

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО
ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
Факультет компьютерных наук

**СБОРНИК СТУДЕНЧЕСКИХ
НАУЧНЫХ РАБОТ
ФАКУЛЬТЕТА КОМПЬЮТЕРНЫХ
НАУК ВГУ**

Выпуск 9

Под редакцией кандидата технических наук Д.Н. Борисова

Воронеж
Издательский дом ВГУ
2015

УДК 004.65 + 004.438.5

ББК Ч481(2)22

С23

Рекомендовано к печати

Ученым советом факультета компьютерных наук ВГУ

С23 **Сборник студенческих научных работ факультета компьютерных наук ВГУ** / под ред. Д. Н. Борисова ; Воронежский государственный университет. – Выпуск 9. – Воронеж : Издательский дом Воронежского государственного университета, 