лучшей частицы: 206,03 ккал/моль. Среднеквадратичное отклонение от нативного положения лиганда составляет 7,59 ангстрем. Программа завершила работу при дисперсии центра координат лиганда, равной 0,000969 ангстрем. Полученная структура представлена на рисунке 2.

В третьем эксперименте для роя из 1024 частиц с шестью наборами соседей алгоритм сошелся за 428 итерации. Значение энергии лучшей частицы равно 70,642967 ккал/моль, среднеквадратичное отклонение от нативного положения лиганда - 5,88 ангстрем, дисперсия центра координат лиганда - 0,000947 ангстрем, время вычислений - 4747 секунд. Полученная структура представлена на рисунке 3.

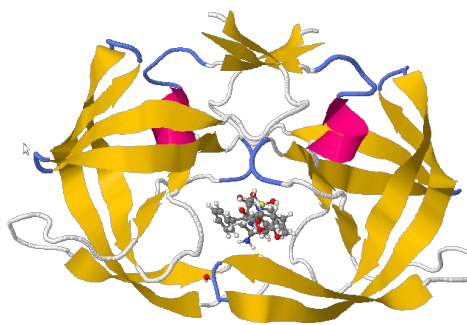


Рисунок 2 – Молекулярный комплекс, полученный в результате второго эксперимента

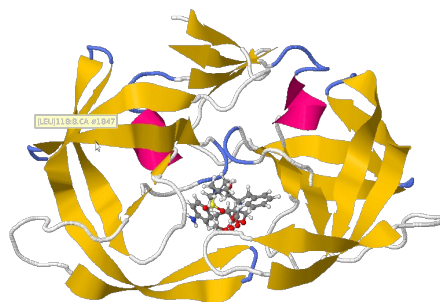


Рисунок 3 – Молекулярный комплекс: эксперимент 3

Заключение

В работе показано, что канонический метод роя частиц в задаче докинга не дает удовлетворительных результатов. Эксперименты с применением модифицированного метода роя частиц показали, что последний может быть эффективно применен для задачи молекулярного докинга.

Литература

1. Brian K. Shoichet, Dale L. Bodian* and Irwin D. Kuntzt, *Molecular Docking Using Shape Descriptors*. // *Journal of Computational Chemistry*, 1992, Vol. 13, No.3, p. 380-397.
2. Григорьев, Ф.В, *Методы молекулярного моделирования супрамолекулярных комплексов: иерархический подход*. // *Российские нанотехнологии*, 2009, Т. 5, с. 47-53.
3. Антух А. Э., Исследование канонического метода роя частиц (PSO) для топологий типа «клика» и «кластер». // *Инженерное образование*, № 06, июнь 2009. <http://technomag.edu.ru/doc/127975.html>
4. Романов А.Н., Кондакова О.А., Григорьев Ф.В., Сулимов А.В., Луцкекина С.В., Мартынов Я.Б., Сулимов В.Б, *Компьютерный дизайн лекарственных средств: программа докинга SOL // Вычислительные методы и программирование*, 2008, Т. 9, с. 213-233.

МЕТОД ОБНАРУЖЕНИЯ АНОМАЛИЙ В ИСХОДНЫХ ДАННЫХ СИСТЕМ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ

Орлов А. О.

Научный руководитель: к.т.н., доцент, Кузовлев Вячеслав Иванович
МГТУ им. Н.Э.Баумана, кафедра ИУ-5, Москва, Россия

THE APPROACH FOR ANOMALY DETECTION IN DECISION SUPPORT SYSTEMS INPUT DATA

Orlov A. O.

Supervisor: Ph.D., Assoc. Prof., Kuzovlev V. I.
MSTU, Moscow, Russia

Аннотация

В статье рассматривается метод обнаружения аномалий в данных систем поддержки принятия решений, основанный на расчете расстояний между объектами данных. Описана формула расчета расстояний между категориальными данными, выраженными в вербальной форме. Приведено сравнение предложенной формулы с существующими, описаны преимущества, состоящие в повышении быстродействия при расчете расстояний. В заключении описываются направления дальнейших исследований.

Abstract

The method of anomaly detection in decision support systems input data is considered. The method is based on data objects distance calculation. The formula for distance between categorical data objects is given. The advantages of suggested formula are also described comparing to existing methods. The main advantage is an increase of distance calculation performance. In conclusion some further research in the anomaly detection area is described.

Введение

Важной особенностью систем поддержки принятия решений (СППР) является способность к прогнозному анализу данных. Суть прогнозного анализа заключается в создании на основе входных данных прогнозной модели, которая далее используется для формирования прогнозов. Под прогнозом понимается суждение о будущих фактах на основе имеющихся статистических данных. Точность прогноза зависит от качества исходных данных, по которым формируется прогнозная модель. Наличие искажений (или шума) в данных оказывает влияние на результат бизнес-процесса и процесса в СППР: происходит анализ искаженных данных, в результате могут вырабатываться неверные и неэффективные решения и организационные воздействия. Для решения данной проблемы при построении прогнозной модели необходимо использовать механизм, способный обрабатывать искаженные данные таким образом, чтобы они оказывали минимальное воздействие на результат работы системы. В данной работе рассматривается такой тип искажений, как аномалии в данных. В отличие от искажений, вызванных отсутствием данных, аномалии требуют дополнительных процедур поиска и идентификации. Аномальные данные могут являться как следствием возникновения сбоев или ошибок случайного характера и не нести физического смысла, так и обладать физическим смыслом, то есть не являться в общем смысле ошибочными. Однако, та часть данных, которая не удовлетворяет некоторым общим закономерностям, считается аномалией, поскольку не может являться достаточным основанием, на котором аналитическая модель создаст прогноз. На рисунке 1 показаны объекты в некотором двумерном пространстве. Здесь аномалиями будут считаться объекты o_1 , o_2 , а также объекты в множестве O_3 , поскольку все они достаточно удалены от областей N_1 и N_2 , содержащих большинство объектов данных.

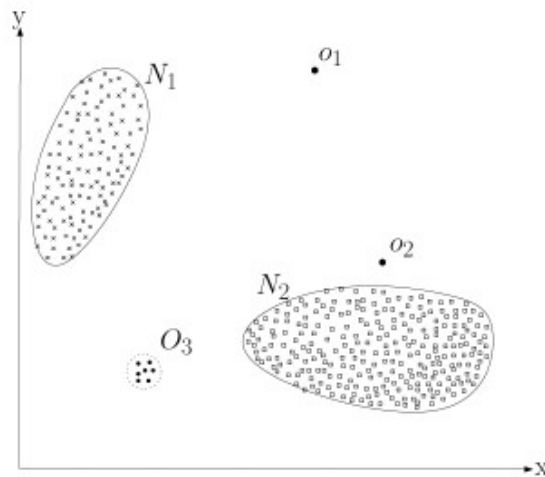


Рисунок 1 – Понятие аномалий в данных

В [4] приведен широкий обзор существующих подходов к решению проблемы обнаружения аномалий. Авторы разбивают существующие методы на несколько категорий по общему характеру. Методы каждой из категорий имеют свои достоинства и недостатки и должны выбираться, по мнению авторов, в зависимости от специфики предметной области каждой отдельно взятой задачи. Создание системы поддержки принятия решений и, в частности, прогнозной модели анализа данных начинается с обучения модели на исходных данных, поэтому рассматривались методы, способные работать на этапе обучения модели. А именно методы, основанные на широко известном подходе «к ближайших соседей», когда объекты анализируются совместно с другими объектами, ближайшими к ним.

В данной работе предлагается метод обнаружения аномалий в данных, основанный на методе расчета показателя локальной аномальности LOF (Local Outlier Factor) [3]. Данный метод основан на оценке плотности расположения объектов, проверяющихся на аномальность. Его преимущество перед другими методами заключается в способности работать с данными, принадлежащими нескольким классам, имеющим разную плотность.

Ключевой проблемой при обнаружении аномалий является поиск расстояний между объектами данных, поскольку в системах принятия решений используются не только числовые данные, шкалы измерений которых часто заранее известны, но и категориальные данные, выраженные в вербальной форме, что затрудняет их сравнение. В [2] проведен обзор существующих критериев оценки расстояний между значениями категориальных атрибутов данных и выбран оптимальный по мнению авторов критерий оценки расстояния. Проблемой данного критерия является его зависимость от общего количества объектов данных, что затрудняет расчет расстояний в динамических системах, объекты данных в которые могут попадать в процессе работы систем, а не только на начальном этапе формирования модели анализа.

Целью данной работы являлась разработка метода обнаружения аномалий в данных, работающего как с числовыми, так и с категориальными данными, а также способного работать на этапе обучения прогнозной модели при построении СППР. Еще одним плюсом разработанного метода является тот факт, что результаты анализа выводятся не в бинарной форме «да-нет», а в числовом виде, отражающим степень аномальности объекта, что позволяет более гибко принимать решения о последующей обработке данных.

Данная работа организована следующим образом. В первом разделе описывается принцип работы метода расчета показателя локальной аномальности. Во втором разделе приведен критерий расчета расстояний между данными, имеющими категориальные значения. Описаны преимущества данного критерия по сравнению с существующими. В заключении приводятся полученные результаты, обсуждаются направления дальнейших исследований в области работы с аномалиями в данных.

1 Метод расчета показателя локальной аномальности данных

Суть метода расчета локальной аномальности заключается в расчете для каждого объекта данных локальной плотности расположения объектов в текущей точке. Объекты, лежащие в областях наиболее низкой плотности, считаются аномалиями или выбросами. Для оценки плотности используется широко известный метод «к ближайших соседей». Преимущество метода LOF перед другими методами, работающими с плотностью расположения объектов, заключается в том, что в LOF рассматривается так называемая «локальная плотность». Таким образом, LOF успешно распознает выбросы в ситуациях, когда в выборке присутствуют объекты разных классов, не являющиеся аномалиями. Объекты внутри отдельного класса могут иметь разную плотность расположения по сравнению с объектами других классов, что осложняет их обработку и обнаружение аномалий, удаленных от всех классов, как показано на рисунке 2.

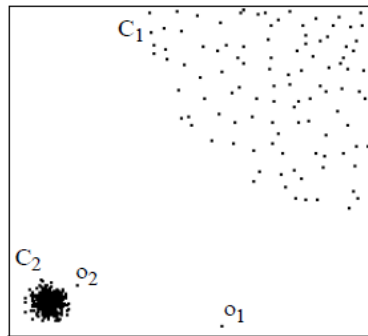


Рисунок 2 – Несколько классов объектов с разной плотностью

Пусть D – множество всех объектов, тогда для любого объекта $p \in D$ расстояние до k -го из ближайших объектов задается формулой:

$$D_k(p) = d(p, o), \quad (1)$$

где $d(p, o)$ – расстояние между объектом p и объектом $o \in D$ таким, что:

а) существует хотя бы k объектов $o' \in D \setminus \{p\}$, для которых выполняется $d(p, o') \leq d(p, o)$;

б) существует не более $(k - 1)$ объектов $o' \in D \setminus \{p\}$, для которых выполняется $d(p, o') < d(p, o)$.

k -окрестностью $N_k(p)$ объекта $p \in D$ называется область, содержащая любой объект q , расстояние от p до которого не больше $D_k(p)$. То есть

$$N_k(p) = \{q \in D \setminus \{p\} \mid d(p, q) \leq D_k(p)\}. \quad (2)$$

Объекты из $N_k(p)$ называются k ближайшими соседями для p . Стоит заметить, что количество объектов в $N_k(p)$ может быть больше k , то есть $|N_k(p)| \geq k$.

Расстоянием достижимости объекта p из объекта o называется расстояние от объекта o до k -го ближайшего объекта, если p принадлежит k -окрестности объекта o . В противном случае расстоянием достижимости является непосредственное расстояние между o и p .

$$RD_k(p, o) = \begin{cases} D_k(o) - \text{если } p \in N_k(o), \\ d(p, o) - \text{иначе.} \end{cases} \quad (3)$$

С учетом определения расстояния до k -го из ближайших объектов расстояние достижимости можно переписать в более компактном виде: $RD_k(p, o) = \max\{D_k(o), d(p, o)\}$.

На рисунке 3 для $k = 3$ показано расстояние достижимости между точками o и $p1$, o и $p2$. Расстояние между o и $p1$ равняется расстоянию до k -го из ближайших объектов, поскольку $p1$ принадлежит k -окрестности точки o . Напротив $p2 \notin N_k(o)$, поэтому $RD_k(p2, o) = d(p2, o)$.

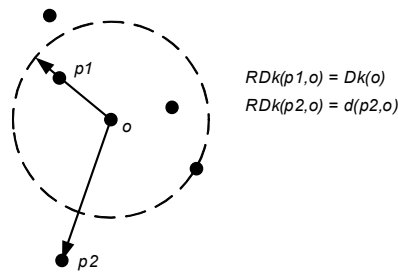


Рисунок 3 – Расстояние достижимости в k -окрестности точки и вне ее

Локальная плотность lrd (local reachability density) в точке p определяется следующим образом:

$$lrd(p) = 1 / \left(\frac{\sum_{o \in N_k(p)} RD_k(p, o)}{|N_k(p)|} \right). \quad (4)$$

То есть локальная плотность в точке p равна обратной величине среднего расстояния достижимости объекта p в его k -окрестности. При уменьшении расстояний достижимости между объектами в k -окрестности локальная плотность растет и приближается к бесконечности при приближении расстояний достижимости к нулю. Иными словами, локальная плотность стремится к бесконечности, если k -окрестность состоит из единственной точки p , однако, данная ситуация исключается по определению k -окрестности. С учетом вышесказанного показатель LOF аномальности объекта p вычисляется следующим образом:

$$LOF(p) = \frac{\sum_{o \in N_k(p)} \frac{lrd(o)}{lrd(p)}}{|N_k(p)|}. \quad (5)$$

То есть степень аномальности объекта p определяется как среднее отношение локальных плотностей точек из k -окрестности точки p к локальной плотности в самой точке p .

2 Расчет расстояний между категориальными данными

Категориальными называются качественные атрибуты, значения которых не принадлежат какой-либо интервальной или порядковой шкале. В [1] предложена формула для вычисления расстояний между значениями категориального атрибута.

Пусть имеется некоторый категориальный атрибут A_n , принимающий значения $D(A_n) = \{x_1, \dots, x_p\}$. Тогда расстояние между значениями x_i и x_j ($i, j \leq p$) при условии, что $x_i \neq x_j$, обозначается как $dist_{A_n}(x_i, x_j)$ и вычисляется следующим образом:

$$dist_{A_n}(x_i, x_j) = \sqrt[2]{\frac{f_n(x_i) + f_n(x_j)}{f_n(x_i) \cdot f_n(x_j)}} \quad (6)$$

Здесь $f_j(x)$ равно количеству объектов, атрибут A_j которых принимает значение x .

То есть расстояние между значениями категориального атрибута рассчитывается как корень из отношения суммы количеств появления соответствующих значений атрибута к их произведению.

По сравнению с существующими критериями расчета расстояний между категориальными значениями атрибутов критерий (6) имеет выигрыш в производительности. Так, в [2] описывается следующий критерий

$$dist_{OF_{A_n}}(x_i, x_j) = \log_2 \frac{N}{f_n(x_i)} \cdot \log_2 \frac{N}{f_n(x_j)} \quad (7)$$

Выигрыш в производительности составляет $\frac{p}{2}$ раз для каждого атрибута объекта данных, где p – это количество значений атрибута, проверяемого на аномальность. Также при добавлении новых данных в систему пропадает необходимость в пересчете расстояний между старыми объектами.

Заключение

Описан метод обнаружения аномалий в данных, способный работать как с количественными, так и с категориальными значениями. Разработан метод расчета расстояний между категориальными значениями, позволяющий существенно ускорить производительность без потери в качестве расчета [1]. Метод позволяет вычислить степень аномальности каждого объекта данных для более гибкой обработки по сравнению с методами, дающими бинарный результат.

Дальнейшие исследования авторы планируют посвятить методам фазификации результатов анализа аномалий в данных. Использование нечеткой логики при обработке данных, содержащих аномалии, позволит пользователям систем поддержки принятия решений вести гибкую настройку системы в удобной форме, в частности, на естественном языке.

Литература

1. Орлов А. О. Проблема поиска расстояний между значениями категориальных атрибутов при обнаружении выбросов в данных // В мире научных открытий. №8.1. 2012. С. 142-155.
2. Boriah, S., Chandola, V., Kumar, V. Similarity measures for categorical data: A comparative evaluation. In Proceedings of the 8th SIAM International Conference on Data Mining, 2008.
3. Breunig, M. M., Kriegel, H.-P., Ng, R. T., Sander, J. LOF: Identifying density-based local outliers. In proceedings of the ACM SIGMOD International Conference on Management of Data. ACM Press, 2000.
4. Chandola, V., Banerjee, A., Kumar, V. Anomaly detection: A survey. ACM Computing Surveys, Vol. 41, No. 3, Article 15, 2009.

СВОЙСТВА БИНАРНЫХ ОПЕРАЦИЙ ЛОГИКО-ГИСТОГРАММНОГО ПРЕДСТАВЛЕНИЯ ДАННЫХ

Папулин С.Ю.

Научный руководитель: к.т.н. Власов А.И.

МГТУ им. Н.Э.Баумана, кафедра ИУ4, Москва, Россия)

PROPERTIES OF BINARY OPERATIONS OF LOGIC-HISTOGRAM DATA REPRESENTATION

Papulin S.Y.

Supervisor: c.t.s. Vlasov A.I.

The Bauman MSTU, Moscow, Russia)

Аннотация

В статье рассматриваются логико-гистограммное представление данных и основные свойства его бинарных операций «И», «ИЛИ», «ИСКЛЮЧАЮЩЕЕ ИЛИ», «КРОМЕ», такие как коммутативность, ассоциативность, дистрибутивность и др.; приводятся основные особенности применения бинарных операций в рамках логико-гистограммного представления.

Abstract

In this article, the logic-histogram data representation and main properties such as commutativity, associativity, distributivity etc., its binary operations “AND”, “OR”, “EXCLUSIVE OR”, “EXCEPT” are considered; moreover, main features of use of the binary operations are provided.

Введение

При анализе данных используются различные способы представления. Одним из наиболее распространенных является гистограммное представление данных. Сама по себе гистограмма достаточно простое средство, которое позволяет количественно определить присутствие некоторого набора элементов в данных. При этом посредством гистограммы возможно узнать о количественных показателях присутствия одного или нескольких (объединенных) элементов из набора. Однако сложности возникают, когда необходимо определить количественные показатели сгруппированных элементов, связанных логическими операциями, т.е. определить количественное значение сложного высказывания. Данную задачу до определенной степени можно решить за счет классических теорий, таких как теория множеств [1], логика высказываний [2,3] (алгебра логики), теория нечетких множество [4], теория вероятности [5]. Однако предлагаемые стандартные решения не позволяют в полной мере осуществить поставленную задачу. Так, если представить гистограмму как множество элементов, то теория множеств (в частности операции над множествами) не учитывает значения элементов гистограммы. При использовании логики высказываний [2,3], например, возможно определить истинность или ложность какого-либо элементного высказывания. Однако в этом случае не определяются количественные показатели. Если применить средства нечеткого анализа [4], то в этом случае даже простое высказывание, состоящее из двух непересекающихся элементов (или наборов объединённых элементов), соединённых конъюнкцией (\wedge или «И»), даст в итоге нулевое значение. Не достигается в полной мере желаемый результат или вовсе искажается и при использовании теорем сложения и умножения вероятностей, а закон и функция распределения случайной величины достаточно ограничены в своих свойствах для формирования и вычисления требуемых значений сложных высказываний.

Поэтому необходимо комплексное средство анализа гистограммных данных, которое использовало бы преимущества вышеуказанных разделов математики и позволяло бы определять количественные показатели элементного состава при использовании сложных высказываний. Одно из таких средств, а именно логико-гистограммное представление

данных, приведено в работе [6] на примере использования в поиске изображений по содержанию.

Далее рассматриваются краткое описание математической составляющей – алгебры – логико-гистограммного представления данных, а также основные свойства бинарных операций, которые являются определяющими элементами для формирования и вычисления количественных значений высказываний.

1 Логико-гистограммное представление данных

Логико-гистограммное представление данные позволяет оценивать количественные значения присутствия элементов в данных с использованием высказываний, состоящих из наименования элементов данных и логических связок. Данное представление включает несколько базовых составляющих: гистограмму данных, алгебру логики гистограммного представления, множество элементов высказывания, функцию вычисления значения высказывания. В работе [6] рассмотрен бинарный случай логико-гистограммного представления и его применение для поиска изображений по гистограммам цвета изображений.

Алгебра логики гистограммного представления имеет следующий вид [6]:

$$\mathcal{H} = \langle H^G; \wedge, \vee, \dot{\vee}, \setminus, \bar{\quad}, \mathbf{0}, \mathbf{1} \rangle, \quad (1)$$

где H^G – множество гистограммных представлений данных, рассматриваемых относительно гистограммы данных H^N , т.е. упорядоченного множества элементов и их количественных значений; $\wedge, \vee, \dot{\vee}, \setminus$ – бинарные (двуместные) операции «И», «ИЛИ», «ИСКЛЮЧАЮЩЕЕ ИЛИ», «КРОМЕ»; $\bar{\quad}$ – унарная (одноместная) операция отрицания; $\mathbf{0}, \mathbf{1}$ – константы ($\mathbf{0}$ – гистограмма, значения всех элементов которой равны нулю, $\mathbf{1} = H^N$, $\mathbf{0}, \mathbf{1} \in H^G$). Далее предполагается, что H^N – нормализованная гистограмма.

В добавление к указанным выше бинарным операциям при необходимости можно добавить объединение (\cup) и пересечение (\cap). Данные операции соответствуют одноименным операциям для нечетких множеств и имеют аналогичные свойства, которые приведены в [4]. Бинарные операции «И», «ИЛИ», «ИСК. ИЛИ», «КРОМЕ» в случае дискретных величин выглядят следующим образом, соответственно:

$$H_1 \wedge H_2 = \begin{cases} H_1, \text{ если } \sum_{i=1}^M H_1(i) < \sum_{i=1}^M H_2(i) \\ \text{иначе} \\ H_2 \end{cases}, \quad (2)$$

$$H_1 \vee H_2 = \left((H(1), \dots, H(M)) \middle| H(i) = \max(H_1(i), H_2(i)) \right), \quad (3)$$

$$H_1 \dot{\vee} H_2 = \begin{cases} H_1, \text{ если } \sum_{i=1}^M H_1(i) > \sum_{i=1}^M H_2(i) \\ \text{иначе} \\ H_2 \end{cases}, \quad (4)$$

$$H_1 \setminus H_2 = \left((H(1), \dots, H(M)) \middle| H(i) = \begin{cases} 0, & H_2(i) > 0 \\ H_1(i), & \text{иначе} \end{cases} \right), \quad (5)$$

где H_1, H_2 – элементы множества H^G ; $H(i)$ – элемент гистограммы; M – количество элементов гистограммы.

Высказывание в логико-гистограммном представлении данных есть множество элементов H^G в сочетании с операциями (1). В итоге, гистограммное высказывание преобразуется в элемент множества H^G посредством введенных операций, а количественный показатель присутствия элементов в гистограмме (значение гистограммного высказывания) вычисляется следующим образом:

$$\Sigma H = \sum_{i=1}^M H(i), \quad (6)$$

где H – элемент множества H^G .

Другими словами, в общем случае гистограммное высказывание вычисляется посредством функции алгебры гистограммной логики. Функция алгебры гистограммной

логики от n переменных x_1, x_2, \dots, x_n называется функция $f_H: (H^G)^n \rightarrow H^G$, т.е. функция, которая произвольному набору (H_1, H_2, \dots, H_n) ставит в соответствие значение $f_H(H_1, H_2, \dots, H_n) \in H^G$, где $H_i \in H^G$. Значение гистограммного высказывания вычисляется посредством функции $\Sigma f_H: H^G \rightarrow [0,1]$, т.е. функции, которая некоторому элементу из H^G ставит в соответствие $\Sigma f_H(H_i) \in [0,1]$. В случае бинарного логико-гистограммного представления для конечных дискретных элементов [6] $\Sigma f_H^D: H^G \rightarrow \{\Sigma H | H \in H^G\}$ или $\Sigma f_H^D: \{0, h_1\} \times \{0, h_2\} \times \dots \times \{0, h_M\} \rightarrow \{\Sigma H | H \in H^G\}$, где $h_j = H(j) \in H$.

В итоге можно записать следующую композицию функций:

$$\begin{aligned} & \Sigma f_H \circ f_H: (H^G)^n \rightarrow [0,1], \text{ или} \\ & (\Sigma f_H \circ f_H)(H_1, H_2, \dots, H_n) = \Sigma f_H(f_H(H_1, H_2, \dots, H_n)) \in [0,1]. \end{aligned} \quad (7)$$

В случае бинарного логико-гистограммного представления для конечных дискретных элементов:

$$\begin{aligned} & \Sigma f_H \circ f_H: (H^G)^n \rightarrow \{\Sigma H | H \in H^G\}, \text{ или} \\ & (\Sigma f_H \circ f_H)(H_1, H_2, \dots, H_n) = \Sigma f_H(f_H(H_1, H_2, \dots, H_n)) \in \{\Sigma H | H \in H^G\}. \end{aligned} \quad (8)$$

Таким образом, для вычисления количественного показателя присутствия элементов может быть использовано логико-гистограммное представление данных. Данное представление позволяет формировать сложные высказывания посредством операций из (1), а количественный показатель вычисляется композицией функций (7).

2 Свойства бинарных операций логико-гистограммного представления

Особенность вычисления количественного показателя заключается в последовательности вычисления гистограммного высказывания посредством f_H , а затем – значения гистограммного высказывания при помощи Σf_H . Поэтому далее свойства бинарных операций логико-гистограммного представления представлены в том же порядке.

Свойства бинарных операций гистограммного высказывания. Для представления свойств введём ограничение, что $\Sigma H_1 \neq \Sigma H_2$, $\Sigma H_2 \neq \Sigma H_3$ и $\Sigma H_1 \neq \Sigma H_3$ кроме случаев, когда элементы H_1, H_2, H_3 равны одному и тому же элементу из H^G . Данное ограничение снимется при вычислении значения гистограммного высказывания.

Операция «И». Результат бинарной операции «И» есть H_1 или H_2 в зависимости от того, каких элементов меньше в данных. При этом H_1 и H_2 обозначают элементы (или наборы элементов) и их количество в данных. Операция «И» имеет следующие основные свойства:

- | | |
|----------------------------|---|
| 1) коммутативность: | $H_1 \wedge H_2 = H_2 \wedge H_1$ |
| 2) ассоциативность: | $H_1 \wedge (H_2 \wedge H_3) = (H_1 \wedge H_2) \wedge H_3$ |
| 3) дистрибутивность: | |
| - относительно «ИЛИ» | $H_1 \wedge (H_2 \vee H_3) \neq (H_1 \wedge H_2) \vee (H_1 \wedge H_3)$ |
| - относительно «ИСКЛ. ИЛИ» | $H_1 \wedge (H_2 \dot{\vee} H_3) = (H_1 \wedge H_2) \dot{\vee} (H_1 \wedge H_3)$ |
| - относительно «КРОМЕ» | $H_1 \wedge (H_2 \setminus H_3) \neq (H_1 \wedge H_2) \setminus (H_1 \wedge H_3)$ |
| 4) свойство нуля: | $H \wedge \mathbf{0} = \mathbf{0}$ |
| 5) свойство единицы: | $H \wedge \mathbf{1} = H$ |
| 6) идемпотентность: | $H \wedge H = H$ |
| 7) свойство поглощения: | $H_1 \wedge (H_1 \vee H_2) = H_1$ |
| 8) законы де Моргана: | $\overline{H_1 \wedge H_2} = \overline{H_1} \dot{\vee} \overline{H_2}$ |

Следующие моменты, относящиеся к выполнению обозначенных выше свойств, нуждаются в уточнении:

- 1) при $\Sigma H_1 = \Sigma H_2$ не выполняется, например, свойство коммутативности;
- 2) дистрибутивность относительно «ИЛИ» не выполняется, например, при $\Sigma H_1 > \Sigma H_2$, $\Sigma H_1 > \Sigma H_3$ и $\Sigma H_1 < \Sigma(H_2 \vee H_3)$;

- 3) дистрибутивность относительно «КРОМЕ» не выполняется, например, при ${}^{\Sigma}H_1 < {}^{\Sigma}H_3$, ${}^{\Sigma}H_3 < {}^{\Sigma}H_2$, $H_2 \vee H_3 = H_2$ и $H_2 \vee H_1 \neq H_2$;
- 4) свойство поглощения относительно операции «ИСКЛ. ИЛИ» выполняется только при условии, что $H_1 \dot{\vee} H_2 = H_1$;
- 5) свойство поглощения относительно операции «КРОМЕ» выполняется только при условии, что $H_1 \setminus H_2 = H_1$;
- 6) закон де Моргана относительно операции «ИЛИ» $\overline{H_1 \wedge H_2} = \overline{H_1} \dot{\vee} \overline{H_2}$ выполняется, если, например, $H_2 \vee H_1 = H_2$ или $H_1 \vee H_2 = H_1$. В общем случае $\overline{H_1 \wedge H_2} \neq \overline{H_1} \dot{\vee} \overline{H_2}$;
- 7) закон де Моргана относительно операции «КРОМЕ» $\overline{H_1 \wedge H_2} = \overline{H_1} \setminus \overline{H_2}$ выполняется, если, например, $H_2 = \mathbf{1}$ и ${}^{\Sigma}H_1 < {}^{\Sigma}H_2$. В общем случае $\overline{H_1 \wedge H_2} \neq \overline{H_1} \setminus \overline{H_2}$.

Операция «ИЛИ». Результат бинарной операции «ИЛИ» есть множество всех элементов H_1 и H_2 . Другими словами, «ИЛИ» соответствует объединению двух множеств. Операция «ИЛИ» имеет следующие основные свойства:

- | | |
|----------------------------|---|
| 1) коммутативность: | $H_1 \vee H_2 = H_2 \vee H_1$ |
| 2) ассоциативность: | $H_1 \vee (H_2 \vee H_3) = (H_1 \vee H_2) \vee H_3$ |
| 3) дистрибутивность: | |
| - относительно «И» | $H_1 \vee (H_2 \wedge H_3) = (H_1 \vee H_2) \wedge (H_1 \vee H_3)$ |
| - относительно «ИСКЛ. ИЛИ» | $H_1 \vee (H_2 \dot{\vee} H_3) = (H_1 \vee H_2) \dot{\vee} (H_1 \vee H_3)$ |
| - относительно «КРОМЕ» | $H_1 \vee (H_2 \setminus H_3) \neq (H_1 \vee H_2) \setminus (H_1 \vee H_3)$ |
| 4) свойство нуля: | $H \vee \mathbf{0} = H$ |
| 5) свойство единицы: | $H \vee \mathbf{1} = \mathbf{1}$ |
| 6) идемпотентность: | $H \vee H = H$ |
| 7) свойство поглощения: | $H_1 \vee (H_1 \setminus H_2) = H_1$ |
| 8) законы де Моргана: | - |

Следующие моменты, относящиеся к свойствам операции «ИЛИ», нуждаются в пояснении:

- 1) при ${}^{\Sigma}H_1 = {}^{\Sigma}H_2$ выполняется свойство коммутативности;
- 2) дистрибутивность относительно «КРОМЕ» не выполняется, например, при ${}^{\Sigma}H_1 < {}^{\Sigma}H_3$, ${}^{\Sigma}H_3 < {}^{\Sigma}H_2$, $H_2 \vee H_3 = H_2$ и $H_2 \vee H_1 \neq H_2$;
- 3) свойство поглощения относительно операции «ИСКЛ. ИЛИ» $H_1 \vee (H_1 \dot{\vee} H_2) = H_1$ выполняется только при условии, что $H_1 \dot{\vee} H_2 = H_1$;
- 4) свойство поглощения относительно операции «И» $H_1 \vee (H_1 \wedge H_2) = H_1$ выполняется только при условии, что $H_1 \wedge H_2 = H_1$ или $H_1 \vee H_2 = H_1$;
- 5) закон де Моргана относительно операции «ИСКЛ. ИЛИ» $\overline{H_1 \vee H_2} = \overline{H_1} \dot{\vee} \overline{H_2}$ выполняется, если, например, $H_1 = H_2$. В общем случае $\overline{H_1 \vee H_2} \neq \overline{H_1} \dot{\vee} \overline{H_2}$;
- 6) закон де Моргана относительно операции «И» $\overline{H_1 \vee H_2} = \overline{H_1} \wedge \overline{H_2}$ выполняется, если, например, $H_2 \vee H_1 = H_2$ или $H_1 \vee H_2 = H_1$. В общем случае $\overline{H_1 \vee H_2} \neq \overline{H_1} \wedge \overline{H_2}$;
- 7) закон де Моргана относительно операции «КРОМЕ» $\overline{H_1 \vee H_2} = \overline{H_1} \setminus \overline{H_2}$ выполняется, если, например, $H_1 = \mathbf{1}$. В общем случае $\overline{H_1 \vee H_2} \neq \overline{H_1} \setminus \overline{H_2}$.

Операция «ИСКЛ. ИЛИ». Результат бинарной операция «ИСКЛ. ИЛИ» в соответствии с [6] есть H_1 или H_2 в зависимости от того, каких элементов (H_1 или H_2) больше в данных. В подобной формулировке не исключается присутствие в данных элемента (или набора), соответствующего H_1 , и одновременно элемента (или набора), соответствующего H_2 .

В такой трактовке операция «ИСКЛ. ИЛИ» имеет следующие основные свойства:

- | | |
|-------------------------|---|
| 1) коммутативность: | $H_1 \dot{\vee} H_2 = H_2 \dot{\vee} H_1$ |
| 2) ассоциативность: | $H_1 \dot{\vee} (H_2 \dot{\vee} H_3) = (H_1 \dot{\vee} H_2) \dot{\vee} H_3$ |
| 3) дистрибутивность: | |
| - относительно «И» | $H_1 \dot{\vee} (H_2 \wedge H_3) = (H_1 \dot{\vee} H_2) \wedge (H_1 \dot{\vee} H_3)$ |
| - относительно «ИЛИ» | $H_1 \dot{\vee} (H_2 \vee H_3) \neq (H_1 \dot{\vee} H_2) \vee (H_1 \dot{\vee} H_3)$ |
| - относительно «КРОМЕ» | $H_1 \dot{\vee} (H_2 \setminus H_3) \neq (H_1 \dot{\vee} H_2) \setminus (H_1 \dot{\vee} H_3)$ |
| 4) свойство нуля: | $H \dot{\vee} \mathbf{0} = H$ |
| 5) свойство единицы: | $H \dot{\vee} \mathbf{1} = \mathbf{1}$ |
| 6) идемпотентность: | $H \dot{\vee} H = H$ |
| 7) свойство поглощения: | $H_1 \dot{\vee} (H_1 \wedge H_2) = H_1$ |
| 8) закон де Моргана: | $\overline{H_1 \dot{\vee} H_2} = \overline{H_1} \wedge \overline{H_2}$ |

По примеру предыдущих операций необходимо прокомментировать некоторые свойства бинарной операции «ИСКЛ. ИЛИ» в соответствии с её трактовкой в работе [6]:

- 1) при ${}^{\Sigma}H_1 = {}^{\Sigma}H_2$ не выполняется, например, свойство коммутативности;
- 2) дистрибутивность относительно «ИЛИ» не выполняется, например, при ${}^{\Sigma}H_1 > {}^{\Sigma}H_3$, ${}^{\Sigma}H_2 < {}^{\Sigma}H_3$ и ${}^{\Sigma}H_1 < {}^{\Sigma}(H_2 \vee H_3)$;
- 3) дистрибутивность относительно «КРОМЕ» не выполняется, например, при ${}^{\Sigma}H_1 > {}^{\Sigma}H_3$, ${}^{\Sigma}H_2 < {}^{\Sigma}H_3$ и ${}^{\Sigma}H_1 < {}^{\Sigma}(H_2 \vee H_3)$;
- 4) свойство поглощения относительно операции «ИЛИ» $H_1 \dot{\vee} (H_1 \vee H_2) = H_1$ выполняется только при условии, что $H_1 \vee H_2 = H_1$;
- 5) свойство поглощения относительно операции «КРОМЕ» $H_1 \dot{\vee} (H_1 \setminus H_2) = H_1$ выполняется только при условии, что $H_1 \setminus H_2 = H_1$;
- 6) закон де Моргана относительно операции «ИЛИ» $\overline{H_1 \dot{\vee} H_2} = \overline{H_1} \vee \overline{H_2}$ выполняется, если, например, $H_1 = H_2$. В общем случае $\overline{H_1 \dot{\vee} H_2} \neq \overline{H_1} \vee \overline{H_2}$;
- 7) закон де Моргана относительно операции «КРОМЕ» $\overline{H_1 \dot{\vee} H_2} = \overline{H_1} \setminus \overline{H_2}$ выполняется, если, например, $H_1 = \mathbf{1}$. В общем случае $\overline{H_1 \dot{\vee} H_2} \neq \overline{H_1} \setminus \overline{H_2}$.

Операция «КРОМЕ». Результат бинарной операции «КРОМЕ» есть множество всех элементов H_1 кроме тех элементов, которые есть в H_2 . Другими словами, «КРОМЕ» соответствует вычитанию одного множества из другого. Операция «КРОМЕ» имеет следующие основные свойства:

- | | |
|----------------------|---------------------------------------|
| 1) свойство нуля: | $H \setminus \mathbf{0} = H$ |
| 2) свойство единицы: | $H \setminus \mathbf{1} = \mathbf{0}$ |
| 3) идемпотентность: | $H \setminus H = \mathbf{0}$ |

Таким образом, операция «КРОМЕ» не является коммутативной, ассоциативной, дистрибутивной, не выполняется свойство поглощения и закон де Моргана, за исключением частных случаев.

Свойства бинарных операций при вычислении значения гистограммного высказывания. В данном случае снимается ограничение ${}^{\Sigma}H_1 \neq {}^{\Sigma}H_2$, ${}^{\Sigma}H_2 \neq {}^{\Sigma}H_3$ и ${}^{\Sigma}H_1 \neq {}^{\Sigma}H_3$, т.к. значение гистограммного высказывания, вычисляемое по функции ${}^{\Sigma}f_H$, учитывает исключительно сумму количественных показателей элементов H_1 и H_2 , а не их элементный состав.

Поэтому свойства бинарных операций при вычислении значения гистограммного высказывания примут следующий вид:

- операция «И»:

- 1) коммутативность:
$$\Sigma(H_1 \wedge H_2) = \Sigma(H_2 \wedge H_1)$$
- 2) ассоциативность:
$$\Sigma(H_1 \wedge (H_2 \wedge H_3)) = \Sigma((H_1 \wedge H_2) \wedge H_3)$$
- 3) дистрибутивность:
$$\Sigma(H_1 \wedge (H_2 \dot{\vee} H_3)) = \Sigma((H_1 \wedge H_2) \dot{\vee} (H_1 \wedge H_3))$$
- 4) свойство нуля:
$$\Sigma(H \wedge \mathbf{0}) = \mathbf{0}$$
- 5) свойство единицы:
$$\Sigma(H \wedge \mathbf{1}) = \Sigma H$$
- 6) идемпотентность:
$$\Sigma(H \wedge H) = \Sigma H$$
- 7) свойство поглощения:
$$\Sigma(H_1 \wedge (H_1 \vee H_2)) = \Sigma H_1$$
- 8) законы де Моргана:
$$\Sigma(\overline{H_1 \wedge H_2}) = \Sigma(\overline{H_1} \dot{\vee} \overline{H_2})$$

- операция «ИЛИ»:

- 1) коммутативность:
$$\Sigma(H_1 \vee H_2) = \Sigma(H_2 \vee H_1)$$
- 2) ассоциативность:
$$\Sigma(H_1 \vee (H_2 \vee H_3)) = \Sigma((H_1 \vee H_2) \vee H_3)$$
- 3) дистрибутивность:
$$\Sigma(H_1 \vee (H_2 \wedge H_3)) = \Sigma((H_1 \vee H_2) \wedge (H_1 \vee H_3))$$

$$\Sigma(H_1 \vee (H_2 \dot{\vee} H_3)) = \Sigma((H_1 \vee H_2) \dot{\vee} (H_1 \vee H_3))$$
- 4) свойство нуля:
$$\Sigma(H \vee \mathbf{0}) = \Sigma H$$
- 5) свойство единицы:
$$\Sigma(H \vee \mathbf{1}) = \mathbf{1}$$
- 6) идемпотентность:
$$\Sigma(H \vee H) = \Sigma H$$
- 7) свойство поглощения:
$$\Sigma(H_1 \vee (H_1 \setminus H_2)) = \Sigma H_1$$

- операция «ИСКЛ. ИЛИ»:

- 1) коммутативность:
$$\Sigma(H_1 \dot{\vee} H_2) = \Sigma(H_2 \dot{\vee} H_1)$$
- 2) ассоциативность:
$$\Sigma(H_1 \dot{\vee} (H_2 \dot{\vee} H_3)) = \Sigma((H_1 \dot{\vee} H_2) \dot{\vee} H_3)$$
- 3) дистрибутивность:
$$\Sigma(H_1 \dot{\vee} (H_2 \wedge H_3)) = \Sigma((H_1 \dot{\vee} H_2) \wedge (H_1 \dot{\vee} H_3))$$
- 4) свойство нуля:
$$\Sigma(H \dot{\vee} \mathbf{0}) = \Sigma H$$
- 5) свойство единицы:
$$\Sigma(H \dot{\vee} \mathbf{1}) = \mathbf{1}$$
- 6) идемпотентность:
$$\Sigma(H \dot{\vee} H) = \Sigma H$$
- 7) свойство поглощения:
$$\Sigma(H_1 \dot{\vee} (H_1 \wedge H_2)) = \Sigma H_1$$
- 8) закон де Моргана:
$$\Sigma(\overline{H_1 \dot{\vee} H_2}) = \Sigma(\overline{H_1} \wedge \overline{H_2})$$

- операция «КРОМЕ»:

- 1) свойство нуля:
$$\Sigma(H \setminus \mathbf{0}) = \Sigma H$$
- 2) свойство единицы:
$$\Sigma(H \setminus \mathbf{1}) = \mathbf{0}$$
- 3) идемпотентность:
$$\Sigma(H \setminus H) = \mathbf{0}$$

Таким образом, свойства бинарных (двуместных) операций логико-гистограммного представления данных складываются из свойств операций гистограммного высказывания и свойств при вычислении значения гистограммного высказывания. Если рассматривать логико-гистограммное представление как композицию функций (7), то в упрощенном виде свойства бинарных операций данного представления возможно определить только как свойства вычисления значения гистограммного высказывания.

Заключение

Логико-гистограммное представление данных – это комплекс средств, позволяющих определять количественные показатели присутствия некоторого элементного состава в данных посредством анализа гистограммы данных. Важным компонентом данного представления является математический аппарат, который используется для определения количественных показателей. Поэтому в данной работе была рассмотрена математическая сторона логико-гистограммного представления данных, которая включает алгебру гистограммной логики, функции вычисления гистограммного высказывания и его значения, а также свойства бинарных операций.

Литература

1. Хаусдорф Ф. Теория множеств: пер. с нем./под ред., с предисл. и под. П.С. Александрова, А.Н. Колмогорова. Изд. 5-е. М.: Издательство ЛКИ, 2010. 304с.;
2. Судоплатов С.В., Овчинникова Е.В. Дискретная математика: Учебник. 2-е изд., переработ. М.: ИНФРА-М; Новосибирск: изд-во НГТУ, 2005. 256с.;
3. Матвеев М.Г., Свиридов А.С., Алейникова Н.А. Модели и методы искусственного интеллекта. Применение в экономике: учеб. пособие. М.: Финансы и статистика; ИНФРА-М, 2008. 448с.;
4. Коньшева Л.К., Назаров Д.М. Основы теории нечетких множеств: Учебное пособие. СПб.: Питер, 2011. 192с.;
5. Кремер Н.Ш. Теория вероятностей и математическая статистика: Учебник для вузов. 2-е изд., перераб. и доп. М.: ЮНИТИ-ДАНА, 2004. 573с.;
6. Папулин С.Ю. Способ поиска изображений по содержанию на основе логико-гистограммного представления признака цвета с применением текстового запроса пользователя // Итоги диссертационного исследования. Том 3. Материалы IV Всероссийского конкурса молодых ученых. М.: РАН, 2012. С.95-106.

ИСПОЛЬЗОВАНИЕ КЛЕТОЧНОГО АСИНХРОННОГО МЕТОДА ДЛЯ РЕШЕНИЯ КРАЕВЫХ ЗАДАЧ ДЛЯ ДУЧП НА ГПУ

Погосский М.П.

Научный руководитель: д.т.н., профессор, Карпенко А.П.

МГТУ им. Н.Э.Баумана, кафедра РК-6, Москва, Россия

ASYNCHRONOUS CELLULAR METHOD USING FOR PDE BOUNDARY VALUE PROBLEMS SOLUTION WITH GPU

Pogoskiy M.P.

Supervisor: Dr., Prof., Karpenko A.P.

MSTU, Moscow, Russia

Аннотация

В статье представлены результаты исследования эффективности клеточного асинхронного метода решения краевых задач для ДУЧП на графических процессорных устройствах. Дана постановка задачи и описание указанного метода ее решения. Обоснована целесообразность модификации метода. Рассмотрена схема отображения алгоритма, реализующего модифицированный метод асинхронных итераций, на архитектуру ГПУ. Представлены результаты численных экспериментов на двух классах ГПУ. В заключении сформулированы основные результаты работы и перспективы ее развития.

Abstract

This paper shows the results of asynchronous cellular method efficiency research for PDE boundary value problems solution with graphical processing units. The problem statement and method description are given. The expediency of this method modification is justified. Also details of the mapping modified algorithm to GPU architecture and numerical experiments results for two types of GPU devices are presented. The results of this work and its perspectives are shown in the conclusion.

Введение

Численное решение краевых задач для дифференциальных уравнений в частных производных (ДУЧП) является одной из центральных проблем математического моделирования. В силу высокой вычислительной сложности этих задач разработано большое число параллельных методов и алгоритмов их решения, ориентированных на различные классы параллельных вычислительных систем. В последние годы значительное внимание уделяется разработке параллельных методов и алгоритмов, ориентированных на графические процессорные устройства (ГПУ). Целью работы является исследование эффективности, так называемого клеточного асинхронного метода решения ДУЧП на ГПУ.

1 Постановка задачи и схема метода

Рассматриваем для простоты записи одномерную модель распространения тепла в стержне в форме уравнения теплопроводности

$$\frac{\partial U(x,t)}{\partial t} = a^2 \frac{\partial^2 U(x,t)}{\partial x^2},$$

при заданных начальных и граничных условиях $U(0,t) = 0$, $U(l,t) = 0$ при $t > 0$ и $U(x,0) = f(x)$ при $0 \leq x \leq l$.

Выполним дискретизацию задачи по пространству и времени с шагами h и τ соответственно. Значения переменных x, t в узлах полученной сетки обозначим в точке (x_i, t_j) , а значения непрерывной функции $U(x,t) - U(x_i, t_j)$. В результате ДУЧП (1) приобретает форму системы конечно-разностных уравнений

$$\frac{w_{i,j} - w_{i,j-1}}{\tau} - a^2 \frac{w_{i+1,j} - 2w_{i,j} + w_{i-1,j}}{h^2} = 0.$$

Здесь m, l – числа шагов дискретизации по переменным x, t соответственно.

Введем обозначение $\lambda = \frac{\tau a^2}{h^2}$. Тогда система (2) приобретает форму системы линейных алгебраических уравнений (СЛАУ) вида

$$w_{i,j-1} = (1 + 2\lambda)w_{i,j} - \lambda w_{i+1,j} - \lambda w_{i-1,j}, i = [1 : m], j = [1 : l]$$

или в матричной форме – вида

$$L_h W_h = U_h,$$

где $L_h = (l_1(w), l_2(w), l_3(w), \dots, l_n(w))$ – разностный оператор, задающий отображение $L_h : R^n \rightarrow R^n$; U_h – сетевая функция; W_h – искомая разностная функция.

В общем случае вид оператора L_h определяет используемый конечно-разностный шаблон. Основным параметром шаблона является его степень соседства r – максимальное расстояние между центральным узлом шаблона и его периферийными узлами. Число соседей узла обозначаем a_h .

Рассматриваем итерационный клеточный локально-асинхронный метод решения СЛАУ (3). В работе [1], в котором рассматривается теория этого метода, изложение ведется в терминах клеточных нейронных сетей. Поэтому можно сказать, что целью нашей работы является исследование эффективности реализации клеточных нейронных сетей на ГПУ.

Каждому из узлов $w_{i,j}$ рассматриваемой конечно-разностной сетки ставим в соответствие клетку $w_i(j); i = [1 : n], j = [1 : l]$, динамику которой определяет итерационная формула вида

$$w_i(j) = w_i(j-1) - \frac{K}{h^2} l_i(w_k) s_k(j), k \in J(u_i),$$

где $0 \leq s_k(j) \leq j$ – задержка между текущим временем t_j и временем $t_{s_k(j)}$, которому соответствует состояние клетки w_k , принадлежащей окружению рассматриваемого узла $w_{i,j}$; J – набор a_h штук номеров клеток этого окружения; $K = t_h a_h^2$. Отметим, что ситуация $s_k(j) = j$ определяет синхронный клеточный метод.

Базовое свойство асинхронного метода состоит в том, что для продолжения итераций в данном узле сеточной области не нужно ожидать завершения итераций в соседних узлах. Исследование погрешности данного метода для одномерной задачи теплопроводности для стержня с использованием трех-точечного шаблона представлено в работе [3].

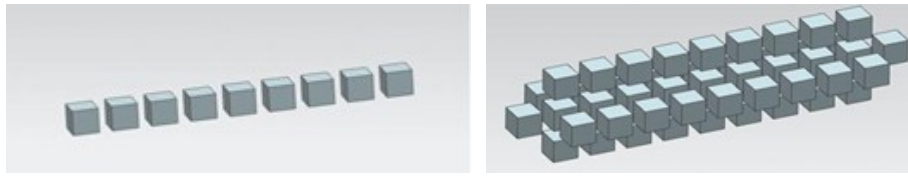
2 Отображение алгоритма на ГПУ

Условно-асинхронный клеточный метод реализован на архитектуре *NVidia CUDA* для трехмерной модели теплопроводности

$$\frac{\partial U(x, y, z, t)}{\partial t} = a^2 \left(\frac{\partial^2 U(x, y, z, t)}{\partial x^2} + \frac{\partial^2 U(x, y, z, t)}{\partial y^2} + \frac{\partial^2 U(x, y, z, t)}{\partial z^2} \right),$$

аналогичной модели (1). Расчеты выполнены для кубического тела, грани которого имеют постоянные температуры 50, 60, 70, 100, 120 и 200 градусов соответственно. Начальные условия внутри границ приняты нулевыми.

С точки зрения эффективности реализации соответствующего алгоритма на ГПУ, ключевой проблемой является проблема оптимального разбиения клеток на блоки. В работе использован шаблон, изображенный на рисунке 2. Полагаем, что используется семи-точечный шаблон и каждой клетке соответствует *CUDA*-нить. Каждый из блоков представляет собой строку клеток по одному из измерений расчетной области (рисунок 3).



а) б)
Рисунок 2 — Используемый вариант разбиения клеток на блоки

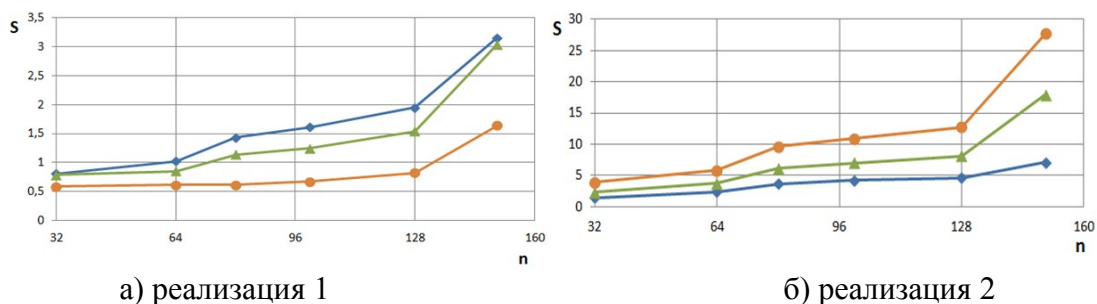
С целью ограничения периода асинхронного выполнения нитей в *CUDA*-программе, реализующей условно-асинхронный алгоритм, приняты следующие решения. Во-первых, число итераций в пределах каждого запуска блока ограничено величиной N , что ограничивает степень асинхронности метода. Во-вторых, производится барьерная синхронизация блоков, реализованная с помощью функции завершения выполнения всех нитей на ГПУ.

Программная реализация алгоритма поддерживает два отображения условно-асинхронного алгоритма ГПУ: 1) с использованием только глобальной памяти; 2) с использованием разделяемой памяти. В зависимости от модели используемого ГПУ и размерности расчетной сетки оптимальной может быть как первая, так и вторая реализации.

3 Вычислительный эксперимент

Вычислительный эксперимент выполнен на вычислительной системе, состоящей центрального процессора и двух ГПУ *GeForce GTS250* и *GeForce GTX470*. Для оценки ускорения S , которое обеспечивает алгоритм и его программная реализация, он был реализован также на центральном процессоре (ЦП) системы. Для асинхронного алгоритма показательным является ускорение в функции параметра K (формула (4)), который определяет скорость сходимости алгоритма, т. е. число итераций, необходимых для перехода клеточной сети в устойчивое состояние (которое и соответствует решению задачи). На основе представленного в работе [3] исследования зависимости числа итераций от коэффициента K в данной работе использовано значение $K=0,2$. Замеры ускорений параллельного алгоритма по сравнению с ЦП в зависимости от числа узлов расчетной сетки проводились для следующих размеров сетки: $32 \times 32 \times 32$, $64 \times 64 \times 64$, $80 \times 80 \times 80$, $100 \times 100 \times 100$, $128 \times 128 \times 128$, $150 \times 150 \times 150$.

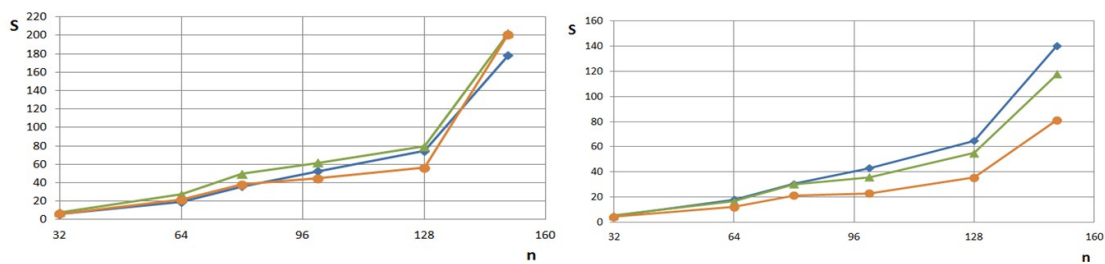
На рисунках 3а, 3б показаны зависимости ускорения, полученные для графической карты *GeForce GTS250* для первой и второй реализации алгоритма соответственно. Ромбиками, треугольниками и кружками помечены кривые для $N = 1, 5, 10$ соответственно. На рисунках 4а, 4б представлены аналогичные результаты для графической карты *GeForce GTX470*.



а) реализация 1

б) реализация 2

Рисунок 3 — Ускорение вычислений в функции числа клеток в ребре кубического поля n : ГПУ *GeForce GTS250*



а) реализация 1

б) реализация 2

Рисунок 4 — Ускорение вычислений в функции числа клеток в ребре кубического поля n : ГПУ *GeForce GTX470*

В случае использования для расчетов графической карты *GeForce GTX470* положительный эффект от использования разделяемой памяти не проявляется, благодаря наличию в этом устройстве КЭШ-памяти *L2*. Поэтому в данном случае при больших размерах поля эффективнее использовать первую реализацию.

Заключение

В работе с помощью имитационной модели выполнено исследование эффективности асинхронного клеточного метода решения ДУЧП. На основе результатов исследования предложена модификация этого метода, предполагающая регулярную глобальную синхронизацию вычислений. Разработаны две *CUDA*-реализации соответствующего алгоритма, одна из которых предполагает использование только глобальной памяти ГПУ, а другая - разделяемой памяти. Проведено широкое экспериментальное исследование эффективности предложенной модификации метода, соответствующего алгоритма и указанного программного обеспечения. Результаты исследования показывают перспективность развития асинхронного клеточного метода решения ДУЧП на ГПУ.

В продолжении работы планируется исследование эффективности метода в сравнении со параллельными численными методами решения краевых задач дифференциальных уравнений в частных производных.

Литература

1. Б.Б. Нестеренко, М.А. Новотарский. Локально-асинхронный метод решения уравнений математической физики на клеточных нейронных сетях // «Компьютерное моделирование и интеллектуальные системы», сборник научных трудов, 2007, с.296 – 307.
2. А.Е. Алексеенко, А.М. Казённов. Реализация клеточных автоматов «игра Жизнь» с применением технологий *CUDA* и *OpenCL* // Компьютерные исследования и моделирование, 2010, Т.2, №3, с. 323–326.
3. А.П. Карпенко, М.П. Погосский. Клеточный асинхронный метод решения ДУЧП на ГПУ / Научный сервис в сети Интернет: поиск новых решений: Труды Международной суперкомпьютерной конференции (17-22 сентября 2012 г., г. Новороссийск). – М.: Изд-во МГУ, 2012, с. 171-176.

РЕШЕНИЕ ЗАДАЧИ БАЛАНСИРОВКИ ЗАГРУЗКИ ГРАФИЧЕСКИМИ МЕТОДАМИ

Вершинин В.Д., Романова А.О.
Научный руководитель д. ф.-м. наук, профессор, Карпенко А.П.
МГТУ имени Н.Э.Баумана, кафедра РК6, Москва, Россия

LOAD BALANCING PROBLEMS SOLVING WITH GRAPHICAL METHODS.

Anastasia Romanova
Supervisor Karpenko A.P.
Moscow State Technical University N.A. N.E. Bauman

Аннотация

В работе рассматривается проблема балансировки загрузки облачной среды. Поставлена задача балансировки загрузки. Предложен метод решения задачи. Представлен алгоритм балансировки.

Abstract

This paper considers the problem of load balancing cloud environment. The task of load balancing is formulated. A method of solving the problem is given. The algorithm for balancing is provided.

Введение

В процессе исследований, посвященных способам адаптации облачной среды к проведению ресурсоемких научных вычислений, авторы столкнулись с проблемой балансировки загрузки облака при таких вычислениях [1]. Основная идея метода балансировки загрузки состоит в распределении вычислений по процессорам таким образом, чтобы суммарная вычислительная и коммуникационная загрузки процессоров были примерно одинаковы. При этом не учитываются коммуникационные загрузки процессоров, обусловленные транзитными обменами и конфликты при обменах вследствие перегрузки коммуникационной сети. Часто при балансировке загрузки коммуникационные расходы не учитываются вообще.

Постановка задачи

Полагаем, что задача $\mathbf{T}=(T_j, j \in [1:n])$ представляет совокупность достаточно больших подзадач T_1, T_2, \dots, T_n , информационные связи между которыми представляют собой, вообще говоря, циклический динамический граф G_T . Таким образом, имеются в виду крупнозернистый параллелизм на уровне крупных фрагментов алгоритма.

Подзадача $T_j, j \in [1:n]$ переходит в состояние «готова к выполнению», если готовы данные для всех ее входов $In_k(T_j), k \in [1:m_j]$. В противном случае подзадача находится в состоянии «ожидание». Готовая к выполнению подзадача, которая назначена на исполнение некоторому виртуальному серверу, переходит в состояние «выполнение». В процессе выполнения подзадачи она не взаимодействует с другими подзадачами, т.е. коммуникации между подзадачами возможны только после их завершения. Положим, что в некоторый момент времени t в состоянии «готова к выполнению» находится набор подзадач $\mathbf{T}'=(T'_1, T'_2, \dots, T'_{n'})$, где $n' \leq n$. Отображение подзадач \mathbf{T}' на архитектуру облака задаем отображающей $(N \times n')$ булевой матрицей X' , в которой $x'_{i,s}=1$ означает, что подзадача $T'_s, s \in [1:n']$ должна быть выполнена на виртуальном сервере $P_i, i \in [1:N]$, а $x'_{i,s}=0$ - что указанная подзадача не должна выполняться на этом сервере.

Рассматриваем два критерия оптимальности отображения - суммарную стоимость $c(X')$ и время $t(X')$ решения подзадач набора \mathbf{T}' :

$$c(X') = \sum_{i=1}^N \sum_{s=1}^{n'} (c^{CPU} + c_i^{RAM} + c_i^{HDD}) p_i q_s x'_{i,s} \rightarrow \min ;$$

$$t(X') = \max_{i \in [1: N]} \sum_{s=1}^{n'} p_i q_s x'_{i,s} \rightarrow \min .$$

В качестве скалярного критерия оптимальности отображения $J(X')$ используем аддитивную свертку критериев $c(X')$, $t(X')$, т.е. полагаем

$$J(X') = \lambda c(X') + (1 - \lambda) t(X') \rightarrow \min , \quad (1)$$

где λ и $(1 - \lambda)$ - относительные веса первого и второго критериев соответственно.

Задачу (1) приходится решать в условиях очевидных ограничений

$$\sum_{s=1}^{n'} q_s^{RAM} x'_{i,s} \leq o_i^{RAM} , \quad \sum_{s=1}^{n'} q_s^{HDD} x'_{i,s} \leq o_i^{HDD} ; \quad i \in [1: N] . \quad (2)$$

Поскольку задача (1), (2) не учитывает временные затраты на коммуникации, эта задача представляет собой задачу балансировки загрузки. Путем наложения соответствующих ограничений на матрицу X' отображение задачи может быть выполнено также на гетерогенный набор виртуальных серверов.

Методы декомпозиции графов

Задачу балансировки можно поставить как задачу разрезания графа (G, D) на N (по числу процессоров в системе) непересекающихся подмножеств так, чтобы суммы весов узлов в каждом подмножестве были приблизительно равны, а сумма весов ребер, которые инцидентны узлам из разных подмножеств, была минимальна. В соответствии с методами [2] разрезание графа (G, D) на подграфы осуществляется в три этапа:

- 1) рекурсивное огрубление графа;
- 2) рекурсивная бисекция огрубленного графа;
- 3) восстановление исходного графа.

Рекурсивное огрубление графа

Суть подхода к задаче огрубления графа состоит в итерационном стягивании смежных узлов графа G^α в узлы графа $G^{\alpha+1}$, где $\alpha=1,2,\dots$ – номер итерации, $G^0=G(0)$. В данной работе рассматриваются алгоритмы паросочетаний, когда граф $G^{\alpha+1}$ строится на основе графа G^α путем нахождения в графе G^α паросочетания и стягивания в мультиузел узлов, входящих в каждую из пар этого паросочетания. При этом непарные узлы графа G^α просто копируются в граф $G^{\alpha+1}$.

В процессе стягивания узлов в мультиузел, вес этого мультиузла находится как сумма стянутых в него узлов. Вес дуги определяется как сумма весов дуг, стянутых в нее в процессе огрубления графа. Помимо веса каждому узлу назначается еще один параметр – m_i , $i \in [1: n^O]$. Он равен сумме весов ребер, стянутых в процессе образования мультиузла. Очевидно, что в узлах графа $G(0)$ значение этого параметра равно нулю.

В работе [3] были рассмотрены три метода построения паросочетаний:

- алгоритм случайных паросочетаний A_{RM} ,
- алгоритм паросочетаний из тяжелых ребер A_{HEM} ,
- алгоритм паросочетаний из тяжелых клик A_{HCM} .

В методе случайных паросочетаний узлы посещаются в случайном порядке. Если узел i не был включен в паросочетание, то беспорядочно выбирается один из его смежных узлов, который также не включен в паросочетание. Если такой узел j существует, ребро e_{ij} включается в паросочетание и маркируются узлы i и j как посещенные. Если не имеется никакого немаркированного смежного узла j , то узел i остается свободным и переходит в следующий граф.

Паросочетание из тяжелых ребер вычисляется с использованием случайного алгоритма, подобного описанному ранее для вычисления случайного паросочетания. Узлы снова посещаются в случайном порядке. Однако теперь, вместо беспорядочного сочетания узла i с одним из его смежных немаркированных узлов, i сочетается с узлом j таким, что вес ребра e_{ij} максимален для всех доступных ребер (наиболее тяжелое ребро). Выигрыш по сравнению с методом случайных паросочетаний заключается в том, что вес ребра соответствует числу стянутых в него ребер. Это означает, что реберная плотность мультиузла, образованного стягиванием узлов i и j , соединенных самым тяжелым ребром, с высокой долей вероятности будет наибольшей из всех возможных вариантов.

Метод построения паросочетаний из тяжелых клик вычисляет паросочетание, стягивая узлы, которые имеют высокую плотность ребер. Таким образом, эта схема вычисляет паросочетание, чья плотность ребер максимальна. Как и в предыдущих схемах вычисления паросочетания, паросочетание из тяжелых клик вычисляется с использованием случайного алгоритма. Узлы посещаются в случайном порядке. Немаркированный узел i сочетается с его немаркированным смежным узлом j так, что плотность ребер мультиузла, созданного объединением i и j , максимальна среди всех возможных мультиузлов, включающих i и другой немаркированный сосед узла i . Плотность ребер между узлами v и u определяется по формуле [4]:

$$B_{ij} = \frac{2 \cdot (m_i + m_j + v_{ij})}{(w_i + w_j) \cdot (w_i + w_j - 1)}$$

Схема *HCM* очень похожа на *HEM*. Единственное различие в том, что *HEM* сочетает узлы, которые только связаны тяжелым ребром независимо от суммарного реберного веса узлов, в то время как *HCM* сочетает узлы, если они связаны тяжелым ребром и если каждый из этих двух узлов имеет большой реберный вес.

На основе описанных выше методов в работе предлагается следующий алгоритм огрубления графа:

- 1) все узлы V^α текущего графа G^α объявляем немаркированными;
- 2) если все узлы V^α принадлежат графу $G(T)$, значит алгоритм достиг цели, его работа завершается, иначе переход к следующему шагу;
- 3) случайным образом выбирается немаркированный узел, еще не включенный в паросочетание - пусть это будет узел i^α ;
- 4) по выбранному методу из числа немаркированных узлов, смежных узлу i^α выбирается узел (пусть это будет узел j^α), также еще не включенный в паросочетание;
- 5) если оба узла или один из узлов пары i^α, j^α не принадлежат графу $G(T)$, то ребро e_{ij}^α включается в паросочетание, и узлы i^α, j^α маркируются;
- 6) если ни одного немаркированного узла, смежного узлу i^α , не существует, то узел i^α маркируется и остается свободным (чтобы затем перенести его в граф $G^{\alpha+1}$);
- 7) если в графе G^α имеются еще немаркированные узлы, то переход к шагу 3, в противном случае переход к следующему шагу;
- 8) образованные пары стягиваются в мультиузлы (причем если один из стягиваемых узлов принадлежит графу $G(T)$, стягивание производится к нему), рассчитываются веса этих узлов и веса соединяющих их дуг, снимается маркировка, переход к шагу 1.

Рекурсивная бисекция огрубленного графа

На втором этапе метода балансировки загрузки выполняется рекурсивное деление графа, полученного на первом этапе алгоритма. Деление выполняется таким образом, чтобы каждая из частей содержала примерно равную часть узлового веса первоначального графа [5].

Бисекция графа может быть выполнена с использованием различных алгоритмов:

- поисковый алгоритм
- алгоритм бисекции путем наращивания.

Восстановление исходного графа

На данном этапе каждый из подграфов, на которые был разделен огрубленный граф на предыдущем этапе, восстанавливается до соответствующей части исходного графа. В простейшем случае этап заканчивается этим восстановлением подграфов.

Поскольку полученные подграфы имеют значительно большее количество степеней свободы, чем огрубленные подграфы, можно попытаться улучшить полученное разделение с помощью какого-либо алгоритма локального уточнения. Можно, например, использовать модификацию рассмотренного поискового алгоритма, разрешив перемещение между подграфами только граничных узлов.

Основная цель локального уточнения заключается в том, чтобы определить для каждого из граничных узлов тот подграф, при перемещении в который вес разрезанных ребер будет минимален при сохранении равномерного распределения узлов по подграфам. Поскольку количество граничных узлов относительно невелико, локальное уточнение может быть выполнено за приемлемое время.

Заключение

В работе предложена модификация алгоритма огрубления графа на основе трех рассмотренных ранее методов: случайных паросочетаний, паросочетаний из тяжелых ребер, также паросочетаний из тяжелых клик.

Дальнейшая работа будет посвящена исследованию эффективности алгоритма для циклических динамических графов, специфичных для решаемой задачи.

Литература

1. Романова А.О. Научные вычисления в облаках [Электронный ресурс] // Наука и образование: электронное научно - техническое издание, 2012, 9 (<http://sntbul.bmstu.ru/doc/479373.html>).
2. Карпенко А.П., Федорук В.Г., Федорук Е.В. Исследование эффективности балансировки загрузки многопроцессорной системы при распараллеливании одного класса вычислительных задач [Электронный ресурс] // Наука и образование: электронное научно - техническое издание, 2007, 8 (<http://technomag.edu.ru/doc/67392.html>).
3. Вершинин В. Д., Карпенко А. П., Плякин Д. А. Некоторые методы огрубления графов при оценке релевантности документов [Электронный ресурс] // Наука и образование: электронное научно - техническое издание, 2012, 6 (<http://technomag.edu.ru/doc/432649.html>).
4. Карпенко А.П. Оценка релевантности документов онтологической базы знаний [Электронный ресурс] // Наука и образование: электронное научно - техническое издание, 2010, 9 (<http://technomag.edu.ru/doc/157379.html>).
5. Бувайло Д.П., Толлок В.А. Быстрый высокопроизводительный алгоритм для разделения нерегулярных графов // Вісник Запорізького державного університету, 2002, № 2, с. 1 – 10.

ПРОГРАММНЫЙ КОМПЛЕКС ДЛЯ ОЦЕНКИ КАЧЕСТВА ПАРЕТО-АППРОКСИМАЦИИ В ЗАДАЧЕ МНОГОЦЕЛЕВОЙ ОПТИМИЗАЦИИ

Грабик А.В., Шибитов И.А.

Научные руководители: ст. преп. Грошев С.В., д.ф-м.н., проф. Карпенко А.П.

МГТУ им. Н.Э. Баумана, кафедра РК-6, Москва, Российская Федерация

SOFTWARE SYSTEM FOR QUALITY ESTIMATION OF PARETO-APPROXIMATION IN THE PROBLEM OF MULTI-OBJECTIVE OPTIMIZATION

Grabik A.V., Shibitov I.A.

Supervisors: s.t., Groshev S.V., Dr. Karpenko A.P.

Bauman Moscow State Technical University, Moscow, Russian Federation

Аннотация

Рассмотрена проблема оценки качества Парето-аппроксимации в задаче многоцелевой оптимизации. Выполнен анализ способов оценки и выбран ряд индикаторов, формализующих качество аппроксимации. Разработаны программы, реализующие вычисление, как унарных, так и бинарных индикаторов.

Abstract

The problem quality estimation of the Pareto-approximation in multi-objective optimization problem is considered. The analysis of the ways to assess the quality was held and number of selected indicators, formalizing quality of the approximation was selected. Created programs that implement the computation as unary as binary criteria of approximation.

Введение

Классические алгоритмы решения задачи многоцелевой оптимизации (МЦО-задачи) основаны на сведении ее к совокупности задач глобальной одноцелевой оптимизации, что не всегда дает нужный результат. Относительно новый и быстро развивающийся класс алгоритмов многоцелевой оптимизации образуют алгоритмы Парето-аппроксимации, предполагающие предварительное построение некоторой конечномерной аппроксимации множества, а тем самым, и фронта Парето.

Пусть множеством допустимых значений вектора варьируемых параметров X является ограниченное и замкнутое множество $D_X = \{X \mid G(X) \geq 0\} \subset \{X\} = R^{|X|}$. Положим, что целевая вектор-функция $F(X) = (f_1(X), f_2(X), \dots, f_{|F|}(X))$ со значениями в целевом пространстве $\{F\} = R^{|F|}$ определена в области D_X . Целью является минимизация в этой области каждой из частных целевых функций $f_1(X), f_2(X), \dots, f_{|F|}(X)$, что условно записываем в виде $\min_{X \in D_X} F(X) = F(X^*) = F^*$, где векторы X^* , F^* - искомое решение задачи многоцелевой оптимизации.

Целевая вектор-функция $F(X)$ выполняет отображение множества D_X в некоторое множество D_F целевого пространства, которое называется множеством достижимости. Введем на множествах D_F , D_X отношения доминирования и строго доминирования. Вектор $F_1 = F(X_1) \in D_F$ доминирует вектор $F_2 = F(X_2) \in D_F$ что записываем в виде $F_1 \bar{\triangleright} F_2$, если среди равенств и неравенств $f_k(X_1) \leq f_k(X_2)$, $k \in [1 : |F|]$ имеется, хотя бы одно строгое. Вектор X_1 доминирует вектор X_2 , то есть $X_1 \triangleright X_2$, если $F(X_1) \bar{\triangleright} F(X_2)$.

Выделим из множества D_F подмножество точек D_F^* - фронт Парето многоцелевой задачи оптимизации (МЦО-задачи), среди которых нет доминирующих друг друга. Множество $D_X^* \in D_X$, соответствующее множеству D_F^* , называют множеством Парето указанной МЦО-задачи. Таким образом, если $X \in D_X^*$, то $F(X) \in D_F^*$.

Оценка качества Парето-аппроксимации

Обычно алгоритмы Парето-аппроксимации строят на основе эволюционных и, чаще всего, генетических алгоритмов. Принципиальным в этом случае является не использование именно эволюционных алгоритмов, а правила формирования фитнес-функции, обеспечивающей перемещение индивидов популяции, в конечном счете, в направлении множества Парето. Эволюция же этих индивидов может протекать по законам других популяционных алгоритмов.

Архивные множества, содержащие не доминируемые точки F_j^{Θ} и соответствующие им точки X_j^{Θ} ; $j \in [1:|\Theta|]$, обозначаем как Θ^F , Θ^X . Однако для простоты записи в обозначениях архивного множества Θ^F решений, недоминируемых в пространстве $\{F\}$, опустим индекс F и для элементов множества Θ используем обозначения вида F_j , $j \in [1:|\Theta|]$. Большинство популяционных алгоритмов Парето-аппроксимации использует итерационное уточнение множеств точек в архивах Θ^F , Θ^X . Если при этом на итерации t появляется новая точка F_i , доминирующая некоторые точки из архива Θ^F , то все доминируемые точки, а также соответствующие им точки из архива Θ^X , удаляем. При удовлетворении некоторого критерия останова, текущее содержимое архивов Θ^F , Θ^X полагаем искомой аппроксимацией фронта D_F^* и множества Парето D_X^* соответственно.

Оценку эффективности алгоритмов Парето-аппроксимации естественно производить на основе качества полученной Парето-аппроксимации и суммарных вычислительных затрат. Таким образом, задача оценки эффективности алгоритмов Парето-аппроксимации сама является, по меньшей мере, двухцелевой. В содержательных терминах качество Парето-аппроксимации может быть оценено с помощью следующих характеристик:

- близость найденных решений к точному множеству Парето рассматриваемой МЦО-задачи;
- равномерность распределения решений в полученной Парето-аппроксимации;
- мощность найденного множества решений.

В практически значимых МЦО-задачах, в которых, как правило, используется более двух целевых функций, визуальный анализ качества соответствующей Парето-аппроксимации затруднен. Поэтому актуальной является задача разработки индикаторов качества Парето-аппроксимации, формализующих указанные характеристики [1]. Индикаторы такого сорта называют унарными (*unary quality indicators*). Качество Парето-аппроксимации также можно оценивать путем сравнения найденных недоминируемых решений с некоторыми другими, например, эталонными решениями. Индикаторы такого сравнения называются бинарными (*binary quality indicators*).

Результаты эксперимента

В работе был рассмотрен ряд унарных и бинарных индикаторов и созданы соответствующие программные модули для их вычисления. Программный комплекс представляет собой набор *Web*-страниц, реализующих сценарий оценки качества Парето-аппроксимации, представленной пользователем, а также заголовочных файлов и файлов с исходным кодом на языке *C++* и *Python* для каждого из индикаторов.

Входными данными для программ являются архивные множества Θ_1 и Θ_2 , которые являются результатом работы разных алгоритмов оптимизации. Дополнительно, для программ, реализующих вычисления значений бинарных индикаторов на основе функций предпочтения ЛПР, входными данными являются ограничения на компоненты вектора вещественных параметров \wedge , а также идеальное решение $F^{**} = (f_1^*, f_2^*, \dots, f_{|F|}^*)$ [1].

В качестве примера на рисунке 1 представлены тестовые данные для программы, вычисляющей значение индикатора гиперобъема разности покрытий $I_{HW}(\Theta_1, \Theta_2)$ [1].

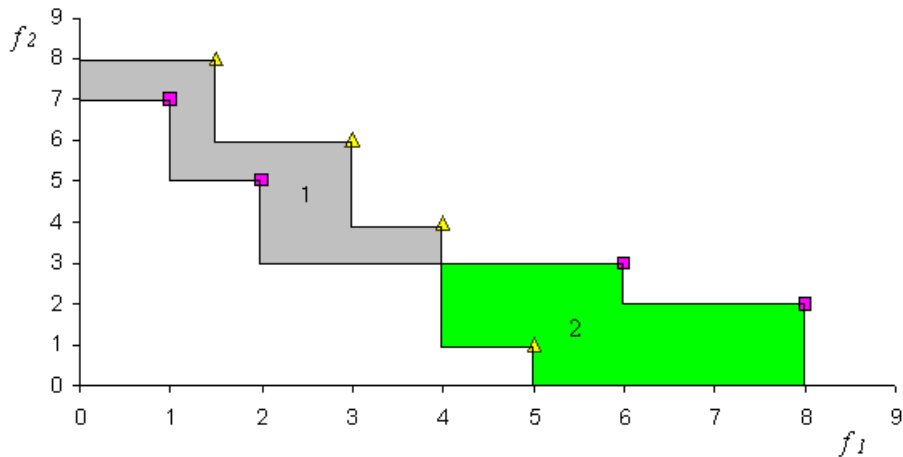


Рисунок 1 – Тестовые данные для программы, вычисляющей значения критерия гиперобъема разности покрытий: \blacktriangle - множество Θ_1 ; \blacksquare - множество Θ_2 ; область 1 - $\delta(\Theta_1, \Theta_2)$; область 2 - $\delta(\Theta_2, \Theta_1)$

На рисунке 2 представлен пример тестовых данных для программ, реализующих вычисление бинарных индикаторов на основе функций предпочтения. Для большей наглядности результатов, значения функции плотности вероятности $\xi(\Lambda)$ [1] на всей области определения были приняты одинаковыми.

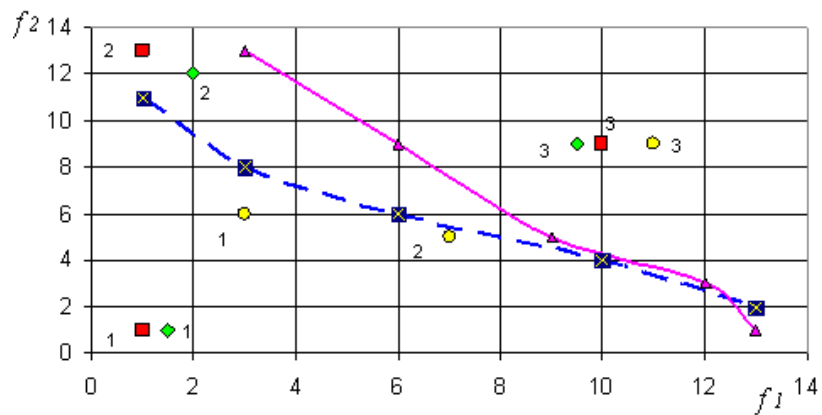


Рисунок 1 – Тестовые данные для программ, реализующих вычисление индикаторов на основе функций предпочтения: \boxtimes - множество Θ_1 ; \blacktriangle - множество Θ_2 ; \bullet - идеальные решения для оценки по критерию $I_{R_1}(\Theta_*, \Theta)$; \blacksquare - идеальные решения для оценки по критерию $I_{R_2}(\Theta_*, \Theta)$; \blacklozenge - идеальные решения для оценки по критерию $I_{R_3}(\Theta_*, \Theta)$

Для данного программного модуля входными данными являются множества Θ_1 и Θ_2 . В данном примере $I_{HV}(\Theta_2, \Theta_1) = 9,0$, $I_{HV}(\Theta_1, \Theta_2) = 7,0$, и поэтому аппроксимация Θ_2 является лучшей.

Заключение

В завершении следует отметить, что всесторонне оценить качество полученной Парето-аппроксимации можно только с помощью широкого набора рассмотренных и иных индикаторов. Поскольку эта задача является многоцелевой ее, как и исходную МЦО-задачу, следует отнести к плохоформализованным задачам, решение которых является принципиально субъективным, полностью определяемым лицом, принимающим решения (ЛПР). В этом случае уместным и обоснованным является создание программного комплекса, реализующего анализ качества аппроксимации с помощью различных критериев и дающего возможность ЛПР понять особенности построенного множества достижимости, а также особенности множества и фронта Парето МЦО-задачи, что, в конечном счете, обеспечивает формирование адекватных представлений ЛПР об этой задаче.

Литература

1. Zitzler E., Deb K., Thiele L. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results // *Evolutionary Computation*, 2000, Vol. 8(2), pp. 173-195.

РАЗРАБОТКА И ИССЛЕДОВАНИЕ ГИБРИДНОГО МУЛЬТИМЕТЕЕВОГО АЛГОРИТМА ЭВОЛЮЦИИ РАЗУМА

Сахаров М.К.

Научный руководитель: д.ф.-м.н., профессор, Карпенко А.П.

МГТУ им. Н.Э.Баумана, кафедра РК-6, Москва, Россия

DEVELOPMENT AND INVESTIGATION OF THE HYBRID MULTI-MEMETIC MIND EVOLUTIONARY COMPUTATION ALGORITHM

Sakharov M.K.

Supervisor: Dr., Prof., Karpenko A.P.

BMSTU, Moscow, Russia

Аннотация

В статье описан разработанный модифицированный алгоритм эволюции разума и представлены результаты исследования его эффективности. На основании анализа результатов исследования предложена гибридизация алгоритма с меметическими алгоритмами. В работе также представлены результаты исследования эффективности гибридного мультиметеевого алгоритма эволюции разума в сравнении с эффективностью исходного алгоритма.

Abstract

This article describes developed modified mind evolutionary computation algorithm and provides results of its efficiency investigation. Hybridization of the algorithm with memetic algorithms was proposed on the basis of study's output. Results of hybrid multi-memetic mind evolutionary computation algorithm's efficiency investigation were also presented in this paper. Comparison of two algorithms was carried out.

Введение

Целью данного исследования является разработка программного комплекса для решения практически значимых задач многомерной оптимизации. Специфика этих задач заключается в отсутствии априорной информации и высокой вычислительной сложности целевой функции. В таких условиях целесообразным является использование адаптивных эвристических алгоритмов поисковой оптимизации, позволяющих с высокой вероятностью локализовывать окрестность глобального оптимума, а также снизить затраты вычислительных ресурсов по сравнению с точными алгоритмами.

В работе выполнены программная реализация модифицированного алгоритма эволюции разума (*МЕС*) [1], а также исследование эффективности этого алгоритма. По результатам исследования алгоритма выявлены его недостатки при оптимизации тестовых функций высокой размерности. На этой основе в работе предложена гибридизация алгоритма *МЕС* с меметическими алгоритмами. В статье представлены результаты исследования эффективности гибридного алгоритма *МЕС* в сравнении с эффективностью исходного алгоритма.

1 Постановка задачи

Рассматриваем детерминированную задачу глобальной условной оптимизации

$$\min_{X \in D} \Phi(X) = \Phi(X^*) = \Phi^*, \quad (1)$$

где $D \subset \mathbb{R}^n$ – область допустимых значений, X^* – искомое оптимальное значение вектора варьируемых параметров, $\Phi(X^*) = \Phi^*$ – соответствующее оптимальное значение целевой функции $\Phi(X)$. Полагаем, что область D представляет собой гиперпараллелепипед

$$D = \{X | x_i^{\min} \leq x_i \leq x_i^{\max}, i \in [1; n]\}. \quad (2)$$

2 Модифицированный алгоритм эволюции разума

Концепция алгоритма эволюции разума предложена в 1998 году в работе С. Ченгай (*S. Chengyi*) и других авторов. Алгоритм *MEC* моделирует скорее некоторые аспекты поведения человека в обществе, чем, как можно было бы предположить, работу человеческого мозга [1].

Исходная версия алгоритма *MEC*, названная авторами простым алгоритмом эволюции разума, построена на основе операций инициализации групп, локальных состязаний (*similar-taxis*) и диссимилиации (*dissimilation*). Эта версия была дополнена нами операцией декомпозиции области поиска перед операцией инициализации (модифицированный *MEC*, *ММЕС*). Такой подход предотвращает преждевременную сходимость вычислительного процесса. Общую схему разработанного алгоритма представляет следующая последовательность шагов.

1) Разбиваем область поиска D на подобласти путем деления интервала $[x_1^{\min}, x_1^{\max}]$ на η равных подынтервалов. Величина η – свободный параметр алгоритма.

2) В каждой подобласти генерируем заданное число групп S_η (еще один свободный параметр алгоритма). Выполняем операцию инициализации каждой группы – генерируем индивидов популяции вокруг главного индивида группы.

3) Проводим операции локальных состязаний в каждой группе. Генерируем вокруг индивида-победителя новых индивидов, определяем для них значение целевой функции и на основании полученных результатов определяем нового победителя. Операция локальных состязаний реализует локальный поиск максимума целевой функции каждой из групп.

4) Проводим операцию диссимилиации. Сортируем группы согласно результатам их лучших индивидов, удаляем отстающие группы и вместо них генерируем новые с помощью операции инициализации. Операция диссимилиации управляет глобальным поиском.

5) Проверяем условия останова – превышено ли число итераций стагнации λ_{stop} , либо максимально допустимое число итераций λ_{max} . Если да, то текущий победитель из всех групп объявляется искомым оптимальным решением X^* и вычисления заканчиваются, иначе переходим к пункту 3.

3 Исследование эффективности алгоритма *ММЕС*

Для исследования эффективности данного алгоритма было выбрано четыре тестовых функции: две многоэкстремальные функции (пятимерные функции Растригина и Гриванка) и две овражные функции (пятимерные функции Розенброка и Захарова). Область поиска представляет собой гиперкуб с ребром $l = 20$ и с центром в начале координат.

Число подобластей декомпозиции принято равным $\eta = 10$, остальные свободные параметры имеют следующие значения: среднее квадратичное отклонение при генерации популяции $\sigma = 1,2$, максимально допустимое число итераций $\lambda_{max} = 1000$, число итераций стагнации $\lambda_{stop} = 100$.

Общее число групп варьируем, изменяя число групп S_η в каждой из подобластей. Вторым варьируемым параметром является размер популяции каждой группы $|S|$. Число индивидов в каждой группе увеличиваем, если не достигается заданная точность локализации глобального минимума функции. Поскольку эффективность алгоритма существенно зависит от случайного начального расположения индивидов, используем метод мультистарта с числом стартов, равным 100.

На рис. 1 представлены средние по мультистартам оптимальные значения целевой функции Φ^* при следующих значениях параметров: размер каждой группы $|S| = 20$; значение параметра S_η меняем от двух до восьми с шагом, равным двум, так что общее число групп $L = 20, 40, 60$ и 80 .

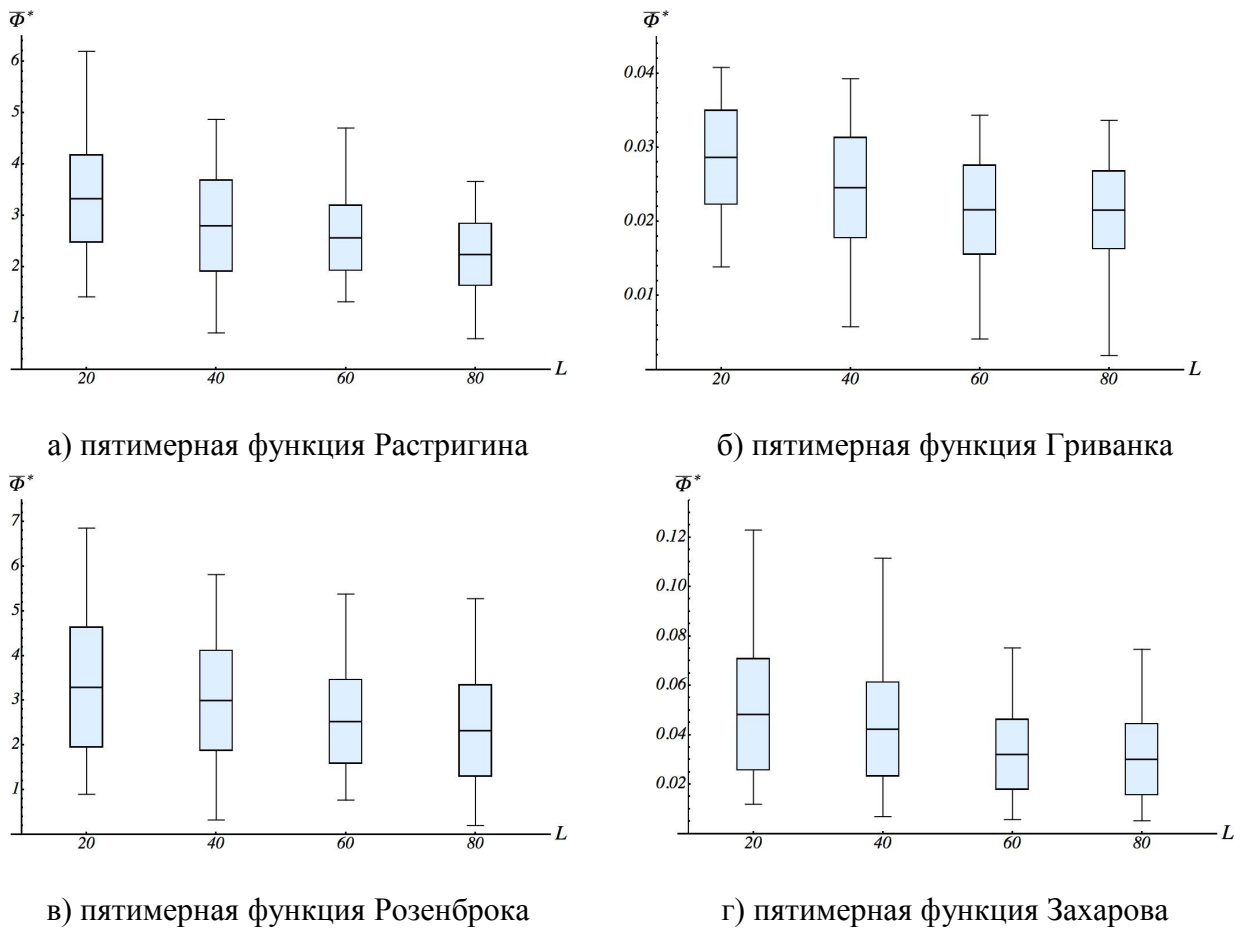


Рисунок 1 – Величина $\bar{\Phi}^*$ в функции числа групп L

В таблице 1 представлены число обращений к целевой функции и среднее число итераций алгоритма *ММЕС* для всех рассмотренных тестовых функций.

Таблица 1 – Вычислительные затраты алгоритма *ММЕС*: $|S| = 20$, $L = 80$

Параметр	Функция Растригина	Функция Гриванка	Функция Розенброка	Функция Захарова
Среднее число обращений к целевой функции за 100 запусков	3,6e+05	2,6e+05	3,4e+05	2,9e+05
Среднее число итераций, \bar{A}	227	188	203	184

Полученные результаты показывают, что предложенный алгоритм плохо справляется с задачей локализации глобального минимума в функциях высокой размерности.

4 Гибридный алгоритм эволюции разума

Меметические алгоритмы (*memetic algorithms, MA*) определяют как гибридные популяционные метаэвристические алгоритмы поисковой оптимизации, основанные на концепции мема и нео-дарвиновском принципе эволюции [2]. В контексте меметических алгоритмов мем является реализацией какого-либо алгоритма локальной оптимизации, уточняющего текущие решения на каждой итерации, либо через некоторое число итераций.

В данной работе рассматриваем мультимемеевую модификацию *MA* (*multi-memes*). Для всех агентов популяций на каждой итерации используем различные мемы в зависимости от их эффективности в данных условиях. Основной задачей конструирования таких алгоритмов является задача выбора целесообразной стратегии использования того или иного

мема из роя доступных мемов. Эти стратегии иногда называют гиперэвристиками (*hyperheuristic*) [2].

Алгоритм *ММЕС* был модифицирован нами таким образом, что на стадии локальных состязаний появился этап локального улучшения решения с помощью следующих мемов:

- алгоритм Солиса Ветса (два набора значений свободных параметров) [3];
- алгоритм Нелдера-Мида.

В качестве гиперэвристики был выбран метод, основанный на жадном принципе. На каждой итерации выбираем алгоритм, эффективность которого оказалась лучше, чем у остальных. Критерием эффективности является наилучшее найденное значение целевой функции $\Phi(X)$. Разработанный алгоритм назовем гибридным мультимемеевым алгоритмом эволюции разума (*НМЕС*).

5 Исследование эффективности алгоритма *НМЕС*

Использованы следующие значения свободных параметров: размер каждой группы $|S| = 5$, число групп $L = 20$. Остальные параметры оставим без изменений. Результаты эксперимента для всех рассматриваемых тестовых функций представлены в таблице 2.

Таблица 2 – К оценке эффективности алгоритма *НМЕС*: $|S| = 5$, $L = 20$

Параметр	Функция Растргина	Функция Гриванка	Функция Розенброка	Функция Захарова
Оценка математического ожидания ошибки локализации глобального минимума, $\bar{\epsilon}$	2,99e-02	1,75e-03	3,46e-07	2,88e-07
Минимальная ошибка локализации за 100 запусков	1,94e-07	1,55e-08	1,80e-08	6,80e-08
Оценка среднего квадратичного отклонения	0.17	3,59e-03	1,33e-07	1,14e-07
Число удачных запусков, $\bar{\epsilon}_1 \leq 10^{-1}$	97	98	100	100
Среднее число обращений к целевой функции за 100 запусков	1,1e+06	1,5e+06	1,06e+06	1,17e+06
Среднее число итераций, \bar{A}	118	145	101	100

Из анализа полученных результатов видно, что гибридный алгоритм успешно справился со всеми тестовыми функциями. При значениях свободных параметров, обеспечивающих минимальные вычислительные затраты, вероятность локализации глобального минимума составила 99 %.

Заключение

Сравнение двух алгоритмов демонстрирует превосходство алгоритма *НМЕС* над алгоритмом *ММЕС* с точки зрения вероятности локализации глобального оптимума. Точность локализации оптимума многоэкстремальных функций различается на порядок, а в случае с овражными функциями – на пять порядков.

Существенным недостатком гибридного алгоритма являются высокие вычислительные затраты – число обращений к целевой функции в десять раз больше чем у алгоритма *ММЕС*.

Литература

1. Jing Jie, Jianchao Zeng. Improved Mind Evolutionary Computation for Optimizations. Proceedings of 5th World Congress on Intelligent Control and Automation, Hang Zhou, China. 2004. pp. 2200-2204.
2. Y.S. Ong, M.H. Lim, N. Zhu and K. W. Wong, "Classification of adaptive memetic algorithms: A comparative study". IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 2006. pp. 141-152.
3. F. J. Solis, R. J.-B. Wets, Minimization by Random Search Techniques, Mathematics of Operations Research, Vol. 6, 1981. pp. 19-30.

АВТОМАТИЗИРОВАННОЕ ПРОЕКТИРОВАНИЕ ХИРУРГИЧЕСКИХ ШАБЛОНОВ В СТОМАТОЛОГИИ

Кильчик А.В., Кожевников Д.Ю., Савелов А.С., Сорокин А.С., Шмакова Н.А.
Научный руководитель: к.т.н., доцент Жук Д.М.
МГТУ им. Н.Э.Баумана, кафедра РК6, Москва, Россия

COMPUTER-AIDED DESIGN OF SURGICAL TEMPLATES IN STOMATOLOGY

Kilchik A.V., Kozhevnikov D.Y., Savelov A.S., Sorokin A.S., Shmakova N.A.
Supervisor: Ph.D. Zhuk D.M.
MSTU, Moscow, Russia

Аннотация

[В работе рассматриваются вопросы автоматизации процесса проектирования хирургических шаблонов в стоматологии. Описаны этапы проведения операции по установке зубных имплантатов. Обоснована необходимость применения хирургических шаблонов в процессе указанной операции. Выполнен обзор существующих подходов к автоматизации процесса проектирования хирургических шаблонов в стоматологии. Предложена новая организация указанного процесса проектирования.](#)

Abstract

The paper discusses a class of problems with surgical templates design automation in stomatology. It also describes the stages of teeth templates setting operation. The paper shows the necessity of surgical templates application during this process. The review of existing methods of the design automation process is provided. Also the paper introduces a new organization of the process.

Введение

В работе [1] была предложена новая структура организации реконструктивно-восстановительного медицинского технологического процесса для челюстно-лицевой хирургии. При этом одним из важнейших этапов восстановления зубного ряда является проектирование хирургической операции установки имплантатов для зубных протезов и соответствующей технологической оснастки. Перед врачом стоит задача планирования хирургического вмешательства и собственно его проведения. В процессе операции производится разрез слизистой оболочки и формирование отверстий. Это достигается сверлением, что во многих случаях является нетривиальной операцией из-за сложной поверхности челюсти, состояния костной ткани, особенностей внутреннего строения челюсти и др. факторов. В подготовленное резьбовое отверстие с помощью специального устройства вворачивается имплантат [2].

Качество технической реализации установки имплантата - один из основных факторов, влияющих на долговечность и успешность использования пациентом зубного протеза.

На помощь хирургу приходят специальные - хирургические шаблоны (ХШ), которые позволяют точно позиционировать инструменты непосредственно во время операции. Шаблон представляет собой элемент, внутренняя поверхность которого полностью повторяет внешнюю поверхность челюсти, и, таким образом, идеально ложится на неё. В местах под сверление располагаются сквозные направляющие отверстия, которые обеспечивают точность позиционирования сверла и минимизируют травмирование тканей пациента.

1 Структура процесса проектирования хирургических шаблонов

На основе анализа последовательности работы программных продуктов для проектирования ХШ (*SimPlant*, *Amira*, *3ds Max*) были выделены следующие обобщенные этапы выполнения проекта.

1) *Диагностика*. Пациент проходит томографическое исследование (КТ, МРТ). На выходе данного этапа формируется томограмма пациента в медицинском формате информационных данных *DICOM*.

2) *Создание локальной анатомической 3D модели пациента*. Специальная программа создает 3D модель челюсти пациента (полигональную либо *NURBS* модель). Полученная модель должна быть достаточно точной, чтобы отклонения от геометрии реальной челюсти не снизили точность позиционирования ХШ.

3) *Размещение имплантатов*. Данный этап трудно поддается автоматизации, поскольку задача размещения имплантатов на поверхности челюсти является плохоформализованной задачей, успех выполнения которой напрямую зависит от квалификации эксперта в данной области. Поэтому основной задачей на данном этапе является предоставление удобного, интуитивно понятного инструмента для эксперта (пародонтолога, хирурга).

4) *Формирование 3D модели шаблона*. 3D модель шаблона формируется отдельной программой, которая использует модель челюсти пациента и результаты этапа размещения имплантатов.

5) *Печать на 3D принтере модели шаблона*. Модель предварительно сохраняется в формате данных *STL*. После печати в шаблон устанавливаются тонкостенные направляющие металлические трубки.

2 Используемые подходы к проектированию хирургических шаблонов

Подходы к автоматизации процесса проектирования ХШ для имплантатов можно разделить на следующие две категории в зависимости от используемых программных продуктов:

- использование специализированной *CAS* системы *SimPlant* или аналогичной ей;
- использование набора программных продуктов общего назначения, например, *AMIRA* (или *Mimics*) и *3ds Max*.

Автоматизация с использованием специализированной CAS системы SimPlant. Процесс проектирования в этом случае включает следующие этапы.

1) Использование программы *Mimics* для создания полигональной 3D модели челюсти пациента по результатам томографического исследования.

2) Размещение экспертом имплантатов по внешней поверхности челюсти. Имплантаты выбираются из существующей базы имплантатов. При этом в модель челюсти интегрированы снимки томограммы, что позволяет эксперту контролировать правильность расположения имплантатов.

3) Создание специальной программой полигональной 3D модели шаблона.

4) Печать на 3D принтере модели шаблона и установка направляющих металлических трубок.

Такая организация процесса проектирования ХШ обладает следующими достоинствами:

- достаточно высокий уровень автоматизации;
- точность модели челюсти, достаточная для точного позиционирования ХШ;
- обширная база параметров имплантатов разных изготовителей;
- визуальный контроль текущего положения имплантатов на 3D модели с интегрированными снимками томограммы.

Для нее также характерны следующие недостатки:

- сложность освоения *CAS* системы *SimPlant* для стоматологов;
- необходимость предварительного обучения эксперта для работы с программным комплексом;
- высокая стоимость создания геометрической модели и *3D* печати полученной модели ХШ.

Автоматизация при помощи программных продуктов AMIRA (или Mimics) и 3ds Max. При данном подходе к проектированию вначале создается *3D* модель челюсти пациента с помощью программных продуктов *AMIRA* или *Mimics* и далее она редактируется в программной системе *3ds Max*. Процесс проектирования в этом случае состоит из следующих этапов.

1) Создание полигональной *3D* модели челюсти пациента по результатам томографического исследования в программе *AMIRA (Mimics)*. В челюстной костной ткани любого человека имеется множество полостей и канальцев, поэтому с топологической точки зрения, оболочка модели будет иметь связность, много большую единицы [3].

2) Редактирование полигональной модели челюсти в программе *3ds Max*. Цель данного этапа - превратить оболочку модели в простейшую (со связностью, равной единице) [3]. Это достигается путем ручного добавления полигонов в полигональную сетку модели.

3) Получение по откорректированной модели челюсти, сопряженной с ней полигональной модели ХШ. Сопряженной с данной моделью будем называть модель, внутренняя поверхность которой совпадает с внешней поверхностью данной модели.

Если не выполнен этап редактирования модели челюсти, то внутренняя поверхность модели, сопряженной с данной, будет иметь характерные неоднородности. Выполненный по такой модели ХШ непригоден для проведения операции из-за особенностей полученной геометрической модели.

4) Размещение имплантатов на модели шаблона.

5) Печать на *3D* принтере модели шаблона и установка направляющих металлических трубок.

Эта организация процесса проектирования ХШ обладает следующими достоинствами:

- универсальность и распространенность ПО;
- невысокая стоимость создания и *3D* печати полученной модели ХШ.

Для нее также характерны следующие недостатки:

- проектирование имеет высокую трудоемкость, поскольку производится практически полностью вручную, за исключением этапа создания модели челюсти;
- необходим специалист достаточно высокой квалификации в области компьютерной графики.

3 Предлагаемая организация процесса проектирования хирургических шаблонов

С учетом достоинств и недостатков вышеупомянутых программных средств и подходов, предлагается следующая организация процесса автоматизированного проектирования ХШ.

1) *Томографическое исследование.* Результатом данного этапа является томограмма интересующей области пациента в формате *DICOM*.

2) *Получение модели челюсти пациента по результатам томографического исследования.*

3) *Автоматическая корректировка модели.* На данном этапе специальная программа добавляет элементы в полигональную сетку модели таким образом, чтобы оболочка, описывающая дополненную модель, имела связность, равную единице.

4) *Построение предварительной модели шаблона.* Разработанная авторами, программа синтезирует замкнутую модель (связность оболочки равна нулю) [3], сопряженную с моделью челюсти.

5) *Размещение имплантатов.* На данной стадии эксперт, используя специальную программу, производит расстановку имплантатов, контролируя их положение относительно моделей челюсти и шаблона. При этом в модель челюсти интегрированы снимки томограммы. Основное требование к данной программе - интерфейс пользователя, интуитивно понятный медицинскому эксперту.

6) *Формирование итоговой 3D модели шаблона.* Программа автоматически строит итоговую модель ХШ, готовую к печати на 3D принтере.

Получение модели челюсти пациента по результатам томографического исследования. Предлагается получать воксельную геометрическую модель челюсти с использованием алгоритмов обработки изображений, реализованных в библиотеке обработки изображений *ITK*. Полученная воксельная модель преобразуется в полигональную.

Автоматическая корректировка модели. Предлагается следующий эвристический подход к корректировке модели.

Выполняется преобразование модели в набор сечений при помощи библиотеки компьютерной графики *VTK*, а затем в каждом сечении выделяется замкнутый контур. Для этого используются алгоритм, основанный на применении *A*-множеств [4]. По полученным плоским контурам восстанавливается откорректированная поверхность челюсти (рис. 2).

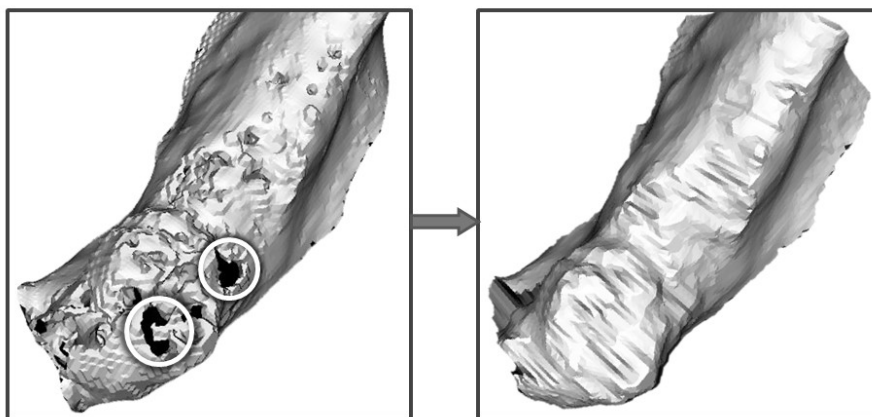


Рисунок 2 - Слева: участок исходной полигональной модели челюсти; справа: участок геометрической модели челюсти после корректировки поверхности

Получение предварительной модели шаблона. Предварительная модель шаблона строится по двум исходным поверхностям: первой - сопряженной с моделью челюсти (откорректированной на предыдущем этапе) и ее смещенной копии.

Размещение имплантатов. Специальная программа размещения имплантатов должна уметь работать с моделью челюсти, предварительной моделью шаблона и со снимками томограммы в любом сечении челюсти. При этом разнообразие виджетов *VTK* позволяет обеспечить необходимую простоту интерфейса.

Формирование 3D модели шаблона. Данный этап использует результаты предыдущих, а именно модель челюсти, предварительную модель шаблона и схему размещения виртуальных имплантатов. Посредством логических операций в трехмерном пространстве формируется итоговая модель шаблона, готовая к печати на 3D принтере.

Заключение

В результате проделанной работы выполнен обзор подходов к автоматизации процесса проектирования ХШ, предложен новый вариант организации процесса проектирования. К его достоинствам можно отнести интуитивно понятный медицинскому эксперту (не имеющему специальных навыков в области информационных технологий) интерфейс, а также автоматическое выполнение всех стадий формирования шаблона, кроме расстановки имплантатов. Дальнейшее развитие разработанной организации процесса проектирования ХШ авторы видят в применении *NURBS* геометрических моделей объектов.

Литература

1. Жук Д.М., Перфильев С.А. CAS системы – системы автоматизированного проектирования в хирургии // Наука и образование: электронное научно-техническое издание, 3, 2011, (<http://www.technomag.edu.ru/doc/168885.html>).
2. Хирургические шаблоны в имплантологии, (<http://www.implantologi.ru/19.html>).
3. Голованов Н.Н. Геометрическое моделирование.- М.: Физматлит, 2002.- 472 с.
4. Fadili M., Melkemi M., ElMoataz A. Non-convex onion-peeling using a shape hull algorithm // Pattern Recognition Letters, 2004, Vol. 25, pp. 1577-1585.
5. Choi B., Park S. A pair-wise offset algorithm for 2D point-sequence curve // Computer-Aided Design, 1999, Vol. 31, pp. 735-745.

АППРОКСИМАЦИЯ ЦЕЛЕВЫХ ФУНКЦИЙ В *AWS*-МЕТОДЕ ПОСТРОЕНИЯ МНОЖЕСТВА ПАРЕТО

Савелов А.С.

Научный руководитель: д.ф.-м.н., профессор, Карпенко А.П.

МГТУ им. Н.Э.Баумана, кафедра РК6, Москва, Россия

OBJECTIVE FUNCTIONS APPROXIMATION IN *AWS*-METHOD OF CONSTRUCTION OF THE PARETO SET

Savelov A.S.

Supervisor: professor Karpenko A.P.

MSTU, Moscow, Russia

Аннотация

Рассматривается внедрение метамоделей целевых функций в адаптивный метод взвешенных сумм (*AWS*-метод) построения множества Парето. Целью работы является сравнительный анализ эффективности квадратичной аппроксимации критериев и аппроксимации нейронными сетями на основе радиальных базисных функций.

Abstract

This paper discusses integration of objective functions metamodels in adaptive weighted sum method (*AWS*-method) of construction of the Pareto set. Work purpose is efficiency comparative analysis of quadratic objective functions approximation and approximation by neural nets based on radial-basis functions.

Введение

Относительно новый и быстро развивающийся класс методов решения задачи многокритериальной оптимизации (МКО-задачи) образуют методы Парето-аппроксимации, предполагающие предварительное построение некоторой конечномерной аппроксимации множества, а тем самым, и фронта Парето. Известно большое число популяционных и непопуляционных методов построения Парето-аппроксимации [1]. Метод адаптивных взвешенных сумм (*Adaptive Weighted Sum method, AWS-method*) [2] – один из современных и хорошо зарекомендовавших себя [3] непопуляционных методов Парето аппроксимации.

Для сокращения затрат на вычисление значений критериальных функций метод *AWS* использует метамоделю этих функций. Поскольку в практически значимых задачах вычислительная сложность критериальных функций обычно является высокой, это обстоятельство позволяет значительно сократить вычислительные затраты.

В работе проводится сравнительный анализ эффективности квадратичной аппроксимации критериев и аппроксимации нейронными сетями на основе радиальных базисных функций.

1 Постановка задачи

Множеством допустимых значений вектора варьируемых параметров X является ограниченное и замкнутое множество $D_X \subset \{X\} = R^{|X|}$. Значения целевой вектор-функции $F(X) = (f_1(X), f_2(X), \dots, f_{|F|}(X))$ принадлежат целевому пространству $\{F\} = R^{|F|}$. Ставится задача минимизировать во множестве D_X каждую из функций $f_1(X), f_2(X), \dots, f_{|F|}(X)$:

$$\min_{X \in D_X} F(X) = F(X^*) = F^* \quad (1)$$

Здесь и далее запись вида $|A|$, где A - вектор или некоторое счетное множество, означает размерность этого вектора или мощность множества соответственно. Полагаем, что частные

целевые функции тем или иным образом нормализованы, так что $f_i(X) \in [0; 1]$ для любого $X \in D_X$; $i \in [1:|F|]$.

Множество и фронт Парето обозначаем $D_X^* \subset D_X$, $D_F^* \subset D_F$, а их конечномерные аппроксимации Θ^X , Θ^F соответственно. Последние множества называем архивными.

2 Схема метода AWS

Метод AWS включает в себя три следующие основные процедуры:

- определение центральной точки;
- формирование метамоделей частных критериев оптимальности;
- решение полученных оптимизационных задач.

Определение центральной точки. На этапе инициализации центральную точку X_C^0 выбираем случайным образом в области D_X . На этом же этапе определяем свободные параметры алгоритма: δ - начальный радиус области доверия (*trust region radius*); $\rho \in (0; 1)$ - коэффициент сужения этой области; δ_{\min} - минимальная величина радиуса области. На итерации $(t+1)$ центральную точку X_C' отыскиваем среди точек текущей Парето-аппроксимации $\Theta^X = \Theta^X(t)$, построенной на предыдущей итерации t .

Элементы архивных множеств Θ^X , Θ^F сортируем по возрастанию первого частного критерия $f_1(X)$ и представляем в виде линейных списков с прежними наименованиями. Расстояние d_j архивной точки F_j^Θ до ближайших к ней в списке Θ^F точек определяем формулой

$$d_j = \|F_{j-1}^\Theta - F_j^\Theta\|_E + \|F_j^\Theta - F_{j+1}^\Theta\|_E, \quad j \in [1:|\Theta|], \quad (2)$$

где $\|\bullet\|_E$ - евклидова норма.

За центральную точку принимаем точку, во-первых, наиболее удаленную от других точек множества Θ^X в смысле расстояния (2), и, во-вторых, не использованную на предшествующих итерациях.

Формирование метамоделей. В каноническом методе AWS метамоделей представляют собой квадратичные аппроксимации $m_1'(X)$, $m_2'(X)$ функций $f_1(X)$, $f_2(X)$ в окрестности точки X_C' :

$$\begin{aligned} m_1'(X) &= f_1(X_C') + \nabla^T f_1(X_C') (X - X_C') + 1/2 (X - X_C')^T H_1(X_C') (X - X_C'); \\ m_2'(X) &= f_2(X_C') + \nabla^T f_2(X_C') (X - X_C') + 1/2 (X - X_C')^T H_2(X_C') (X - X_C'). \end{aligned}$$

Здесь $H_1(X_C')$, $H_2(X_C')$ - матрицы Гессе функций $f_1(X)$, $f_2(X)$ в точке X_C' .

Если $|\Theta| > 2$, то дополнительно строят метамоделей

$$m_p'(X) = \lambda_1^p m_1'(X) + \lambda_2^p m_2'(X), \quad m_q'(X) = \lambda_1^q m_1'(X) + \lambda_2^q m_2'(X),$$

а если $|\Theta| = 2$ или $|\Theta| = 1$ - еще и метамоделей $m_p'(X) = \lambda_1^p m_1'(X) + \lambda_2^p m_2'(X)$. Правила определения весовых множителей $(\lambda_1^p, \lambda_2^p) = \Lambda^p$, $(\lambda_1^q, \lambda_2^q) = \Lambda^q$ представлены в работе [2].

Решение оптимизационных задач. Процедура предполагает решение задач

$$\min_{X \in D_C'} m_1'(X) = m_1'(\hat{X}_1'), \quad \min_{X \in D_C'} m_2'(X) = m_2'(\hat{X}_2'), \quad (3)$$

где текущую область доверия (*trust region*) D_C' определяет формула

$$D_C' = \{X \mid X \in D_X, \|X - X_C'\| \leq \delta\}.$$

Если $|\Theta| > 2$, то решения \hat{X}'_1, \hat{X}'_2 задач (3) позволяют отыскать приближенно оптимальные по Парето точки \hat{X}'_p, \hat{X}'_q , принадлежащие области доверия D'_c , путем решения задач

$$\min_{X \in D'_c} m'_p(X) = m'_p(\hat{X}'_p), \quad \min_{X \in D'_c} m'_q(X) = m'_q(\hat{X}'_q). \quad (4)$$

В процессе итераций текущий радиус области доверия уменьшаем по правилу $\delta = \rho \delta$ до достижения минимально допустимой его величины δ_{\min} . Новое состояние архивного множества Θ^X получаем путем добавления в него точек $\hat{X}'_1, \hat{X}'_2, \hat{X}'_p, \hat{X}'_q$ и исключения из полученного набора доминируемых решений. Аналогично, множество Θ^F образуем путем добавления в него точек $F(\hat{X}'_1), F(\hat{X}'_2), F(\hat{X}'_p), F(\hat{X}'_q)$.

3 Квадратичная и нейросетевая аппроксимации целевых функций

Для построения квадратичной аппроксимации целевых функций $f_i(X)$, $i \in [1:2]$ используем центральный композиционный план (ЦКП) [4] с центром в точке X'_c . С целью упрощения вычислений оценки функции отклика (МНК-оценки), «звездное» плечо выбираем таким образом, чтобы ЦКП оказался ортогональным. В качестве ядра ЦКП выбран план полного факторного эксперимента ПФЭ 2^2 , поскольку для этого требуется произвести небольшое число испытаний функции $N_0 = 4$. Таким образом, необходимое число испытаний функции для построения одной метамоделей равно $N = 2^2 + 2 \cdot 2 + 1 = 9$. Учитывая, что решение оптимизационных задач (3) и (4), составленных на базе полученных метамоделей $m'_1(X)$ и $m'_2(X)$, производится методом штрафных функций в совокупности с методом Нелдера-Мида, данные вычислительные расходы на создание метамоделей ожидаются приемлемыми.

Для построения нейросетевой аппроксимации целевых функций использована нейронная сеть на основе радиальных базисных функций (РБФ) [5]. Число нейронов в сети принято равным числу испытаний в указанном ЦКП. Центры РБФ расположены в испытываемых точках. Обучающую выборку формируем из точек, случайным образом сгенерированных в пределах области доверия. Мощность обучающей выборки D выбираем по формуле $|D| = 2 \cdot L$, где L – размерность скрытого слоя нейросети. Выборка такой мощности не должна привести к избыточным вычислительным расходам на испытания функций и в то же время должна придать нейросети свойство обобщения.

4 Реализация метода и организация экспериментов

Для решения задач условной оптимизации (3), (4) в программной реализации использован метод штрафных функций в совокупности с известным методом Нелдера-Мида локальной безусловной оптимизации. Вместо самих целевых функций $f_1(X)$, $f_2(X)$ использованы их метамоделей согласно описанным выше схемам.

Программа написана на языке программирования C++. Эксперименты выполнены на персональном компьютере в следующей конфигурации: процессор - Intel Pentium Dual-Core T4500, 2,3 ГГц; оперативная память - DDR-3, 3 Гб. Исследование выполнено на трех известных двухкритериальных тестовых задачах многокритериальной оптимизации: ZDT3, ZDT6, ZDT7 [3].

Производительность методов аппроксимации оцениваем на основе числа испытаний целевых функций $n_E(f_i)$, $i = 1, 2$. Качество аппроксимации множества Парето оцениваем с помощью следующих критериев: мощность множества решений (*overall non-dominated*

vector generation) $I_{ONVG}(\Theta) = |\Theta| \rightarrow \max$, где Θ - архив решений; близость найденных решений к точному множеству Парето рассматриваемой МКО-задачи, выражаемая функционалом GD (*generational distance*) [2]; равномерность распределения решений в полученной Парето-аппроксимации, выражаемая функционалом SP (*spacing*) [2]. Результат фиксируется после $\hat{t} = 500$ итераций *AWS*-метода.

5 Результаты экспериментов

Некоторые результаты экспериментов приведены в таблице 1.

Таблица 1 – Результаты экспериментов

Тип аппроксимации	$n_E(f_1)$	$n_E(f_2)$	$I_{ONVG}(\Theta)$	GD	SP
Тестовая задача <i>ZDT3</i>					
Квадратичная	7500	7500	438	0,0007	0,017
Нейросетевая	12000	12000	91	0,005	0,046
Тестовая задача <i>ZDT7</i>					
Квадратичная	7500	7500	133	0,001	0,015
Нейросетевая	12000	12000	388	0,0005	0,009

При решении задач *ZDT3* (несвязный, хотя и выпуклый фронт Парето) и *ZDT7* (непрерывный выпуклый фронт Парето) метод обеспечил приемлемое качество Парето-аппроксимации при любом типе аппроксимации критериев. Сходимость к точному решению и равномерность результатов отличаются незначительно. В плане плотности решения результаты разнятся в зависимости от задачи, но обеспечивают достаточную плотность при обоих типах метамоделей. Однако, для данных задач число испытаний целевых функций оказалось значительно меньше при использовании квадратичной аппроксимации. Причина состоит в дополнительных расходах на обучение нейронной сети.

Самой сложной для метода оказалась задача *ZDT6* (невыпуклый фронт Парето), для которой не удалось получить приемлемую Парето-аппроксимацию ни при одном из типов аппроксимации целевых функций. Причина состоит в сложной структуре множества Парето.

Заключение

Результаты исследования показывают, что оба способа аппроксимации обеспечивают приемлемое качество Парето-аппроксимации в случае выпуклого, хотя, быть может, и несвязного фронта Парето. Для некоторых задач, имеющих вогнутый фронт Парето, аппроксимация критериев приводит к тому, что метод расходится. Из анализа критериев качества Парето-аппроксимации можно сделать вывод о том, что метамоделли на основе квадратичной аппроксимации и на основе РБФ нейросетей показывают одинаковые результаты относительно качества полученного множества Парето. Однако, аппроксимация на основе нейросетей сопровождается значительно большими затратами. Поэтому в случае двумерных критериев предпочтение следует отдать метамоделлям на основе квадратичной аппроксимации.

В продолжение работы планируется исследовать эффективность различных способов аппроксимации для задач большой размерности.

Литература

1. Карпенко А.П., Митина Е.В., Семенихин А.С. Популяционные методы аппроксимации множества Парето в задаче многокритериальной оптимизации. Обзор // Наука и образование: электронное научно-техническое издание, 4, 2012, (<http://www.technomag.edu.ru/doc/363023.html>).
2. Jong-hyun Ryu, Sujin Kim, Hong Wan. Pareto front approximation with adaptive sum method in multiobjective simulation optimization // Proceedings of the 2009 Winter Simulation Conference, pp. 623 – 633.
3. Карпенко А.П., Савелов А.С., Семенихин А.С. Метод адаптивных взвешенных сумм в задаче Парето-аппроксимации // Наука и образование: электронное научно-техническое издание, 6, 2012, (<http://technomag.edu.ru/doc/423283.html>).
4. Асатурян В.И. Теория планирования эксперимента. – М.: Радио и связь, 1983 – 248 с.
5. Хайкин С. Нейронные сети: полный курс.- М.: И.Д. Вильямс, 2006.- 1104 с.

РАЗРАБОТКА ПРОГРАММЫ РЕШЕНИЯ СИСТЕМ ЛИНЕЙНЫХ АЛГЕБРАИЧЕСКИХ УРАВНЕНИЙ В ДЕСЯТИЧНОЙ АРИФМЕТИКЕ

Сахаров М.К.

Научный руководитель к.т.н. доцент Маничев В.Б.

МГТУ им. Н.Э.Баумана, кафедра РК-6, Москва, Россия

SOFTWARE DEVELOPMENT FOR SOLVING SYSTEMS OF LINEAR ALGEBRAIC EQUATIONS WITH A USE OF DECIMAL ARITHMETIC

Sakharov M.K.

Supervisor: Ph.D. assistant professor Manichev V.B.

BMSTU, Moscow, Russia

Аннотация

В статье рассмотрена проблема получения численного решения систем линейных алгебраических уравнений с точностью 15 верных значащих цифр. Для обеспечения требуемой точности было решено использовать десятичную арифметику. Разработано программное средство реализующее предложенный авторами десятичный формат представления числа с плавающей точкой. На основании результатов тестирования разработанной программы были сделаны выводы о целесообразности продолжения разработок в данной области.

Abstract

This article deals with the problem of numerical solution of systems of linear algebraic equations with high precision of 15 correct digits after decimal point. In order to provide such precision decimal arithmetic was used. Software implementing decimal arithmetic was developed and tested. Recommendations on the appropriateness of the follow-up study in this area were given on the basis of test's results.

Введение

Данная работа является продолжением цикла работ посвященных проблемам достоверности и точности решения систем ОДУ в САЕ подсистемах САПР [1-3]. В работе [1] показано, что для гарантии получения корректного и достоверного решения разнообразных систем ОДУ численный метод их решения должен быть *AL*-устойчивым, т.е. абсолютно устойчивыми строго в левой (*Absolute Left*) комплексной полуплоскости. Программная реализация *AL*-устойчивых неявных методов интегрирования сводится к многократному решению соответствующих систем линейных алгебраических уравнений (СЛАУ) на каждом шаге интегрирования, что, как правило, приводит к нескольким тысячам и более обращений к программе-решателю СЛАУ на всем заданном отрезке интегрирования.

В работе [2], на примере решения тестовых и практических задач моделирования технических систем показано, что для получения качественно корректного решения систем ОДУ необходимо на всех шагах интегрирования обеспечивать численное решение соответствующих СЛАУ с точностью в 15 верных значащих цифр для всех элементов вектора решения СЛАУ.

В данной работе предложен метод решения СЛАУ с гарантированной точностью в 15 знаков с использованием десятичной арифметики. Выполнены его программная реализация и тестирование. Результаты тестирования и их анализ также представлены в статье.

1 Постановка задачи

Классическая постановка задачи решения СЛАУ имеет вид

$$A \cdot X = B, \quad (1)$$

где $A = (a_{ij})$ – матрица коэффициентов размерности $n \times n$, такая что $\det(A) \neq 0$; $B = (b_1, \dots, b_n)^T$ – вектор-столбец правых частей; $X = (x_1, \dots, x_n)^T$ – вектор неизвестных.

Необходимо вычислить элементы вектора X с гарантированной точностью в 15 верных значащих цифр.

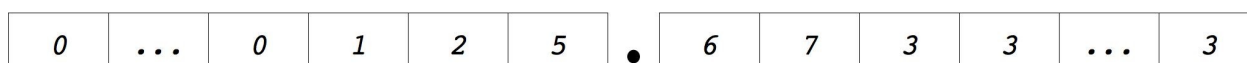
2 Десятичная арифметика

Стандартный тип *double* (или *long double*) для представления вещественных чисел в компиляторах языка Си на обычных персональных компьютерах позволяет работать с числами, мантисса которых не превосходит 16-17-ти десятичных разрядов. В подавляющем большинстве случаев при решении СЛАУ этой длины мантиссы вещественных чисел хватает для получения решений с инженерной точностью. Но при решении разнообразных систем ОДУ (особенно жестких и сверхжестких) иногда необходимо решать СЛАУ, для получения «сверхточного» решения которых, такого количества десятичных разрядов мантиссы не хватает. В этом случае, ошибки при выполнении арифметических операций в процессе численного решения СЛАУ увеличиваются и сильно сказываются на точности конечного результата. Следовательно, для уменьшения ошибок при выполнении арифметических операций надо повысить точность их выполнения.

Для обеспечения требуемой точности было решено использовать десятичную арифметику. Каждое число представляет собой два линейных массива элементов длиной M . Первый массив поэлементно хранит целую часть числа, второй – дробную. На рис. 1 представлен пример хранения числа в данном формате.



а) шаблон представления числа



б) пример представления числа $\frac{10851}{150}$

Рисунок 1 – Форма представления числа с плавающей точкой

Для хранения знака числа используется логическая переменная: если она равна нулю, то число положительное, единице – отрицательное. Для данного формата представления чисел были определены основные арифметические операции с учетом знаков: сложение, вычитание, умножение и деление. Также были определены операции присваивания и инициализации нулем и единицей.

Такое представление обусловлено высоким ростом производительности современных вычислительных систем, а также развитием технологий удаленного доступа, позволяющих производить вычисления удаленно на суперкомпьютерах. Это снижает значимость временных затрат на вычисление, позволяя сосредоточиться на проблеме обеспечения высокой точности.

3 LU – разложение

LU – разложение – это «высокий уровень» алгебраического описания исключения Гаусса. Разложением называется представление матрицы A в виде произведения двух матриц, $A = LU$, где L – нижняя треугольная матрица, а U – верхняя. *LU* – разложение используется для решения систем линейных уравнений. Если известно *LU* – разложение матрицы A , то исходная система может быть записана как

$$L \cdot U \cdot X = B. \quad (2)$$

Эта система может быть решена за два шага. На первом шаге решается система

$$L \cdot Y = B. \quad (3)$$

Поскольку L – нижняя треугольная матрица, эта система решается непосредственно прямой подстановкой. На втором шаге решается система

$$U \cdot X = Y. \quad (4)$$

Так как U – верхняя треугольная матрица, то система (4) решается обратной подстановкой. В данной работе был реализован алгоритм Краута (*Crout*) для LU – разложения [4].

Отметим погрешность, свойственную прямым методам решения СЛАУ. Погрешность вычислений, т.е. выполнения арифметических операций, которая определяется типом используемых данных, нами была устранена благодаря использованию предложенного формата представления числа. Однако, величина погрешности сильно зависит и от конкретных значений элементов матрицы. Каждая матрица имеет так называемое число обусловленности $Cond(A)$, определяемое через нормы прямой и обратной матриц. Чем эта величина больше, тем больше будет погрешность вычисления вектора неизвестных. Число обусловленности определяет также связь погрешности вычисления вектора неизвестных с погрешностями задания матрицы коэффициентов и вектора свободных членов, которые в реальных задачах часто берутся из физических измерений и могут иметь значительную погрешность. Системы с большим значением числа обусловленности (много большим единицы) называются плохо обусловленными. Для них получение достаточно точного решения невозможно или является затруднительным.

С учетом вышесказанного, для тестирования разработанного программного средства были выбраны плохо обусловленные СЛАУ.

4 Тестирование

Первая тестовая задача представляет собой задачу Уилкинсона (*Wilkinson*) [5]:

$$\begin{pmatrix} 0,78 & 0,563 \\ 0,913 & 0,659 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0,217 \\ 0,254 \end{pmatrix} \quad (5)$$

Точное решение этой задачи имеет вид: $x_1 = 1$, $x_2 = -1$.

Вторая тестовая задача – задача Вохминцева [5]:

$$\begin{pmatrix} 0.1190 & -3.470 & 1.780 \times 10^{-7} & -0.5240 & 0.005350 \\ 438\,000.0 & 0.8660 & 3.250 \times 10^{-7} & 0.005660 & 1.940 \times 10^{-6} \\ 5600.0 & 0.0350 & -7.740 \times 10^{-6} & 0.02180 & 14\,100.0 \\ 458\,000.0 & -0.860 & 8.0 \times 10^{-9} & 0.5700 & -0.0007150 \\ -16\,900.0 & -6120.0 & -5.120 \times 10^{-6} & -261\,000.0 & -6.450 \times 10^{-7} \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} = \begin{pmatrix} -3.869649822 \\ 438\,000.860338385 \\ 19\,700.05679226 \\ 457\,999.709285008 \\ -284\,020.000\,005\,765 \end{pmatrix} \quad (6)$$

Точное решение этой задачи представляет собой единичный вектор.

Обе задачи были решены методом LU – разложения с использованием стандартного типа *double* и с использованием предложенного формата представления чисел. Сравнение проводилось по двум критериям – время решения и число верных значащих цифр в решении. Результаты эксперимента представлены в таблице 1.

Вычисления проводились на персональном компьютере с двухъядерным процессором Intel Core 2 Duo 2,53 ГГц и оперативной памятью 4 ГБ, в среде компилятора GCC на операционной системе MAC OS 10.6.

Таблица 1 – Сводная таблица результатов тестирования

Тестовая задача	Время решения, с		Число верных значащих цифр при $M=40$	
	Тип <i>double</i>	Десятичная арифметика	Тип <i>double</i>	Десятичная арифметика
Тест 1. Задача Уилкинсона Обусловленность системы $Cond(A) = 10^6$	0,001	76,899	10	33
Тест 2. Задача Вохминцева Обусловленность системы $Cond(A)=10^{12}$	0,001	418,204	2	26

Из анализа полученных результатов можно сделать вывод, что разработанное программное средство успешно решает тестовые задачи с необходимой точностью, значительно превосходящей точность стандартных типов данных. Наши эксперименты показали, что для получения решений СЛАУ с требуемой точностью надо выполнять вычисления с вещественными числами, у которых мантисса имеет более $2 \cdot m$ десятичных разрядов (в зависимости от числа обусловленности $cond(A)$ матрицы коэффициентов СЛАУ), где m – требуемое количество верных значащих цифр для всех значений элементов вектора решения СЛАУ. Существенным недостатком является значительное увеличение времени вычислений.

Заключение

В работе было проведено тестирование разработанного программного средства. Из результатов тестирования видно что необходимая точность решения была легко достигнута, в то время как временные затраты на вычисление оказались высоки. Несмотря на это, данное направление является перспективным для дальнейших разработок. Регулярность структуры представления данных предоставляет широкие возможности для распараллеливания вычислений. Также вычислительные затраты могут быть снижены за счет выбора более эффективных алгоритмов арифметических операций. Еще одним возможным направлением является использование свободных библиотек для представления чисел с плавающей точкой, у которых мантисса имеет большое число десятичных разрядов.

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. 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. – часть 1 - №7, часть 2 - №8.
3. Д.М. Жук, В.Б. Маничев, М.К.Сахаров SADEL – библиотека «сверхточных» решателей для программного комплекса ПА10 (SADEL-PA10) // Сборник докладов V Всероссийской научно-технической конференции “Проблемы разработки перспективных микро- и наноэлектронных систем – 2012 МЭС-2012”. Россия, Москва, ИППМ РАН. С. 147-153
4. Дж. Голуб, Ч. Ван Лоун, Матричные вычисления – М.: Мир, 1999. 548 с.
5. Маничев В. Б., Глазкова В. Н., Кожевников Д. Ю., Кирьянов Д. А., Сахаров М. К. Решение систем линейных алгебраических уравнений с гарантированной точностью. Вестник МГТУ им. Н.Э. Баумана. Сер. “Приборостроение”. 2011. № 4. – С. 25-36

МОДУЛЬ ВНЕДРЕНИЯ СКРЫТОЙ ИНФОРМАЦИИ НА ОСНОВЕ МЕТОДОВ СТЕГАНОГРАФИИ НА ОСНОВЕ ТОНКОГО КЛИЕНТА ДЛЯ СОЗДАНИЯ СХЕМ РАЗЛИЧНОЙ ФИЗИЧЕСКОЙ ПРИРОДЫ

Шестаков М.Е.

Научный руководитель: доцент, кандидат наук, Волосатова Т.М.

МГТУ им. Н.Э.Баумана, кафедра РКБ, Москва, Россия

MODULE EMBEDDED HIDDEN INFORMATION BASED ON THE STEGANOGRAPHY METHOD BASED ON THIN CLIENT FOR THE CREATION OF SCHEMES OF DIFFERENT PHYSICAL NATURE

Shestakov M.E.

Supervisor: Associate Professor, Ph.D., Volosatova T.M.

MSTU, Moscow, Russia

Аннотация

В статье рассматриваются принципы модульного построения тонкого клиента для создания и редактирования схем различной физической природы. Подробно рассмотрен метод нанесения цифрового водяного знака. Кратко представлено сравнение возможностей современных технологий создания тонкого клиента. В заключении представлены рекомендации по выбору технологий.

Abstract

The article describes module structure principles of the thin client to create and edit circuits of different physical nature. Detailed digital watermarking technology described. Summarizes the comparison of the capabilities of modern technology to create a thin client. In conclusion, presented recommendations for the choice of technology.

Введение

Целью работы является создание приложения для защиты документации, используя методы стеганографии, на основе редактора схем различной физической природы. Модульность приложения WebCAD позволяет реализовать данный метод без существенной переработки исходного приложения. Тем самым, расширяется исходный функционал приложения, появляются более совершенные методы защиты чертежей.

Построение схем является неотъемлемой частью инженерных задач. Автоматизация данного процесса позволяет ускорить их решение, снизить затраты и уменьшить количество ошибок. Разработка приложения велась с использованием систем командной разработки, таких как система контроля версий.

Инновационным подходом в создании приложений подобной направленности является клиент-серверная архитектура. Большинство существующих аналогов требуют наличия дополнительного программного обеспечения для корректной работы.

1 Передача информации методом стеганографии

Стеганографическая система (стегосистема) — объединение методов и средств используемых для создания скрытого канала для передачи информации. Стеганография в наше время часто применяется, как правило, для встраивания цифровых водяных знаков, являющееся основой для систем защиты авторских прав и DRM (Digital rights management) систем.

Мы реализуем метод LSB.LSB (Least Significant Bit, наименьший значащий бит) — суть этого метода заключается в замене последних значащих битов в контейнере (изображения, аудио или видеозаписи) на биты скрываемого сообщения. Разница между пустым и заполненным контейнерами должна быть не ощутима для органов восприятия человека.

Также будем использовать для шифрования/дешифрования bmp файл, не содержащий палитру. В таком bmp файле каждые 3 байта определяют 3 цвета пикселя.

Подготовка к реализации

Так как проводится работа с битами информации, а цвет одного пикселя занимает один байт, то потребуются методы преобразования байта в биты и наоборот.

Итак, расположение в bmp информации будет следующим:

- Пиксель 0,0: признак того, что в файле есть текстовая информация. В качестве признака используется символ /

- Пиксели 0.1 — 0.3: размер текстовой информации, записанной в файл

- Пиксели 0.4 и до конца файла: собственно текстовая информация

В коде в переменной хранится код символа "/", который является признаком того, что в файле хранится зашифрованная информация. Далее этот код преобразуется в массив бит. Затем каждый из трех составляющих цветов пикселя преобразуется в массив бит, после этого в красном цвете заменяются младшие 2 бита на биты символа "/", в зеленом заменяются младшие 3 бита на биты символа "/" и в синем также заменяются младшие 3 бита цвета. Из 3 новых полученных цветов создается новый цвет пикселя и устанавливается вместо предыдущего цвета.

Способ записи информации, то есть 2 бита, 3 бита и 3 бита выбран для удобства работы, ибо в один пиксель записывается сразу байт информации.

Далее в файл в файл записывается количество символов исходного текста. Каждая цифра занимает один байт, поэтому максимальная длина исходного текста — $999 - 4 = 995$ символов (4 — это один пиксель на признак присутствия информации в файле и три пикселя на размер текстовой информации). При необходимости можно увеличить, взяв пиксели не с 0.1 по 0.3, а с 0.1 по 0.4 например, и так далее. В цикле каждая цифра количества исходного текста преобразуется в массив бит и записывается в младшие пиксели цвета по принципу, описанному выше.

Затем происходит запись информации по принципу: Допустим, имеется 8-битное изображение в градациях серого. 00000000b обозначает чёрный цвет, 11111111b — белый. Всего имеется 256 градаций. Также предположим, что сообщение состоит из 1 байта — например, 01101011b. При использовании 2 младших бит в описаниях пикселей, нам потребуется 4 пикселя. Допустим, они чёрного цвета. Тогда пиксели, содержащие скрытое сообщение, будут выглядеть следующим образом: 00000001 00000010 00000010 00000011. Тогда цвет пикселей изменится: первого — на $1/255$, второго и третьего — на $2/255$ и четвёртого — на $3/255$. Такие градации, мало того, что незаметны для человека, могут вообще не отобразиться при использовании низкокачественных устройств вывода.

Метод LSB несложен в реализации, и его можно использовать для сокрытия нужной информации в bmp файле. Но существенным недостатком метода является то, что размер bmp имеет большой размер, что делает этот метод нежизнеспособным для передачи конфиденциальной информации по сети интернет.

Особенности реализации

Одной из особенностей приложения является его модульная архитектура. Использование данной архитектуры позволяет добавлять новую функциональность без существенной переработки приложения. На базе приложения WebCAD разработан модуль с реализацией стеганографического метода.

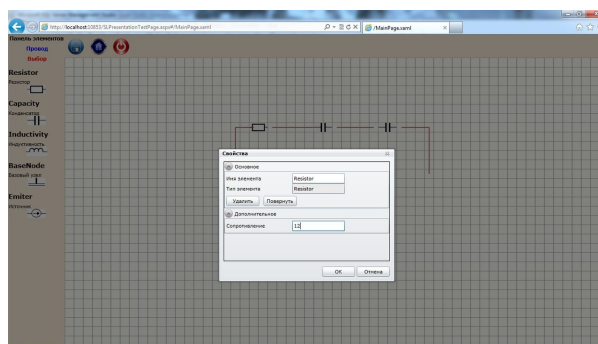


Рисунок 1 - Окно свойств элемента

Функциональность данной реализации метода стеганографии наименьшего значащего не ограничивается возможностью внедрения текстовой информации в файл изображения чертежа. Главным преимуществом является возможность внедрения любой информации, вплоть до другого изображения. Данный метод позволяет сделать скрытой информацию как для человеческого глаза, так и для компьютера. Таким образом, метод наименьшего значащего бита может являться защитой чертежа в случае попытки открыть его вне нашей системы. В случае попытки открыть чертеж в системе, не предназначенную для его воспроизведения или модификации, чертеж будет недоступен для пользователя.

Заключение

В результате данной работы был реализован модуль системы WebCAD для внедрения информации на основе метода стеганографии. В отличие от другого программного обеспечения с подобным функционалом, пользователю не требуется сторонних приложений для работы, кроме веб-браузера с установленным плагином Silverlight. Достоинством данной реализации является простота ее использования при богатом функционале. К недостаткам можно отнести возможность удаления зашифрованной информации из файла, из-за использования простого алгоритма, что решается подбором и разработкой более качественных алгоритмов. В рамках данного проекта поставленные задачи были выполнены. Более того, имеется техническая возможность для расширения функционала за счет модульности приложения.

Литература

1. Троелсен Э., Язык программирования C# 2010 и платформа .NET 4.0. — М.: Вильямс, 2011. 1344 с.
2. Макдональд М., Microsoft ASP.NET 4 с примерами на C# 2010 для профессионалов. — М.: Вильямс 2010. 1424 с.
3. Конахович Г.Ф., Пузыренко А. Ю., Компьютерная стеганография. Теория и практика. — М.: мк-Пресс, 2006. 288 с.
4. Lowy J., Programming WCF Services — O'Reilly Media, 2008, 784 с.
5. Brown P., Silverlight 4 in Action — Manning Publications, 2010, 800 с.
6. Комбинированные методы защиты данных в САПР. - Волосатова Т.М., Денисов А.В., Чичварин Н.В. // Информационные технологии, Приложение №5, 2012 – 32 с.: ил.
7. Защита проектной документации от несанкционированного доступа. - Волосатова Т.М., Денисов А.В., Чичварин Н.В. // 9 Международная конференция «Эффективные методы автоматизации подготовки и планирования производства», сборник трудов, с141-145. М.: 2012.

ОБЗОР МЕТОДОВ ЛОКАЛИЗАЦИИ ОБЪЕКТОВ В 3D

Шмакова Н.А.

Научный руководитель: к.т.н., доцент, Волосатова Т.М.
МГТУ им. Н.Э.Баумана, кафедра РК6, Москва, Россия

REVIEW OF LOCALIZATION METHODS FOR 3D OBJECTS

Shmakova N.A.

Supervisor: Ph.D., assistant professor, Volosatova T.M.
MSTU, Moscow, Russia

Аннотация

В статье рассматриваются различные современные подходы к локализации объектов в 3D, а также приводятся их краткие оценки. Основанием для данных подходов являются сопоставления краев, графов частей объектов, CAD STL модели, классификаторы, зависящие от положения объекта, а также 2,5D карты глубины. В заключение представлены характеристики сенсора kinect, используемого для получения карты глубины сцены.

Abstract

The paper shows modern approaches to 3d objects localization and their comparison. These methods are based on edges, constellations of object parts, CAD STL models, pose-specific classifiers and 2.5 depth maps. In conclusion there represented kinect sensor characteristics which are used for scene depth mapping.

Введение

Машинное зрение для промышленного применения требует быстрых и простых в использовании решений. Зачастую они сводятся к работе с 2D изображениями при помощи контроля освещенности. В этой области существует множество хорошо изученных методов. Однако, иногда 2D информации недостаточно, например, для управления манипуляторами, решения задач дистанционного управления жестами и многих других приложений. В каждом из вышеуказанных случаев решается подзадача локализации определенного типа объектов в трехмерном пространстве.

В данной работе рассмотрены различные современные подходы к локализации объектов в 3D (рис. 1), а также дана их краткая оценка.

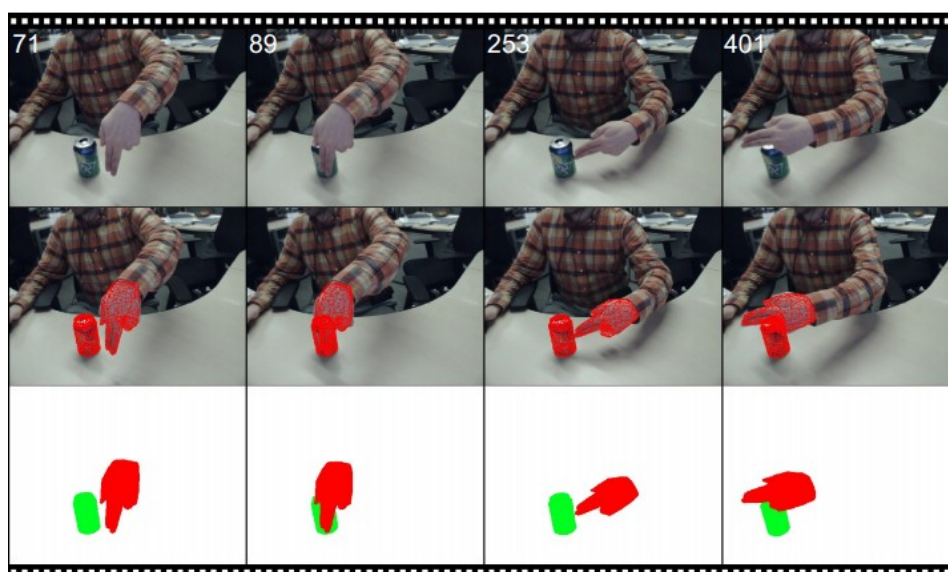


Рисунок 1 - Динамическая локализация объектов в 3D [1]

1 Локализация 3D объектов на основе сопоставления краев для скоростных и высокоточных производственных приложений машинного зрения

Процесс локализации начинается с сохранения изображения начального состояния конвейера и ожидания, пока первый объект на нем не перекроет подсветку. Затем происходит сохранение снимков объекта и их сегментация. Фон удаляется простым вычитанием первоначальных снимков из текущих. Далее изображение бинаризуется по порогу, возможно адаптивному. Выделение контуров при помощи фильтра Канни сопровождается их дилатацией. Полученные бинарные изображения масштабируются, и их точки сопоставляются для формирования стереоскопического скелета исследуемой формы. 3D положение этого «скелета» является ответом на вопрос локализации объектов на конвейере.

Оценка точности локализации проходила на экспериментах, в которых вычислялось расстояние между фактическим центром масс локализуемого объекта и результатами работы предложенной системы. Средняя погрешность вычисления на 50 примерах не превышает 20 мм при расположении объекта на расстоянии 1 м от камеры. Метод является робастным и подходит для промышленного применения. Область применения предложенной системы ограничена из-за использования сложной системы подсветки [2].

2 Локализация 3D объектов на основе окклюзии контуров с использованием STL CAD моделей

Шаблон для сопоставления представляет собой набор особых 2D изображений искомого объекта с разных точек обзора. Он автоматически генерируется из CAD модели и состоит из геометрических особенностей, а также из точек в трехмерном пространстве. К геометрическим особенностям относятся углы и дуги, которые восстанавливаются по точкам STL модели.

Для оценки эффективности был проведен ряд экспериментов с реальными и синтетическими сценами на изображениях 640x480 пикселей с 256 градациями серого. Время вычислений на машине с процессором Intel Core2Duo E6850 и графической картой GeForce 8800 Ultra при наличии 123527 гипотез составляет 416,72 секунд, что довольно долго для промышленного применения. Однако, из результатов экспериментов, видно, что метод обладает высокой робастностью. Уменьшить затраты времени на локализацию можно с помощью отсева части гипотез на этапе начального сравнения [3].

3 3D локализация в рамках определенного класса объектов с использованием графов частей сцены

Предполагается, что дано множество 3D экземпляров формы, состоящих из набора вершин и граней. Принадлежность вершины поверхности описывается дескриптором smv. В данной работе используется инвариантный к ориентации дескриптор HKS (Heat Kernel Signature).

Задано множество частей, расположенных в 3D пространстве для каждого тестового объекта. В рамках данного метода поставлена цель - изучить графическую модель. После обучения модель используется для объединенной оценки оптимального расположения частей объекта в запрашиваемой сцене максимизацией целевой функции.

Было проведено сравнение методов локализации на основе древовидных и полносвязных графов. Задача заключалась в поиске оптимальной конфигурации узлов модели или сцены, что эквивалентно решению проблемы разметки, где каждый узел графа (часть объекта) может принять одну из N возможных меток (приблизительно 7000 позиций вершин). В свою очередь, каждое ребро графа может иметь NxN комбинаций меток вершин на его концах. Оптимизация включала в себя перебор унарных и попарных условий для всех вариантов разметки.

Несмотря на высокую производительность метода, основанного на древесных графах, происходит серьезная потеря качества. Поэтому рекомендуется использовать полносвязные

графы с гораздо более сложными моделями, которые не гарантируют оптимального решения, но позволяют оценить близость к нему [4].

4 Эффективный поиск 3D объектов на основе множественных классификаторов, зависящих от положения

Предлагается эффективный метод обнаружения 3D объектов, основанный на быстром вычислении случайных папоротников (Random Ferns) над местными гистограммами ориентированных градиентов. Папоротники представляют собой независимые наборы «случайных» условий, которые проверяются на классифицируемой части изображения. Гистограммы особенностей получаются простым сравнением бинарных выходов, означающих наличие того или иного объекта. Случайные папоротники разделены между классами объектов и используются для локализации, основанной на двухступенчатом подходе. Подразумевается наличие оценщика положения и множества классификаторов для каждой конкретной ориентации, которые могут быть обучены менее чем за минуту, в то время как обнаружение объекта займет около 1 секунды. Данная работа использует новый подход на основе преобразования Хафа и случайных папоротников для эффективной и робастной оценки положения объекта в 3D.

Представленный метод обеспечивает высокую скорость обнаружения и эффективность для локализации объектов, снятых с различных точек обзора. Основное достоинство метода заключается в том, что локальные особенности, сгруппированные в случайные папоротники, вычисляются единожды и независимо. Первое условие сокращает время вычислений, а второе позволяет довольно легко распараллеливать процесс вычислений [5].

5 Локализация при трекинге головы и руки на основе 2.5D данных

Трекинг тела предоставляет пространственно-временную информацию, такую как траектории движения точек интереса или углы соединения на модели тела. Эти данные могут быть использованы на этапе классификации жестов, расширяя входную информацию для классификатора, что ведет к более устойчивой его работе.

Рассматривается трекинг головы и рук человека. Положение головы оценивается в зависимости от расстояния между объектом и камерой. Подобная оценка устойчива к частичным окклюзиям и быстрому движению, а также помогает определить регион, где потенциально могут находиться руки.

Трекинг требует проведения двух подготовительных шагов, таких как извлечение фона и обнаружения объектов. Которые представляют наибольший интерес в контексте данной работы.

Маска переднего плана просто и быстро извлекается из 2,5D оценки глубины изображения по постоянному порогу. Для того чтобы подавить шумы, обрабатывается несколько последовательных кадров сцены, а затем результаты усредняются.

Рассмотренный метод локализации, основан на предварительно полученных 2,5D данных об исследуемой сцене, на точность которых накладываются определенные ограничения. С его помощью можно решать только те задачи, в которых глубина объекта интереса в сцене значительно отличается от глубины фона. Вместе с тем, при выполнении вышеуказанных условий, метод показывает высокую скорость вычислений и устойчивость к зашумлениям [6].

6 Использование сенсора Kinect для получения карты глубины.

Kinect – это инфракрасный и звуковой сенсор, разработанный для игровой консоли Xbox360. Значительно позднее появилась возможность подключения к персональным компьютерам и разработки приложений. Сенсор подключается к компьютеру через USB. Важный аспект аппаратной части – диапазон «рабочих» расстояний для дальномера.



Рисунок 2 - Пример карты глубины сцены, полученной при помощи сенсора Kinect

Развитые инструменты разработки позволяют без труда получать карту глубины (рис. 2), динамическое 3D изображение, звуковые гистограммы и осуществлять трекинг тела человека.

Заключение

В данной работе были рассмотрены пять различных подходов к локализации объектов в трехмерном пространстве: на основе сопоставления краев, графов частей объектов, по CAD STL модели, на основе классификаторов, зависящих от положения объекта, а также на основе 2,5D карты глубины. Каждый метод находит свое применение в узкой предметной области. В подобных случаях эффективность выбранного подхода существенно зависит от решаемой задачи.

Существует множество технических средств, упрощающих те или иные этапы локализации объектов в 3D. Одним из таких средств является сенсор Kinect, который позволяет в режиме реального времени получать карту глубины объекта, не вдаваясь в детали реализации.

Литература

1. Victor A. Prisacariu PWP3D: Real-time Segmentation and Tracking of 3D Objects / Victor A. Prisacariu, Ian D. Reid // Proceedings of the 20th British Machine Vision Conference - 2012.- С. 1 – 23.
2. Demuynck, O. Edge based 3D object localization for fast and precise industrial machine vision applications / O. Demuynck, C. Pacheco Cedeño, A. L. Moore // International journal of systems applications, engineering & development.- 2009.- Issue 2, Volume 3.- С. 46 – 52.
3. Maruyama, K. 3D Object Localization based on occluding contour using STL CAD model / Y. Kawai, T. Yoshimi, F. Tomita // Pattern Recognition, 2008. 19th International Conference on computing & processing (Hardware/Software); Signal Processing & Analysis.- 2008.- С. 1 – 4.
4. Prasad, M. Class-specific 3D localization using constellations of object parts / M. Prasad, J. Knopp, L. Van Gool // BMVC 2011. The 22nd British Machine Vision Conference.- 2011.- С. 34.1 – 34.11.
5. Villamizar, M. Efficient 3D Object Detection using Multiple Pose-Specific Classifiers / M. Villamizar, H. Grabner, F. Moreno-Noguer, J. Andrade-Cetto, L. Van Gool, A. Sanfeliu // BMVC 2011. The 22nd British Machine Vision Conference.- 2011.- С. 20.1 – 20.10.
6. Suau, X. Real-time head and hand tracking based on 2.5D data / X. Suau, J. Ruiz-Hidalgo, J. R. Casas // IEEE transactions on multimedia, vol. 14, no. 3.- 2012.- С. 575 – 585.

ПОСТРОЕНИЕ МОДЕЛИ НИЖНЕЙ ЧЕЛЮСТИ ЧЕЛОВЕКА. МОДЕЛИРОВАНИЕ ПЕРЕЛОМОВ.

Соля Д.М., Соля Д.П.

Научный руководитель: к.т.н., доцент, Жук Д.М.
МГТУ им. Н.Э.Баумана, кафедра РК6, Москва, Россия

CREATION OF HUMAN MANDIBLE MODEL. FRACTURE MODELLING.

Soia D. M., Soia D.P.

Supervisor: PhD, Docent, Zhuk D. M.
MSTU, Moscow, Russia

Аннотация

В ходе исследовательской работы, в системе Solid Edge ST4 была создана модель нижней челюсти, а также имитация ее переломов. Были проверены разные способы крепления частей челюсти металлическими пластинами. Также был проведен статический анализ с помощью встроенного в эту систему модуля Nastran. Были проведены виртуальные испытания, которые показали адекватность модели. В процессе работы были изучены возможности дальнейшего применения Solid Edge ST4 в подобных исследованиях.

Summary

During the research, the human mandible model was created in Solid Edge ST4, along with imitation of its fractures. Different methods of fixing parts of broken mandible by using metal plates were tested. The static analysis of this model in built-in Nastran module was also conducted. Simulation was carried out, which showed the adequacy of the model. During the research, different possibilities of further use of Solid Edge in future works in this field were studied.

Введение

При переломе нижней челюсти, важно оказать своевременную и качественную помощь пострадавшему. В противном случае могут возникнуть серьёзные осложнения, которые осложнят лечение и могут значительно ухудшить качество жизни человека. При лечении переломов для фиксации частей челюсти используются специальные шины, проволочные или в виде металлических пластин. Для облегчения выбора пластин, а так же их расположения, которое обеспечит скорейшее и наиболее полное выздоровление, предпринимаются попытки построения моделей и расчётных схем [1, 2], с помощью которых можно определить величину и картину распространения напряжений в костной ткани челюсти. На текущий момент, однако, не существует единых методов построения подобных моделей.

В ходе исследования мы решили проверить возможность использования при создании таких моделей современных систем автоматизированного проектирования (САПР) на примере Solid Edge ST4. Использование современных, расширяемых систем позволит упростить процесс выбора шины и места её крепления в каждом конкретном случае.

По данным, полученным из центрального научно-исследовательского института стоматологии и челюстно-лицевой хирургии (ЦНИИСиЧЛХ), нами была построена в системе Solid Edge ST4 модель нижней челюсти человека. Затем мы имитировали несколько переломов и произвели статических анализ для нескольких вариантов наложения шин. Целями данной работы были, во-первых, проверка применимости для решения таких задач методов и средств используемых при проектировании и анализе механических систем, поскольку челюсть можно рассматривать с точки зрения механики, во-вторых, анализ рассматриваемых способов крепления частей челюсти.

Результаты работы показали, что использование САПР для решения такого рода задач возможно, а так же, что рассмотренные варианты крепления в большинстве случаев позволяют шинам выдерживать возникающие нагрузки и обеспечивать целостность челюсти.

Основная часть доклада разбита на разделы следующим образом: в разделе 1 описывается модель нижней челюсти человека, её геометрические и механические характеристики, а так же места приложения нагрузки и её размеры; в разделе 2 рассмотрены несколько случаев переломов и проведён анализ распространения напряжений при разных вариантах крепления.

1 Создание модели нижней челюсти человека в Solid Edge ST4

На основании данных, полученных от группы хирургов ЦНИИСиЧЛХ в системе Solid Edge ST4 была построена модель нижней челюсти человека. Размеры модели соответствуют приведённым на рисунке 1.

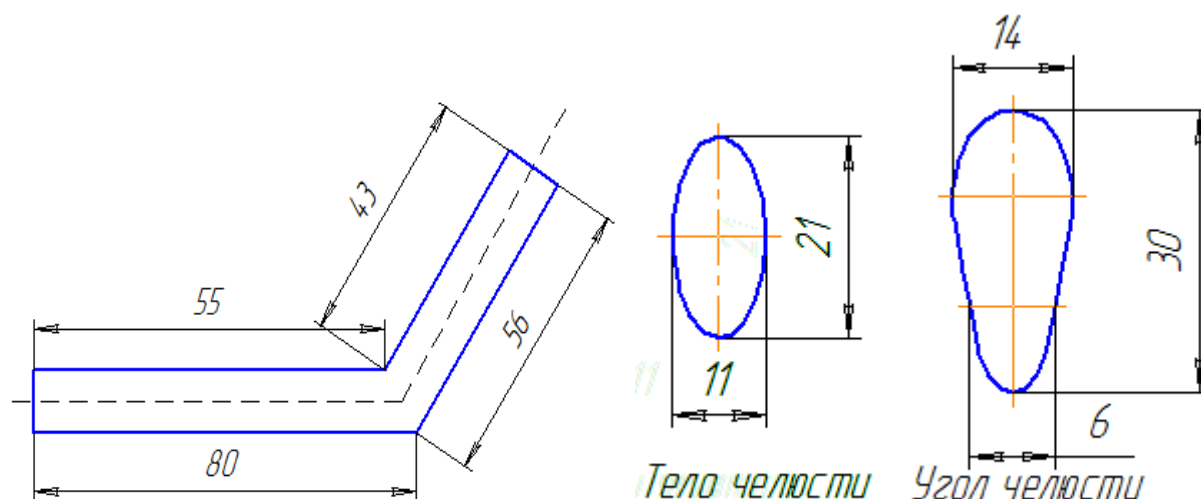


Рисунок 1 – размеры нижней челюсти человека. Угол между телом и ветвью челюсти составляет 120°

В здоровом состоянии вся нагрузка распространяется по кортикальному слою. Модель представляет собой модель кортикального слоя, толщина которого принята 2 мм для всей модели челюсти.

На основании обобщения приведённых в различных источниках [3, 4] данных, нами приняты следующие механические характеристики кортикального слоя:

- модуль Юнга $E = 2$ ГПа;
 - предел прочности: $\sigma = 60 \cdot \rho^2$, где $\rho = 1,85$ г/см³ – плотность.
- Таким образом, $\sigma = 205$ МПа;
- коэффициент Пуассона $\mu = 0,29$.

Фиксация челюсти для статического анализа происходит следующим образом: в суставах производится запрет перемещения с разрешением вращения. При моделировании жевания на челюсть накладываются ограничения движения в районе премоляров-моляров. В случае откусывания - в районе резцов.

Силы приложены на мышечковых отростках по 750 Н с каждой стороны и на углах по 900 Н с каждой стороны. При данных нагрузках напряжения не должны превышать предел прочности. Проведённый во встроенном в Solid Edge ST4 модуле Nastran статический анализ, результаты которого показаны на рисунке 2 и в таблице 1, показал адекватность модели.

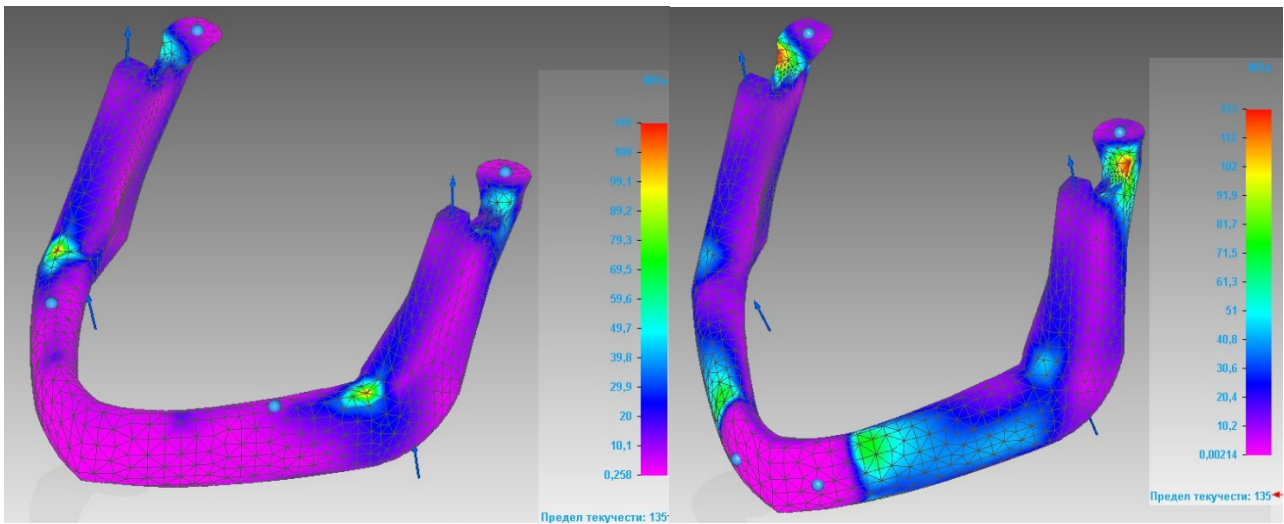


Рисунок 2 – распространение напряжений по фон Мизесу для челюсти без переломов

Таблица 1 – Максимальные значения напряжения по фон Мизесу для челюсти без переломов

Способ фиксации	Максимальное напряжение, МПа
На молярах	119
На резцах	123

В качестве шин были выбраны металлические пластины из титанового сплава со следующими характеристиками[5]:

- модуль Юнга $E = 115$ ГПа;
- предел прочности $\sigma = 990$ МПа;
- плотность $\rho = 4,4$ г/см³;
- коэффициент Пуассона $\mu = 0,33$.

Размеры пластины: длина - 25 мм, толщина - 1мм, ширина - 3 мм.

2 Моделирование соединения частей челюсти при переломах

Рассмотрим три наиболее часто встречающихся случая переломов: в углу челюсти, в области клыка и в области сустава. В первых двух случаях части челюсти будут соединяться с внешней стороны в области перелома металлическими пластинами. Металлические пластины будут прикрепляться по одиночке сверху и снизу челюсти, а так же парно, как показано на рисунках 3, 4. При переломах в области сустава, соединение осуществляется с помощью одной пластины расположенной вдоль ветви как показано на рисунке 5. Статический анализ так же проводился в модуле Nastran. Для упрощения расчётов, пластины и челюсть соединяются путём склейки. Результаты анализа приведены в таблице 2.

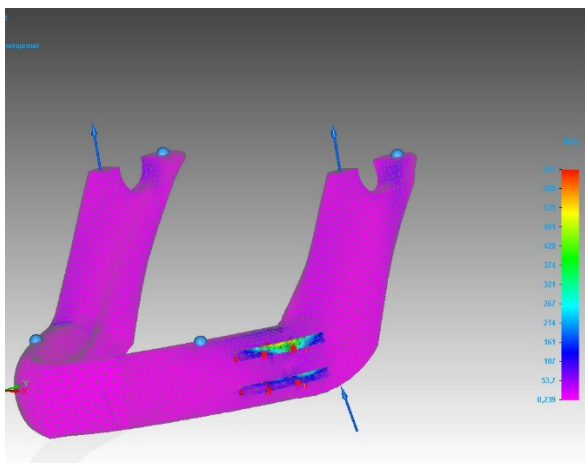


Рисунок 3 – статический анализ при переломе в углу челюсти.

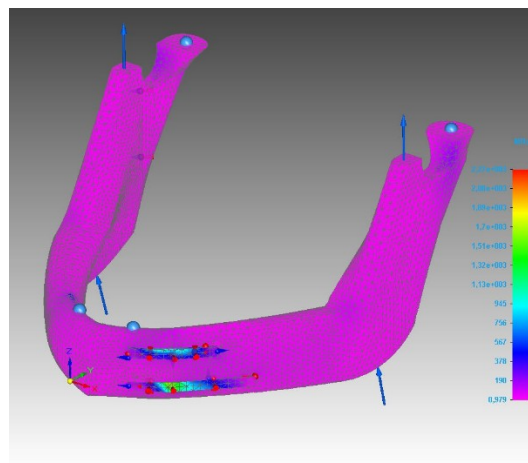


Рисунок 4 – статический анализ при переломе в области клыка

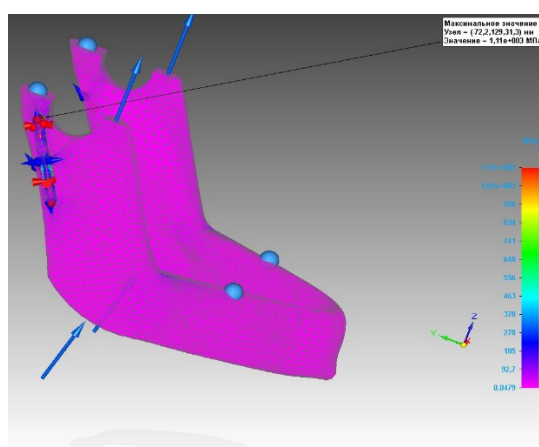


Рисунок 5 – статический анализ при переломе в области сустава.

Таблица 2 – Результаты анализа

Тип перелома	Способ фиксации перемещения	Способ наложения пластины	Максимальное напряжение, возникающее в пластинах, МПа
Перелом в углу	Фиксация на молярах	Одна пластина сверху	911
		Одна пластина снизу	515
		Две пластины	642
	Фиксация на резцах	Одна пластина сверху	942
		Одна пластина снизу	836
		Две пластины	673
Перелом в области клыка	Фиксация на молярах	Одна пластина сверху	164
		Одна пластина снизу	145
		Две пластины	190
	Фиксация на резцах	Одна пластина сверху	4,41e+003
		Одна пластина снизу	3,09e+003
		Две пластины	2,43e+003
Перелом в области сустава	Фиксация на молярах	Одна пластина	1,11e+003
	Фиксация на резцах	Одна пластина	1,85e+003

Согласно данным, представленным в таблице 2, в большинстве случаев напряжения, возникающие в пластинах не превышают предельные значения. Напряжения, возникающие в костной ткани лежат в допустимых пределах для костной ткани. Большие напряжения, возникающие в пластинах при откусывании при переломе в области клыка обусловлены большой длинной действующего рычага. Максимальное напряжение при переломе в области сустава сосредоточено в одной точке и обусловлено неточностью вычислений, возникшей из-за упрощённого вида модели пластины. В целом, в данном случае напряжение, возникающее в пластине не превышает 650 МПа. Так же следует отметить, что при статическом анализе использовались те же значения действующих сил со стороны мускулатуры, что и для здоровой челюсти. В реальности, при лечении подобных травм, подобные усилия возникать не будут и данный случай можно рассматривать как предельный, особенно при откусывании.

Заключение

В ходе проведённой работы нами было установлено, во-первых, что предложенные варианты наложения пластин в большинстве случаев действительно позволяют обеспечить надёжное скрепление частей челюсти, и, во-вторых, методы и средства, применяемые при построении и анализе механических систем, в целом, можно использовать для решения задач, подобных описанной в докладе. Однако, для повышения точности анализа необходимо построение более сложных моделей челюсти, отражающих индивидуальные особенности строения челюсти конкретного пациента. Такие модели можно получать, преобразуя данные, полученные в результате, например, томографии в трёхмерную твердотельную модель челюсти. Разработка методов подобного преобразования позволит существенно улучшить наше представление о поведении челюсти как механической системы.

Использование моделей, которые могут быть получены путём такого преобразования, приведёт к увеличению времени вычислений, но значительно снизит затраты времени на построение модели, которое, например, в ходе этой работы велось вручную.

Использование современных САПР позволит получать адекватные модели в случае множественных переломов и сложных случаев при травмах лицевой части черепа. Это позволит сократить время на подготовку операций, а так же уменьшит количество ошибок.

Мы хотим выразить благодарность коллективу хирургов ЦНИИСиЧЛХ за предоставленные данные.

Литература

1. Чуйко А.Н., Маргвелашвили А.В., О расчете прочности нижней челюсти при функциональной нагрузке (в норме). *Стоматолог.* – 2009. - №2. – С.46-50.
2. Попович В.С., Семенников В.И. и Семенникова Н.В., ПРОГНОЗИРОВАНИЕ МЕТОДА ФИКСАЦИИ ОТЛОМКОВ НИЖНЕЙ ЧЕЛЮСТИ ПРИ ТРАВМАХ. *ПОЛЗУНОВСКИЙ ВЕСТНИК* – 2011. – №2/2. – С.340-345
3. S. Dendorfer, H.J. Maier, D. Taylor, J. Hammer, Anisotropy of the fatigue behaviour of cancellous bone, *Journal of biomechanics* 1 January 2008 (volume 41 issue 3 Pages 636-641 DOI: 10.1016/j.jbiomech.2007.09.037)
4. Чуйко А.Н., Шинчуковский И.А., ОПРЕДЕЛЕНИЕ ОСНОВНЫХ МЕХАНИЧЕСКИХ ХАРАКТЕРИСТИК КОСТНЫХ ТКАНЕЙ НА БАЗЕ ДАННЫХ КОМПЬЮТЕРНОЙ ТОМОГРАФИИ, *СОВРЕМЕННАЯ СТОМАТОЛОГИЯ* 1/2011 – С. 90-98
5. Strózyk P., Szust A., Effect of mandible reinforcement on static and fatigue strength, after partial mandible branch marginectomy. *Wroclaw University of Technology, Faculty of Mechanical Engineering.*

ИНТЕРФЕЙС «МОЗГ-КОМПЬЮТЕР» - ПОДХОДЫ К РЕАЛИЗАЦИИ ИНТЕРФЕЙСА И МЕТОДЫ ОБРАБОТКИ ДАННЫХ

Сотников П.И.

Научный руководитель: д.ф.-м.н., доцент, Карпенко А.П.

МГТУ им. Н.Э.Баумана, кафедра РК-6, Москва, Россия

«BRAIN-COMPUTER» INTERFACE - BCI TYPES AND DATA PROCESSING METHODS

Sotnikov P.I.

Supervisor: Dr., Prof., Karpenko A.P.

MSTU, Moscow, Russia

Аннотация

В статье рассматриваются различные типы нейрокомпьютерных интерфейсов, принципы их работы, основные методы и алгоритмы преобразования электрической активности головного мозга человека в управляющие команды внешнему техническому устройству. Особое внимание уделяется неинвазивным интерфейсам, основанным на анализе электроэнцефалограмм.

Abstract

This article aims to review the various types of brain-computer interfaces. Furthermore, this article describes the basic methods and algorithms for translating a variety of different electrophysiological signals of the human brain into commands that operate a computer display or other external device. Particular attention is paid to non-invasive interfaces that are based on analysis of the electroencephalogram signals.

Введение

Интерфейс мозг-компьютер (ИМК) (англ. — Brain-Computer Interface, BCI) является новой системой коммуникации человека с машиной (например, с компьютером или протезом), основанной на непосредственном преобразовании намерений человека, отраженных в регистрируемых сигналах мозга, в управляющие команды. Для таких систем могут использоваться различные сигналы, отражающие активность мозга: электроэнцефалограмма (ЭЭГ), магнитоэнцефалограмма (МЭГ), позитронно-эмиссионная томография (ПЭТ), электрокортикограмма (ЭКоГ), импульсная активность нейронов, распределение и интенсивность кровотока в мозге и т.д. Однако наибольшее число исследований касается использования электрических сигналов.

1 Классификация систем мозг-компьютерного интерфейса

В системах мозг-компьютерного интерфейса используются либо биопотенциалы мозга, зарегистрированные с поверхности головы человека, либо биопотенциалы, снимаемые непосредственно с поверхности коры головного мозга или от отдельных нейронов глубоких структур мозга. В первом случае ИМК называются *неинвазивными*, во втором – *инвазивными* [1,2].

Среди различных типов интерфейса мозг-компьютер самый значительный успех был получен при применении инвазивных технологий ИМК.

Исследовательской группой Джозефа О’Доэрти (*Department of Biomedical Engineering, Duke University, USA*), был разработан интерфейс мозг-компьютер, позволяющий обезьянам, как контролировать внешнее техническое устройство (курсор на мониторе компьютера, или виртуальная рука), так и получать обратную связь в виде искусственных тактильных ощущений, вызванных микростимуляцией первичной соматосенсорной коры [3]. Микростимуляции возникали каждый раз, когда курсор или виртуальная рука, касались виртуальных объектов, причем интенсивность стимуляции кодировала определенные свойства виртуального объекта (текстуру). В ходе эксперимента обезьяны использовали ИМК для поиска одного из трех визуально неотличимых объектов,

на основании данных о его текстуре, полученных с помощью искусственных тактильных ощущений.

Эксперименты, проведенные исследовательской группой Брайана Пасли (*Helen Wills Neuroscience Institute, University of California Berkeley, USA*) с использованием инвазивного ИМК, показали возможность восстановления звуковой информации, воспринимаемой человеком, на основе данных, полученных с электродов, расположенных в районе верхней височной извилины коры головного мозга [4]. Используемый в ходе экспериментов метод восстановления речевой информации позволил обнаруживать медленные и средние временные флуктуации, которые соответствуют скорости произношения слогов, однако восстановленная речь является все еще неразборчивой для человека.

Несмотря на то, что инвазивные методы ИМК открывают большие перспективы для исследований, их применение сильно ограничивается необходимостью проведения сложной хирургической операции, а также проблемами, связанными с постепенным зарастанием электродов соединительными тканями. Это приводит со временем к значительному ухудшению электрического контакта с мозгом, с полным его нарушением впоследствии [1]. В связи с этими обстоятельствами наибольшее развитие получили ИМК *неинвазивного* типа.

В литературе, посвященной исследованию ИМК, существующие *неинвазивные* интерфейсы мозг-компьютер по принципу работы подразделяют на три типа.

1) ИМК, основанные на анализе медленных корковых потенциалов (*slow cortical potentials, SCP*). Принцип работы *SCP-BCI* основан на парадигме *оперантного обусловливания*, которая предполагает, что испытуемые способны сознательно изменять и удерживать интенсивность своей корковой поляризации с помощью биологической обратной связи.

2) ИМК, основанные на анализе вызванных потенциалов (*event-related potential, ERP*). Вызванный потенциал (ВП) – это электрическая реакция мозга на внешний раздражитель. В существующих *ERP-BCI* используются в основном два следующих класса вызванных потенциалов.

а) ВП *P300*. Компонент *P300* возникает в ответ на неожиданный, редко предъявляемый (например, предъявляемый с вероятностью 0,2) значимый стимул, когда он появляется среди часто предъявляемых незначимых стимулов. *P300* зависит от внимания испытуемого, но не от физических параметров стимула.

б) ВП *SSVEP*. Зрительный вызванный потенциал *SSVEP (steady state visual evoked potential)* возникает преимущественно в затылочных частях коры, как реакция на визуальные вспышки, повторяющиеся с определенной частотой.

3) ИМК, основанные на анализе сенсомоторных ритмов (*sensorimotor rhythms - SMR*), *SMR-BCI*. В таких интерфейсах для управления внешним устройством, например, рукой робота, пользователь мысленно представляет то или иное движение. В состоянии покоя сенсомоторные ритмы имеют высокую амплитуду (наблюдается *синхронизация* ритма), в то время как при активации соответствующих сенсорных или моторных участков коры, их интенсивность падает (наблюдается *десинхронизация* ритма) [5].

Приведенные три типа ИМК в настоящее время получили широкое распространение. Однако среди работ можно выделить еще два следующих, менее изученных подхода.

4) *Прямые интерфейсы мозг-компьютер*, способные улавливать мысленное представление слова непосредственно в том виде, в каком оно возникает в мозгу. Подход заключается в том, что испытуемому нужно просто подумать «да» или «нет», и при этом не требуется интенсивного обучения с помощью обратной связи [6].

5) Интерфейсы, основанные на распознавании характерных паттернов ЭЭГ, *жестко не ангажированных мозговыми процессами*. Подход базируется на наличии значительной корковой пластичности, которая позволяет мозгу адаптироваться к ИМК таким образом, что управляемое им внешнее устройство (например, протез руки) воспринимается как естественный исполнительный орган человеческого тела [7]. Однако требуется длительное обучение испытуемого с помощью биологической обратной связи.

2 Преобразования активности мозга в команду внешнему устройству

В основе систем ИМК лежит математический анализ нейрофизиологических данных, алгоритмы распознавания характерных признаков и их классификации.

В большинстве работ в качестве *пространственно-временного паттерна ЭЭГ* исследуется пространственное распределение амплитуд различных ритмов ЭЭГ по поверхности головы, перестройка которых, отражает доминирование тех или иных когнитивных процессов.

Практически все методы классификации предполагают сведение задачи классификации динамических данных к задаче классификации статических данных. Для такого сведения необходимо осуществить переход от временного ряда $\{x(t), t = \overline{1, T}\}$, описывающего сигнал, к некоторому вектору признаков v размерности M , характеризующему этот временной ряд.

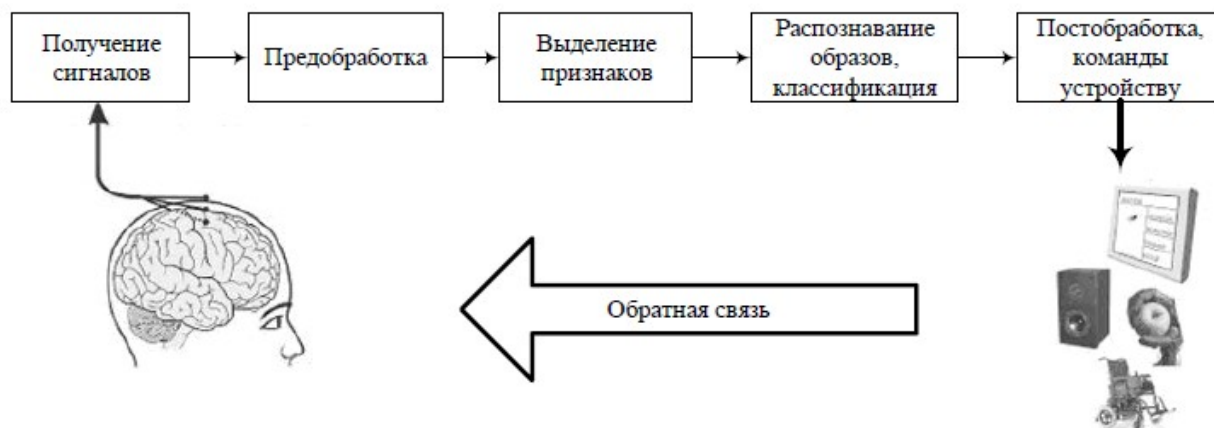


Рисунок 1 – Архитектура ИМК

В общем случае архитектура ИМК (рис. 1) включает следующие компоненты.

- 1) Регистрация электрической активности мозга (инвазивным или неинвазивным способом).
- 2) Предобработка данных (фильтрация от шумов и удаление компонентов ЭЭГ, не связанных с мозговой деятельностью – артефактов).
- 3) Выделение признаков ЭЭГ, характеризующих тип электрической активности мозга.
- 4) Классификация выделенных признаков (распознавание типа мыслительной деятельности).
- 5) Постобработка результатов классификации и генерация команд техническому устройству. Типы команд могут быть очень различными в зависимости от конкретного ИМК. Например, это может быть перемещение курсора на экране компьютера, включение/выключение света, движение механической рукой и т.п.
- 6) Обратная связь с пользователем (*neurofeedback*). Обратная связь используется для адаптации пользователя к ИМК. Это может быть визуальная, слуховая, тактильная или иная связь. В некоторых ИМК обратная связь отсутствует.

3 Методы выделения и классификации паттернов ЭЭГ

На начальном шаге осуществляется фильтрация сигналов от шумов, устранение артефактов или другие операции, позволяющие на ранних этапах отсеять неинформативные данные, используя априорно известную информацию о природе сигнала. Наиболее частыми видами артефактов [8] являются помехи от электрических полей, создаваемых устройствами передачи и эксплуатации промышленного электрического тока, а также физиологические артефакты, связанные с проявлениями жизнедеятельности организма (электромиограмма (ЭМГ), электроокулограмма (ЭОГ) и др.). Подавление артефактов осуществляют с помощью следующих устройств и методов.

- а) Фильтры нижних и высоких частот (устранение помех от электрических полей).
- б) Методы, основанные на *регрессионном анализе во временной или частотной областях*.
- в) Методы, основанные на линейном разложении сигнала ЭЭГ на составляющие компоненты, выделении артефакта (например, ЭОГ) и восстановлении ЭЭГ без компоненты артефакта. Для этого применяют *анализ главных компонент* (АГК) и *анализ независимых компонент* (АНК).

На следующем этапе происходит выделение характерных признаков (паттернов) ЭЭГ сигнала. Для чего применяются:

а) *Дискретное преобразование Фурье*. В качестве признаков в данном случае выступает спектральная мощность сигнала в “дельта” (2-4) Гц, “тета” (4-8) Гц, “нижнем альфа” (8-10) Гц, “верхнем альфа” (10-12) Гц, “бета” (12-35) Гц и “гамма” (35-50) Гц частотных диапазонах.

б) *Дискретное вейвлет преобразование*. В качестве признаков используют коэффициенты вейвлет-разложения, либо статистические параметры, характеризующие частотные поддиапазоны, например, среднее значение абсолютных величин коэффициентов, среднеквадратичное отклонение коэффициентов и др.

в) *Авторегрессионная модель*. ЭЭГ сигнал рассматривается как выход линейного фильтра с учетом воздействия белого шума. В качестве признаков выступают коэффициенты авторегрессии. Данный метод используется в *SMR-BCI* [5].

г) *Канонический корреляционный анализ*. В качестве характерных признаков используют коэффициенты корреляции сигналов, поступающих по нескольким каналам ЭЭГ, с сигналом стимула. Алгоритм применяется в *SSVEP-BCI* [9].

д) *Фильтр Калмана*. Фильтр Калмана представляет собой рекурсивный фильтр, оценивающий вектор состояния динамической системы в текущий момент времени, используя измерения на текущем такте работы и оценку состояния (в виде оценки состояния системы и оценки погрешности определения этого состояния) на предыдущем такте работы. В качестве характерных признаков при определении типа мыслительной деятельности в [10] выступают элементы матрицы перехода между состояниями.

е) *Методы «слепого» разделения сигналов (Blind Source Separation)*. Получая набор данных, в нашем случае временную последовательность, методы слепого разделения сигналов, например, *анализ независимых компонент* (АНК), предпринимают попытку найти такое (обычно линейное) преобразование сигнала, которое возвращает набор статистически независимых составляющих сигнала. В качестве характерных признаков используется спектральная мощность преобразованного сигнала в различных частотных диапазонах [6].

ж) *Фрактальный анализ*. Метод основан на определении фрактальной размерности хаотических компонент сигнала ЭЭГ. В [11] описано применение фрактального анализа для оценки амплитуды флуктуаций альфа-ритма при диагностировании эпилептической активности пациентов. Возможность применения фрактального анализа для получения вектора характерных признаков ЭЭГ сигнала в рамках интерфейса мозг-компьютер требует дополнительных исследований.

Для повышения точности кластеризации и классификации данных необходимо выделение наиболее информативных характерных признаков. Для решения данной задачи используют методы, позволяющие уменьшить размерность вектора характерных признаков:

а) *Генетический алгоритм*. Алгоритм позволяет выделить подмножество признаков, превосходящее по полезным свойствам полный набор признаков. Критерием полезности подмножества признаков служит точность классификации, достигаемая на данном подмножестве.

б) *Анализ главных компонент*. Метод позволяет найти такое ортогональное преобразование в новую систему координат (подпространство характерных признаков меньшей размерности), для которого выборочная дисперсия данных вдоль координат будет максимальной.

На этапе классификации данных с целью распознавания типа мыслительной деятельности применяют следующий математический аппарат.

а) *Линейный дискриминантный анализ (Linear Discriminated Analysis)*. Метод позволяет сформировать разделяющую гиперплоскость в m -мерном пространстве характерных признаков. Стоит отметить, что метод является одним из наиболее эффективных линейных методов классификации, применяемых в интерфейсе мозг-компьютер, и требует небольшое число обучающих выборок для получения достоверного вывода классификатора.

б) *Байесовский классификатор (Bayesian classifier)*. Метод представляет собой простой вероятностный классификатор, основанный на применении теоремы Байеса со строгими предположениями о независимости событий.

в) *Метод опорных векторов (Support vector machine)*. Пусть каждая точка в пространстве признаков принадлежит только одному из двух классов. Метод опорных векторов выполняет в пространстве признаков поиск такой разделяющей гиперплоскости, чтобы расстояние между двумя ближайшими точками, лежащими по разные стороны гиперплоскости, было максимальным [6].

г) *Метод k -средних (k -means)*. Метод стремится минимизировать суммарное квадратичное отклонение точек кластеров от центров этих кластеров. На каждой итерации алгоритма вычисляется центр масс каждого кластера, полученного на предыдущем шаге, затем векторы разбиваются на кластеры вновь в соответствии с тем, какой из новых центров оказался ближе по выбранной метрике.

е) *Искусственные нейронные сети*. В связи с тем, что характер ритмической активности ЭЭГ специфичен для каждого индивида, классификатор векторов характерных признаков ЭЭГ должен адаптивно настраиваться на индивидуальные особенности конкретного испытуемого, что можно достичь, применяя нейросетевой подход. Учитывая это, во многих работах для классификации используется многослойный персептрон. Для выделения кластеров в пространстве характерных признаков применяют сети с самоорганизацией на основе конкуренции [12]. Коррекция весов нейронов в ходе обучения выполняется по алгоритму Кохонена.

4 Направления и перспективы развития

Наиболее простыми в реализации и применении являются ИМК на основе компонента P300 и на основе анализа сенсомоторных ритмов. В настоящее время фирма *intendiX®* выпускает устройство *intendiX®SPELLER*, реализующее функции виртуальной клавиатуры на основе компонента P300 [13]. Также библиотеки, содержащие математические алгоритмы и методы, используемые в реализации ИМК указанных типов, включены в состав открытых программных проектов *EEGLab* [14], *BCI2000* [15] и *OpenVibe* [16], направленных на разработку и совершенствование методов анализа ЭЭГ.

К недостаткам ИМК на основе компонента P300 можно отнести необходимость предъявления внешнего стимула для работы интерфейса и необходимость концентрации внимания пользователя, что ограничивает условия комфортного применения. ИМК на основе анализа сенсомоторных ритмов не требуют наличия внешнего стимула, но уступают по скорости работы. В связи с чем, представляет интерес разработка прямых интерфейсов мозг-компьютер, предназначенных для непосредственного преобразования мысленного представления объекта в команду компьютера. Работы [6] и [17] показывают, что данная задача в принципе решается с удовлетворительной точностью. В работе [17] испытуемому предъявлялись изображения двух типов «дом/лицо». Затем на основе данных ЭЭГ делалась попытка определить, изображение какого типа видит испытуемый. Результаты показывают, что при формировании вектора характерных признаков с помощью генетического алгоритма точность классификации методом k -средних достигает 79%.

Развитием прямых интерфейсов мозг-компьютер может служить создание словаря терминов (понятий, визуальных образов), хорошо различимых с помощью характерных паттернов сигнала ЭЭГ, возникающих при мысленном представлении испытуемым данных

терминов. Сравнительно невысокая точность классификации может быть увеличена за счет применения новых подходов к выделению характерных признаков, например, фрактального анализа или методов «слепого» разделения сигналов. Что является предметом для дальнейших исследований.

Литература

1. А.А. Фролов, В.Ю. Рощин, Интерфейс мозг-компьютер. Реальность и перспективы // Нейронные сети. – 2006г. стр. 82-125.
2. А.Г. Трофимов, В.И.Скругин, Системы нейрокомпьютерного интерфейса. Обзор // Информационные технологии. - Журнал, 2011г. № 2, стр. 2-11.
3. Joseph E. O'Doherty, Mikhail A. Lebedev, Peter J. Ifft. Active tactile exploration enabled by a brain-machine-brain interface // J. Nature, 479(7372), pp. 228–231, 2012.
4. Brian N. Pasley, Stephen V. David, Nima Mesgarani, Adeen Flinker, Shihab A. Shamma. Reconstructing Speech from Human Auditory Cortex // PLoS Biology 10(1): e1001251, 2012.
5. Bao-Guo Xu, Ai-Guo Song, Pattern recognition of motor imagery EEG using wavelet transform // J. Biomedical Science and Engineering, 2008, 1, 64-67 pp.
6. David A. Peterson, Feature Selection and Blind Source Separation in an EEG-Based Brain-Computer Interface. - EURASIP Journal on Applied Signal Processing 2005:19, 3128–3140 pp.
7. А.Я. Каплан, Пластичность мозга и проблема интерфейса "мозг-компьютер": паттерны ЭЭГ как управляющие сигналы // Материалы XV Международной конференции по нейрокибернетике. Ростов-на-Дону. Издательство ЮФУ. – 2009. Том 2. 283 с.
8. Зенков Л.Р. Клиническая электроэнцефалография (с элементами эпилептологии). – М.: Медпресс-информ, 2004. – 368 с.
9. Zhonglin Lin, Changshui Zhang, Wei Wu, Xiaorong Gao, Frequency recognition based on canonical correlation analysis for SSVEP-based BCIs. // Biomedical Engineering, 2007, 54, pp 1172 – 1176.
10. А.Н. Omidvarnia, F. Atry, S.K. Setarehdan, and B.N. Arabi. Kalman filter parameters as a new EEG feature vector for BCI applications // In Proceedings of the 13th European Signal Processing Conference Eusipco2005. Citeseer, 2005.
11. Урицкий В.М. Фрактальная диагностика нарушения альфа-ритма при эпилепсии // Биофизика. – 1999. – Т. 44, №6. – стр. 1109-1114.
12. В.И. Скругин, А.Г. Трофимов, А.О. Роик, Р.А. Наумов, Алгоритм классификации сигналов ЭЭГ на основе анализа в частотно-временной области // «Наука и образование». – Электронный журнал. 2010г. №8 <http://technomag.edu.ru/>
13. <http://www.intendix.com/>
14. <http://sccn.ucsd.edu/eeglab/>
15. <http://www.bci2000.org/BCI2000/Home.html>
16. <http://openvibe.inria.fr/>
17. Е.А.Марченко, М.К. Бахнян, BCI на основе воображаемых образов и устранение избыточности ЭЭГ по высокочастотной составляющей. // Материалы XVI Международной конференции по нейрокибернетике. Том 2. Ростов-на-Дону. Издательство ЮФУ 2012. стр. 38-40.

НЕЧЕТКИЕ БАЗЫ ДАННЫХ И ЗАПРОСЫ

Валиуллин А.М.

Научный руководитель: к.т.н., доцент, Волосатова Т.М.

МГТУ им. Н.Э.Баумана, кафедра РК-6, Москва, Россия

FUZZY DATABASE SYSTEMS AND QUERIES

Valiullin A.M.

Supervisor: Dr., Prof., Volosatova T.M.

MSTU, Moscow, Russia

Аннотация

В статье рассматриваются вопросы обработки нечеткой информации, необходимость использования нечетких баз данных, способы хранения и извлечения такой информации в базах данных, описаны методы перевода нечетких запросов и модели построения "точных" запросов к классическим реляционным базам данных. Также нечеткие запросы могут быть описаны на расширенном языке FSQL для нечетких систем. Показана эффективность и результативность методов построения на приведенных в статье примерах.

Abstract

In this paper we examine problems of fuzzy data processing, necessity of using fuzzy databases, techniques used in the storage and retrieval of fuzzy data in a database, methods of translating fuzzy queries into precise queries for classical relational databases. Also fuzzy queries can be expressed in the Fuzzy SQL language. The efficiency and effectiveness of our approach is demonstrated by experimental results.

Введение

Все чаще в наши дни в запросах к реляционным базам данных, которые формулирует человек, часто присутствуют неточности и неопределенности. В целях увеличения применимости традиционных СУБД были введены новые методы для работы с неточной и неопределенной информацией. Использование теории нечетких множеств в базах данных привело к развитию нечетких баз данных.

Впервые проблема математического описания, методов и средств обработки данных в условиях нечеткости и неопределенности была описана в 1984 году. С тех пор было предложено много моделей описания и представления нечеткой информации в реляционных базах данных. Модель Баклиса-Петри [1] была первой моделью, которая отображала подобные связи в реляционной модели. Звиели и Чен первыми предложили подход [2] к объединению нечеткой логики с ER (Entity-Relationship) моделью. Эта модель разрешает нечеткие атрибуты в сущностях и связях. Чен и Керре [3] представили нечеткое расширение нескольких основных EER (Extended Entity-Relationship) концепций. Чодхри, Мойн и Рунденштейнер предложили метод [4] для построения нечетких реляционных баз данных согласно расширениям ER модели, предложенные Звиели и Ченом. Галиндо, Уррутиа и Пиаттини [5] описали использование нечеткой EER модели для моделирования базы данных и представления смоделированной нечеткой информации в реляционных базах.

1 Неопределенность, неточность, нечеткость и несоответствие данных

Различают 4 основных типа дефекта в описании информации: неопределенность, неточность, нечеткость и несоответствие. Все эти понятия имеют разный смысл. Различия между неопределенностью и неточностью детально описаны в трудах Дюбуа и Прада (1988).

Неопределенность означает отсутствие достоверной информации о состоянии объекта. Неопределенность возникает из-за неполноты знаний, относящихся к некоторому событию или наличию свойства какого-либо объекта. Например: «Завтра в 17:00 будет идти дождь.» Как мы знаем, прогноз погоды на завтра может сбыться, а может и не сбыться, то

есть с какой-то долей вероятности дождя может завтра и не быть. Описания такого рода носят неопределенный характер.

Неточность относится к способу описания самого события, свойства. Например: «Возраст Михаила между 20 и 25 годами.» Это означает, что возраст Михаила с равной вероятностью может быть равен целым числам от 20 до 25. Такие описания носят дефект неточности.

Нечеткое высказывание содержит расплывчатое описание события или свойства. Например: «Михаил молодой исследователь.» Возраст Михаила описывается с помощью лингвистического термина «молодой». Нужно заметить, что значение этого термина зависит от контекста высказывания, так как большая бабочка меньше, чем небольшой слон.

Несоответствие информации означает, что мы имеем высказывания, которые не имеют пересечения. Например: «Михаилу 25 лет.» и «Возраст Михаила между 28 и 30 годами.» В этом случае мы не можем описать значение возраста Михаила, который будет удовлетворять этим двум высказываниям. Решение таких проблем было описано в работах Дюбуа, Ланга и Прадо (1992).

Итак, неопределенность означает, что хорошо описанное событие или явление может произойти, а может и не произойти, тогда, как неточность связана с неточным описанием и плохим определением события, явления или переменной. Нечеткость появляется с использованием лингвистического термина. Несоответствие возникает в ситуациях, когда нет возможности определения события, которое бы удовлетворяло различным источникам.

2 Нечеткие запросы

Нечеткие запросы - перспективное направление в современных системах обработки информации, они появились в связи с необходимостью смягчить булеву логику в реляционных базах данных. Этот инструмент дает возможность формулировать запросы, встречающиеся в повседневной жизни «Вывести список недорогих иномарок в Москве», что невозможно при использовании стандартного механизма запросов.

Механизм нечетких запросов к реляционным базам данных базируется на теории нечетких множеств, которая была впервые предложена Лотфи Заде.

Продемонстрируем ограниченность четких запросов на следующем примере.

Пример. Пусть требуется получить сведения о менеджерах по продажам не старше 25 лет, у которых сумма годовых сделок превысила 200 т.р. по такому-то региону. Данный запрос можно записать на языке SQL следующим образом:

```
SELECT * FROM Managers WHERE (Age <= 25 AND Sum > 200000);
```

Менеджер по продажам 26 лет с годовой суммой продаж в 400 т.р., или 19 лет с суммой в 198 т.р. не попадут в результат запроса, хотя их характеристики почти удовлетворяют требованиям запроса.

Нечеткие запросы помогают справиться с подобными проблемами «пропадания» информации.

3 Функции принадлежности

Существуют различные пути решения проблем нечетких запросов. В данной работе более подробно рассмотрим построение нечетких запросов с помощью введения функций принадлежности. Задача поиска информации в таких запросах решается с помощью формализованного описания множества значений термов логического выражения. Моделью таких структур является лингвистическая переменная и функция принадлежности [6].

Функцией принадлежности называется функция, которая позволяет вычислить степень принадлежности произвольного элемента x множества X к нечеткому множеству F .

Для примера формализуем нечеткое понятие «Возраст сотрудника компании». Это и будет название соответствующей лингвистической переменной. Зададим для нее область определения $X = [18; 70]$ и три лингвистических термина - "Молодой", "Средний", "Выше среднего". Последнее, что осталось сделать - построить функции принадлежности для

каждого лингвистического термина. Выберем трапециевидные функции принадлежности со следующими координатами:

- "Молодой" = [18, 18, 28, 34],
- "Средний" = [28, 35, 45, 50],
- "Выше среднего" = [42, 53, 70, 70].

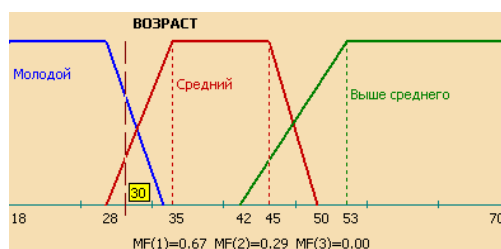


Рисунок 1 - Функции принадлежности термов переменной «Возраст сотрудника»

Теперь можно вычислить степень принадлежности сотрудника к каждому из нечетких множеств:

$$\begin{aligned} \mu[\text{«Молодой»}](30) &= 0.67 \\ \mu[\text{«Средний»}](30) &= 0.29 \\ \mu[\text{«Выше среднего»}](30) &= 0 \end{aligned}$$

Основное требование при построении функций принадлежности – значение функций принадлежности должно быть больше нуля хотя бы для одного лингвистического термина.

Вернемся к примеру с менеджерами о продажах. Для простоты предположим, что вся необходимая информация находится в одной таблице со следующими полями: ID - номер сотрудника, AGE - возраст и SUM - годовая сумма сделок:

ID	AGE	SUM
1	23	120 500
2	25	164 000
3	28	398 000
4	31	489 700
5	33	251 900

Лингвистическая переменная «Возраст» была задана ранее. Определим еще одну лингвистическую переменную для поля SUM с областью определения $X = [0; 600000]$ и аналогично построим для них функции принадлежности:

- "Малая" = [0, 0, 0, 200000],
- "Средняя" = [90000, 180000, 265000, 330000],
- "Большая" = [300000, 420000, 600000, 600000].



Рисунок 2 - Функции принадлежности термов переменной «Годовая сумма сделок»

К такой таблице можно делать нечеткие запросы. Например, получить список всех молодых менеджеров по продажам с большой годовой суммой сделок:

*SELECT * FROM Managers WHERE (age = "Молодой" AND sum = "Большая");*

Рассчитав для каждой записи агрегированное значение функции принадлежности μ , получим результат нечеткого запроса:

ID	AGE	SUM	MF
3	28	398 000	0,82
4	31	489 700	0,50

Записи 1, 2, 5 не попали в результат запроса, т.к. для них значение функции принадлежности равно нулю. Записей, точно удовлетворяющих поставленному запросу ($\mu = 1$), в таблице не нашлось. Менеджер по продажам 28 лет и годовой суммой 398000 соответствует запросу с функцией принадлежности 0,82. Аналогичный четкий запрос мог бы быть сформулирован, например, так:

*SELECT * FROM Managers WHERE (age <= 28 AND sum >= 420000);*

Его результат является пустым. Однако если мы немного расширим рамки возраста в запросе, то рискуем упустить других сотрудников с чуть более большим или меньшим возрастом. Поэтому можно сказать, что нечеткие запросы позволяют расширить область поиска в соответствии с изначально заданными человеком ограничениями.

Заключение

В данной работе показан эффективный метод обработки нечетких запросов с помощью введения функций принадлежности. Существует много различных методов задания функций принадлежности. Выбор того или иного метода зависит от задачи, существующей ситуации (например, наличия экспертов) и других параметров.

Реляционная модель представления и реализации нечетких моделей может стать связующим звеном для интеграции с существующими реляционными базами данных. Также в настоящее время активно развивается расширение языка SQL, так называемый стандарт FSQL. Расширенный язык FSQL позволит получить новый инструмент для извлечения новых ранее недоступных данных и знаний.

Литература

1. B.P. Buckles, F.E. Petry (1982). A fuzzy representation of data for relational databases. Fuzzy Sets and Systems no. 7, pages 213-226.
2. A. Zvieli, P. Chen (1986). ER modelling and fuzzy databases. In Proceedings of the Second International Conference on Data Engineering. pages 320-327, LA, USA.]
3. G.Q. Chen, E.E. Kerre (1998). Extending ER/EER concepts towards fuzzy conceptual data modelling. Proceedings of the IEEE International Conference on Fuzzy Systems. Pages 1320-1325, Anchorage, AK, USA.]
4. N. Chaudhry, J. Moynes, E. Rundensteiner (1994). A design methodology for databases with uncertain data. Proceedings of the Seventh International Working Conference on Scientific and Statistical Database Management. Pages 32-41, Charlottesville, VA, USA.
5. J. Galindo, A. Urrutia, M. Piattini (2006). Fuzzy Databases: Modelling Design and Implementation. Hershey, USA: IDEA Group.] Введение в математическое моделирование. Учебное пособие. Под ред. П. В. Трусова. — М.: Логос, 2004. 200 с.
6. Технологии анализа данных [Электронный ресурс]. М., 1995-2013. URL: <http://www.basegroup.ru>. (Дата обращения: 10.02.2013).

ПАРАЛЛЕЛЬНЫЙ АЛГОРИТМ РЕШЕНИЯ СИСТЕМ АЛГЕБРАИЧЕСКИХ УРАВНЕНИЙ С ЛЕНТОЧНОЙ МАТРИЦЕЙ КОЭФФИЦИЕНТОВ.

Желдаков А.В.

Научный руководитель: к.т.н. доцент Федорук В.Г.
МГТУ им. Н.Э.Баумана, кафедра РК6, Москва, Россия

PARALLEL ALGORITHM OF SOLVING A LINEAR SYSTEM OF ALGEBRAIC EQUATIONS WITH THE TAPE COEFFICIENT MATRIX.

Zheldakov A.V.

Supervisor: Ph.D. associate professor Fedoruk V.G.
MSTU, Moscow, Russia

Аннотация

В статье предложен параллельный алгоритм решения систем линейных алгебраических уравнений. Приведена схема алгоритма для вычислительной системы типа мультимикропроцессор. Исследована эффективность алгоритма на различных классах задач. В заключении отмечены достоинства и недостатки алгоритма.

Abstract

In this article was offered a parallel algorithm for solving systems of linear algebraic equations. A scheme of the algorithm for multiprocessor system was offered. The efficiency of the algorithm was investigated for different classes of tasks. In conclusion the advantages and disadvantages of the algorithm were given.

Введение

При решении задач математической физики численными методами, такими как метод конечных элементов [1] (МКЭ), метод конечных разностей [2] (МКР) часто возникает необходимость нахождения решения систем линейных алгебраических уравнений [3] (СЛАУ) высоких порядков. С распространением мультимикропроцессорных систем появилась проблема адаптации точных алгоритмов решения СЛАУ на данную архитектуру. В работе рассмотрен алгоритм параллельного решения СЛАУ с многодиагональной матрицей коэффициентов на системе типа мультимикропроцессор.

Предлагаемый алгоритм реализует метод Гаусса [3] и состоит из трёх основных этапов: трансформация задачи, прямой ход алгоритма, обратный ход алгоритма.

1 Постановка задачи

Задача нахождения решения СЛАУ может быть представлена в виде

$$Ax = c, \quad (1)$$

где A - квадратная ленточная матрица коэффициентов с шириной ленты L , x - вектор-столбец неизвестных и c - вектор-столбец свободных членов. Под решением задачи понимают такую совокупность чисел u_1, u_2, \dots, u_n , которая при подстановке на место неизвестных x_1, x_2, \dots, x_n обращает все уравнения этой системы в тождества. Для большинства практических задач определитель матрицы A отличен от 0, следовательно, задача (1) имеет единственное решение.

2 Трансформация задачи

Пусть N - число диагоналей, содержащих отличные от 0 элементы матрицы A (ненулевые диагонали), $d_k, k \in [1:N]$ - вектор, содержащий элементы k -ой ненулевой диагонали. Размерность векторов d_k постоянна и равна n . Индексы компонентов векторов $d_k, k \in [1:N]$ соответствуют индексам строк матрицы A в которых они находятся. Если какой либо компонент векторов $d_k, k \in [1:N]$ выходит за пределы матрицы A , то данные компоненты тождественно равны 0. Будем считать, что диагональ матрицы A , представленная вектором d_l , не содержит нулевых элементов.

Введем смещение S – натуральное число, лежащее в интервале $[0, N - 1]$. Значение S определяется по формуле

$$S = \max_{i \in [1:n]} (i) - 1, \quad A_{i,1} \neq 0.$$

Расширим вектор x задачи (1), добавив в его начало S переменных, а также введём в систему S дополнительных уравнений:

$$\begin{cases} x_1 = 0, \\ x_2 = 0, \\ \dots \\ x_S = 0, \end{cases}$$

где x_1, x_2, \dots, x_S неизвестные, добавленные в (1). Введённым уравнениям, присвоим номера $i, i \in [n + 1: n + S]$, тогда модифицированная задача примет вид

$$A'x' = c', \quad (2)$$

где A' – матрица коэффициентов модифицированной задачи, x' – вектор-столбец неизвестных модифицированной задачи и c' – вектор-столбец свободных членов модифицированной задачи. Размерность матрицы A' составит $(n + S) \times (n + S)$, размерности векторов x' и c' будут равны $(n + S)$.

Далее к строкам матрицы A' с индексом $i \in [1:S]$ прибавим строки матрицы A' , полученные после модификации (1), по следующему правилу – к строке с индексом $i \in [1:S]$ прибавляется строка с индексом $j = i + n$.

На основе теоремы Кронекера – Капелли легко показать существование и единственность решения задачи (2) при условии существования и единственности решения задачи (1).

3 Представление данных

В задаче (1), как правило, матрица A является сильно разреженной, поэтому нецелесообразно хранить все элементы матрицы. Далее матрица A будет полностью определяться следующими величинами:

- S – смещение матрицы A ,
- n – размерность одной строки матрицы A ,
- L – ширина ленты матрицы A ,
- N – число ненулевых диагоналей матрицы A ,
- $d_k, k \in [1:N]$ – векторы размерностью n , содержащие элементы k -ой ненулевой диагонали матрицы A ,
- p – вектор размерности N , где $p_i, i \in [1:N]$ равен номеру i -ой ненулевой диагонали, отсчитанному от d_1 .

Вектора c и x сохраняют свое представление, указанное в (2).

Для матрицы A' дополнительно введем вектора $b_t, t \in [1:S]$. Каждый из этих векторов имеет размерность L и содержит отличные от 0 элементы строки матрицы A' с номером $(n + t), t \in [1:S]$.

4 Прямой ход алгоритма

Сначала вычтем первые S уравнений задачи (2) из последних S уравнений системы по следующему правилу - из уравнения с номером $j \in [n + 1: n + S]$ вычитается уравнение с номером $i = j - n$.

После чего начинается итерационный процесс, состоящий из $(n - S)$ итераций. Каждая итерация состоит из двух шагов. На первом шаге компоненты векторов $b_t, t \in [1:S]$ и c' вычисляются по формулам

$$\begin{aligned} b_{t,j}^i &= b_{t,j}^{i-1} - K_i \cdot d_{i+s,j}, \\ c_{n+t}^i &= c_{n+t}^{i-1} - K_i \cdot c_i', \\ K_i &= \frac{b_{t,1}^{i-1}}{d_{1,i+s}}, \end{aligned}$$

$$j = p_m, m \in [1:N], i \in [1:n - S], t \in [1:S]. \quad (3)$$

В (3) индекс i - номер итерации, j это номер ненулевой диагонали в векторе \mathbf{p} , стоящий на позиции m , $b_{t,j}^i$ - значение j -го компонента вектора \mathbf{b}_t на итерации с номером i , c_{n+t}^i - значение компоненты вектора \mathbf{c} с индексом $(n + t)$ на итерации с номером i .

На втором шаге итерации выполняется циклический сдвиг векторов $\mathbf{b}_t, t \in [1:S]$ на l элемент влево.

После выполнения $(n - S)$ итераций получим полностью заполненную подматрицу размерности $S \times S$. Перед выполнением следующего этапа требуется синхронизировать работу программы. Для последних S уравнений задачи (2) выполним прямой ход алгоритма Гаусса.

5 Схема метода для системы типа мультипроцессор

Опишем схему метода решения задачи на вычислительной системе с данной архитектурой.

- 1) Процессор Q_1 назначается главным (*master*-процессор) все остальные вспомогательные (*slave*-процессоры). На *master*-процессоре запускаем основной процесс алгоритма.
- 2) Загружаем в оперативную память системы данными, содержащими исходную матрицу A и вектор \mathbf{c} .
- 3) На *master* - процессоре выполняем модификацию задачи (1), согласно изложенному выше алгоритму.
- 4) Запускаем p потоков управления, каждому из которых передадим на обработку $z = \lfloor \frac{S}{p} \rfloor$ векторов $\mathbf{b}_t, t \in [1:S]$. Если S не кратно p , то последний поток управления получит меньше векторов $\mathbf{b}_t, t \in [1:S]$ чем остальные. Распределение потоков управления по процессорам выполняется планировщиком операционной системы. При обработке данных участвуют как *master*-процессор, так и *slave*-процессоры.
- 5) Каждый поток управления по формуле (3) производит модификацию векторов $\mathbf{b}_t, t \in [i \cdot z + 1 : (i + 1) \cdot z], i$ - номер потока управления.
- 6) На *master*-процессоре выполним прямой ход стандартного алгоритма Гаусса для последних S уравнений модифицированной задачи.
- 7) На *master*-процессоре выполняем обратный ход алгоритма Гаусса и завершаем работу программы.

6 Программная реализация

Алгоритм был реализован на языке высокого уровня C++ с использованием стандартных библиотек многопоточного программирования операционной системы UNIX.

Приведем результаты работы программы для некоторых СЛАУ.

$$n = 100000, S = 400, N = 5.$$

Кол-во ядер	1	2	3	4
T_{min} (сек.)	8,12	4,42	3,03	2,09
S_{min}	1,00	1,83	2,67	3,87

$$n = 100000, S = 700, N = 5.$$

Кол-во ядер	1	2	3	4
T_{min} (сек.)	31,40	16,22	11,17	8,19
S_{min}	1,00	1,93	2,81	3,83

$n = 3000000, S = 400, N = 5.$

Кол-во ядер	1	2	3	4
$T_{\text{мп}}$ (сек.)	243	127	86	64
$S_{\text{мп}}$	1,00	1,910	2,820	3,790

$n = 3000000, S = 700, N = 5.$

Кол-во ядер	1	2	3	4
$T_{\text{мп}}$ (сек.)	957	492	322	254
$S_{\text{мп}}$	1,00	1,94	2,97	3,76

Теперь для $n = 3000000, N = 5$ исследуем зависимость $S_{\text{мп}}$ от S при использовании системе типа мультипроцессор состоящей из 4 ядер.

S	16	32	64	128	256	512	1024	2048
$T_{\text{мп}}$ (сек.)	0,001	0,095	3,15	7,39	75,11	93	1192	11260
T_1 (сек.)	0,0037	0,254	11,28	26,17	266,93	328	2250	18353
$S_{\text{мп}}$	3,70	3,58	3,58	3,55	3,55	3,52	1,88	1,63

Данные результаты полностью согласуются с аналитической оценкой эффективности алгоритма. Так с ростом ширины ленты эффективность алгоритма падает, а при одинаковой ширине ленты ускорение практически не зависит от размерности исходной задачи n .

Заключение

Отметим основные преимущества алгоритма:

- Простота программной реализации.
- Высокие показатели ускорения для задач с малым смещением матрицы коэффициентов.
- Позволяет экономить память вычислительной системы.
- Алгоритм может быть легко адаптирован под различные SIMT (Single Instruction Multiple Threads) системы с общей памятью.
- Поскольку каждый поток управления оперирует изолированным набором векторов $\mathbf{b}_t, t \in [1:S]$, то нет необходимости множественного доступа по записи к исходным данным.

Отметим основные недостатки алгоритма.

- Возможности распараллеливания зависят от параметра S матрицы A .
- Падение ускорения с ростом S матрицы A .
- При выполнении этапа модификации необходимо выделение дополнительной памяти.
- В данной реализации возможно несбалансированность нагрузки процессоров вычислительной системы. Но при больших значениях S несбалансированность будет минимальна.

Литература

1. В. А. Ильин, Э. Г. Позняк Линейная алгебра, М.: Наука — Физматлит, 1999, 297с.
2. В. А. Ильин, Г. Д. Ким Линейная алгебра и аналитическая геометрия, М.: ТК Велби, Изд-во Проспект, 2007, 400с.
3. Ортега Дж. Введение в параллельные и векторные методы решения линейных систем / пер. с англ. — М.: Мир, 1991, 541с.
4. Б. Эккель. Философия C++. Введение в стандартный C++. 2-е изд. —СПБ Питер, 2004, 571с.
5. Керниган Б, Ритчи Д. Язык программирования Си. — 2-е изд. — М.: Вильямс, 2007, 304с.
6. POSIX Threads Programming. URL: <https://computing.llnl.gov/tutorials/pthreads>. Дата обращения 19.10.2012.

СТРУКТУРНО-ФУНКЦИОНАЛЬНЫЙ АНАЛИЗ ДЕЯТЕЛЬНОСТИ ПО ИЗГОТОВЛЕНИЮ УСТРОЙСТВА «ТОЛЩИНОМЕР МТП-01»

Рогова В.С.

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

STRUCTURAL AND FUNCTIONAL ANALYSIS OF MANUFACTURING UNIT «THICKNESS MTG-01»

Rogova V.S.

Supervisor: Docent, Ph.D. Vlasov A.I.
MSTU, Moscow, Russia

Аннотация

В статье рассматривается деятельность по изготовлению устройства «Толщиномер МТП-01». Подробно исследован технологический процесс сборки устройства, посредством структурно-функциональной модели технологического процесса в нотации IDEF. Кратко представлена концептуально-абстрактная модель изготовления толщиномера. В заключении представлены рекомендации по модернизации технологического процесса.

Annotation

The paper describes the activities for the production of the device "Thickness MTG-01." Studied in detail the assembly process devices, through structural and functional model of the technological process in a notation IDEF. Summarizes the conceptual and abstract model of manufacturing thickness gauge. In conclusion one can say that make recommendations on the modernization process.

Введение

Работа посвящена исследованию ТП изготовления устройства «Толщиномер покрытий магнитный МТП-01» и движения конструкторско-технологической документации по этапам производственного процесса (ПП).

Объектом исследования является процесс изготовления устройства «Толщиномер покрытий магнитный МТП-01» (в дальнейшем толщиномер), включая процесс документального сопровождения ПП.

Актуальность работы заключается в необходимости разработки и внедрения системы электронного документооборота (ЭД) с целью обеспечения прозрачности и контролируемости производственных процессов на каждом этапе жизненного цикла изделия.

Предпосылками внедрения средств автоматизации являются:

- существует большой документопоток входящих, исходящих и внутренних (служебных) документов, рассмотрение которых серьезно увеличивает сроки выполнения работ;
- совещаний у руководителей разного уровня становится больше и их продолжительность затягивается;
- оперативность принятия и выполнения решений низкая и постоянно снижается;
- нахождение виновных в нарушении исполнительской дисциплины становится проблематичным;
- количество форм для отчетности растет;
- существует проблема утечки информации и нарушения коммерческой тайны.

Игнорирование вышеназванных ситуаций или попытки решить эти задачи без помощи автоматизированных программных систем приведет к еще большему усугублению проблем.

Функции современных систем ЭД. Системы автоматизации документооборота, предлагаемые на рынке, в основном, отечественных производителей либо интеграторов зарубежного программного обеспечения. Среди известных СЭД российских поставщиков можно назвать: Дело 12.0 (2012, <http://www.bytemag.ru>), БОСС-Референт 3.2.2 (2009,

<http://www.boss-referent.ru>), CompanyMedia 3.0 (2004, <http://education.intertrust.ru>), DIRECTUM 4.2 (2006, <http://www.directum.ru>), DOCUMENTUM 6.7 (2011, <https://community.emc.com>), DocsVision 5 (2012, <http://www.docsvision.com>), ЕВФРАТ-Документооборот v.14.5 (2009, <http://www.infocomdv.ru>), Optima-Workflow 1.18 (2006, <http://software.optima.ru>), LanDocs 2 (2000, <http://www.crn.ru>), Lotsia PDM Plus 5.10 (2012, <http://www.plm-conference.com>).

Функции, предлагаемые СЭД своим пользователям, весьма разнообразны. В первом приближении их можно разделить на такие категории:

- хранение и поиск документов;
- поддержка канцелярии;
- маршрутизация и контроль исполнения документов;
- аналитические отчеты;
- информационная безопасность;
- дополнительные (специфические) функции.

Централизованное хранение документов – едва ли не единственная цель перехода на электронный документооборот для маленьких компаний. В связи с этим стоит обратить внимание на поставщика хранилища данных, используемого в той или иной системой ЭД. Могут использоваться [1]:

- хранилища Lotus Notes/Domino (например, БОСС-Референт 3.2.2 (2009, <http://www.boss-referent.ru>), CompanyMedia 3.0 (2004, <http://education.intertrust.ru>));
- собственные форматы хранения данных (ЕВФРАТ-Документооборот v.14.5 (2009, <http://www.infocomdv.ru>));
- Microsoft SQL Server в различных редакциях (Дело 12.0 (2012, <http://www.bytemag.ru>), DIRECTUM 4.2 (2006, <http://www.directum.ru>), DocsVision 5 (2012, <http://www.docsvision.com>), LanDocs 2 (2000, <http://www.crn.ru>) и др.);
- Oracle (ДОК ПРОФ 2.0 (2008, <http://www.cnews.ru>) и др.);
- одновременная поддержка MS SQL и Oracle (Дело 12.0 (2012, <http://www.bytemag.ru>), ЕВФРАТ-Документооборот v.14.5 (2009, <http://www.infocomdv.ru>) и др.);
- другие СУБД.

Среди функций для поиска документов различают [1]:

- поиск по атрибутам (полям) документов;
- поиск по вложенным в документы файлам (полнотекстовый поиск);
- сложный поиск (с использованием логических операций).

При выборе СЭД потребителям приходится искать компромиссное решение, по максимуму удовлетворяющее следующим критериям [1]:

- обеспечение требуемой функциональности с возможностью дальнейшей расширяемости системы;
- минимальная совокупная стоимость владения и быстрая окупаемость системы;
- достаточный уровень технической поддержки;
- зарекомендовавший себя производитель с реальными внедрениями;
- учет отечественной законодательной базы;
- субъективные предпочтения заказчика.

Для выбора системы ЭД для внедрения на предприятии, рекомендуется предпринять ряд следующих шагов [1]:

- оценить совокупную стоимость владения выбранных систем, обратите внимание на стоимость ПО сторонних разработчиков, необходимого для работы системы ЭД;
- изучить схемы лицензирования, предлагаемые поставщиками системы ЭД, и выберите наиболее оптимальные для себя варианты;
- проанализировать с комплексом услуг по оказанию технической поддержки, оцените самого поставщика системы ЭД по заявленным им внедрениям;
- установить демо-версию системы, изучите демонстрационные материалы (видеоролики), предлагаемые поставщиками системы ЭД;

- выделить функциональные возможности анализируемых систем, которые являются критически важными с точки зрения законодательства, совместимости с установленным ПО, удобства работы сотрудников и т.п.

Окончательное решение в пользу выбора той или иной системы документооборота может быть получено, например, путем средневзвешенного анализа оценок, за соответствие перечисленным критериям. Проведя обзор критериев современных систем ЭД можно сделать вывод, что ни одно из рассматриваемых решений не обеспечивает сопровождение технологического документооборота, который составляет основу контроля, анализа, оперативного управления и планирования производственной деятельности. Поэтому необходимо решить эту проблему собственной разработкой, которая обеспечит сопровождение технологической документации.

Целью работы является разработка структурно-функциональной модели ТП изготовления толщиномера.

Для достижения заявленных целей в работе предусматривается решение следующего комплекса задач:

- разработка контекстной диаграммы и декомпозиции ТП изготовления устройства, выполненных в нотации IDEF0;
- разработка декомпозиций основных операций ТП изготовления устройства, выполненных в нотации IDEF3;
- представление потоков данных ТП в нотации DFD;
- разработка диаграммы дерева узлов ТП с полной иерархией;
- разработка FEO-диаграммы ТП;
- проведение оптимизации модели по временному, стоимостному и UDP критериям;
- заключение о соответствии модели реальному ТП и выработка рекомендаций по оптимизации ТП;

Точка зрения, с которой разрабатывалась модель – точка зрения **главного технолога**. Для этой точки зрения характерно описание ТП сборки устройства без вникания в подробности других сфер деятельности предприятия.

Методы, используемые для решения поставленной задачи: структурно-функциональная декомпозиция ТП изготовления устройства в нотациях IDEF0 и IDEF3, элементы реляционной алгебры и теории обработки данных.

Источниками для создания модели послужили:

- курс лекций по курсу ТПЭС за 6 семестр [2];
- материалы курсовой работы по курсу ТПЭС за 7 семестр [3];
- документация на толщиномер [4];
- государственный стандарт Р 50.1.028-2001 «Методология функционального моделирования» [5].

Назначение модели: выявление наиболее слабых и проблемных мест ТП изготовления устройства, формализация движения документооборота по этапам производственного процесса, с целью его дальнейшей автоматизации.

1 Разработка концептуально-абстрактной модели процесса изготовления устройства «Толщиномер МТП-01»

Концептуально-абстрактная модель изготовления толщиномера представлена на рисунке 1.



Рисунок 1 – Концептуально-абстрактная модель процесса изготовления устройства «Толщиномер МТП-01»

Спецификация модели представлена в таблице 1.

Таблица 1 – Спецификация концептуально-абстрактной модели процесса изготовления устройства «Толщиномер МТП-01»

Ключевое слово (понятие)	Графический образ понятия	Описание свойств понятия
Основное понятие		Процесс производства толщиномера МТП-01
Воздействие		Воздействие на основное понятие либо результат всего ТП изготовления прибора
Связь		Связь между частями модели

Концептуально-абстрактная модель является исходной моделью. Ключевыми словами являются следующие: основное понятие – «Изготовление толщиномера МТП-01», воздействия – «Комплектующие», «Оборудование», «Персонал», «Расходные материалы», «ГОСТы, ОСТы», «ТЗ», «Производство» и связи между ними.

Важным свойством (и одновременно – ограничением) является то, что все ветви модели однородны, нет разных обозначений для качественно различных сущностей. Это сразу накладывает ограничение на характер информации – пытаться описать бизнес-процесс с помощью Mind Map бесполезно, лучше использовать специализированные графические нотации – IDEF0, IDEF3, DFD. На Mind Map хорошо ложится первозданный хаос при первичном анализе. Поэтому далее разработаем структурную модель ТП изготовления толщиномера.

2 Разработка контекстной диаграммы ТП

На верхнем уровне декомпозиции (рис. 2) представлена общая модель изготовления толщиномера в виде контекстной диаграммы в нотации IDEF0. На входе модели комплектующие. На основании ГОСТов, ОСТов и ТЗ с помощью персонала, оборудования и расходных материалов создается конечный продукт: годная продукция или брак.

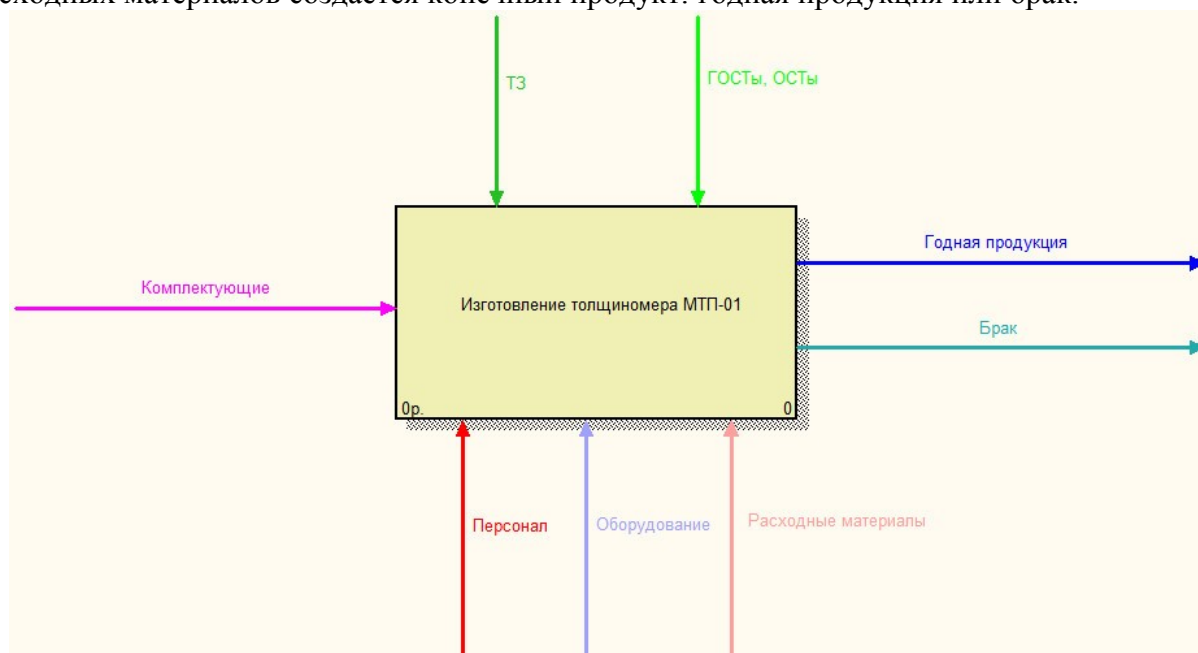


Рисунок 2 – Контекстная диаграмма в нотации IDEF0 «Изготовление тощиномера МТП-01»

Спецификация контекстной диаграммы представлена в таблице 2.

Таблица 2 – Спецификация контекстной диаграммы изготовления толщиномера

Input Arrow(s) of “Изготовление толщиномера МТП-01” Activity	
Name	Definition
Комплектующие	ДПП, ИЭТ, корпус, винты
Output Arrow(s) of “Изготовление толщиномера МТП-01” Activity	
Name	Definition
Годная продукция	Изделие, прошедшее настройку и контроль
Брак	Изделие, не прошедшее контроль
Control Arrow(s) of “Изготовление толщиномера МТП-01” Activity	
Name	Definition
ТЗ	Исходный документ для разработки изделия
ГОСТы, ОСТы	Документы государственного стандарта и документы отраслевого стандарта
Mechanism Arrow(s) of “Изготовление тощиномера МТП-01” Activity	
Name	Definition
Персонал	Монтажник, сборщик
Оборудование	Паяльная станция, пинцет, мультиметр, блок питания, бокорезы, отвертка
Расходные материалы	Припой, клей, канифоль

На верхнем уровне декомпозиции модель ТП изготовления устройства «Толщиномер МТП-01» представлена в виде контекстной диаграммы, выполненной в нотации IDEF0, которая отражает взаимодействие исследуемого процесса с внешней средой. В результате

выполнения процесса на основе преобразования входных воздействий, которыми являются комплектующие, формируются результаты, представляющие собой годную продукцию и брак. Данный процесс осуществляется под управлением ГОСТов, ОСТов и ТЗ, определяющих работу предприятия на территории РФ. Работа производится с помощью оборудования, персонала и расходных материалов.

Далее проведем декомпозицию процесса до атомарного уровня для обеспечения прозрачности процесса изготовления толщиномера.

3 Первый уровень декомпозиции процесса изготовления устройства «Толщиномер МТП-01»

На первом уровне декомпозиции (рис. 3) модель изготовления устройства «Толщиномер МТП-01» представляет собой четыре взаимосвязанные работы, представленные в нотации IDEF0: «Распаковка ИЭТ», «Сборка электронной ячейки», «Контроль и настройка», «Установка в корпус». Все эти работы выполняются на оборудовании с привлечением персонала под управлением ГОСТов, ОСТов и ТЗ.

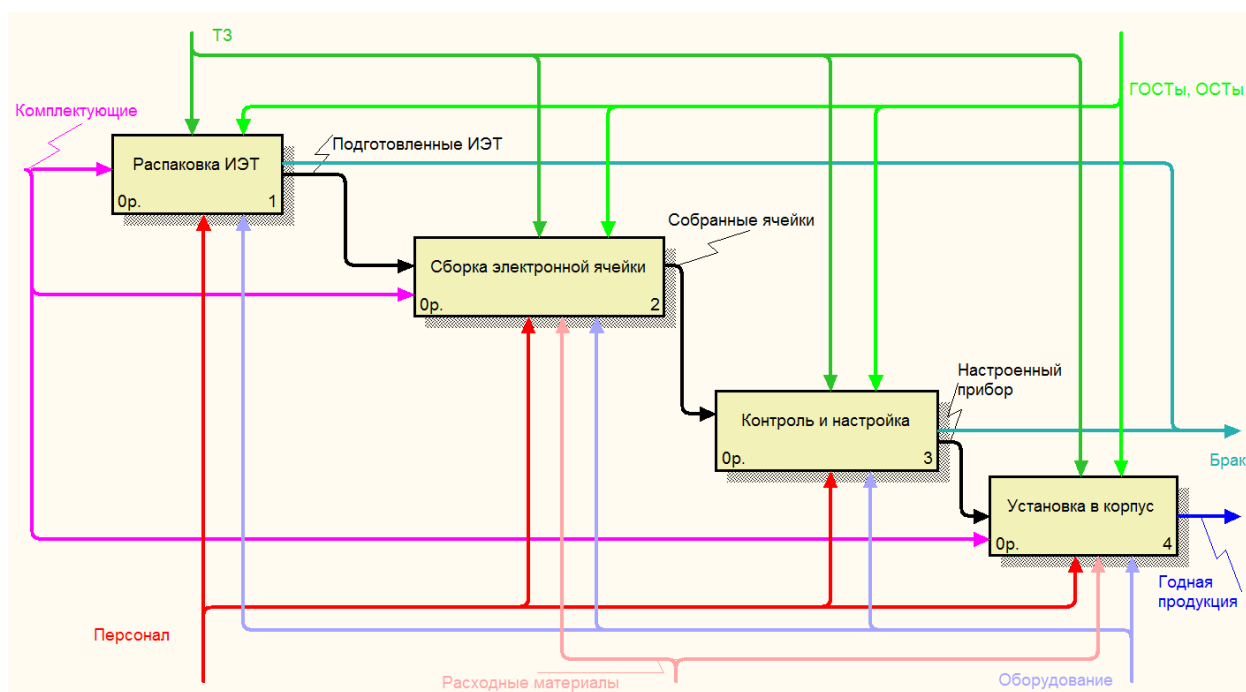


Рисунок 3 – Первый уровень декомпозиции процесса изготовления устройства «Толщиномер МТП-01»

Работа «Распаковка ИЭТ» преобразует входные воздействия в виде комплектующих под управлением ГОСТов, ОСТов и ТЗ с помощью персонала и оборудования. Результатом работы является либо брак, либо подготовленные ИЭТ, которые поступают на вход следующей работы «Сборка электронной ячейки». Данная работа преобразует комплектующие и подготовленные ИЭТ под управлением ГОСТов, ОСТов и ТЗ посредством персонала, оборудования и расходных материалов. Результатом работы являются собранные ячейки, которые поступают на вход следующей работы «Контроль и настройка», где они преобразуются либо в брак, либо в настроенный прибор. Работа осуществляется под управлением ГОСТов, ОСТов и ТЗ при помощи персонала и оборудования. Результат данной работы является - настроенный прибор, который является входом для последней работы «Установка в корпус». Данная работа преобразует входные воздействия, такие как настроенный прибор и комплектующие, в годную продукцию под управлением ГОСТов, ОСТов и ТЗ с помощью персонала, оборудования и расходных материалов.

Заключение

Результатом проведения исследования является разработка автоматизированной информационной системы электронного документооборота предприятия, занимающегося изготовлением типовых радиоэлектронных изделий на примере устройства «Толщиномер покрытий магнитный МТП-01»

Преимуществом использования структурно-функционального анализа является возможность обеспечения прозрачности производственного процесса на каждом этапе жизненного цикла изделия.

Литература

1. Выбор системы электронного документооборота [Электронный ресурс]. Режим доступа: www.fosdoc.ru, свободный. – Проверено 31.10.2012
2. Власов А.И., Иванов А.М. Визуальные модели управления качеством на предприятиях электроники// Наука и образование: электронное научно-техническое издание. 2011. № 11. С. 34-34.
3. Власов А.И., Кирбабин О.Е., Шепель А.С. Методология визуального проектирования как инструмент организации поддержки учебного процесса // Новые информационные технологии и менеджмент качества (NIT&MQ`2008). Материалы международного симпозиума/Редкол.: А.Н.Тихонов (пред.) и др.; ФГУ ГНИИ ИТТ "Информика". - М.:ЭГРИ, 2008.-219 с.: ил. С.110-112.
4. Паспорт прибора «Толщиномер покрытий магнитный МТП-01». – М.: МНПО НИИИИИИ Спектр, 2012, 25с.
5. ГОСТ Р 50.1.028-2001 Информационные технологии поддержки жизненного цикла продукции «методология функционального моделирования». – М.: ИПК Издательство стандартов, 2001, 78 с.

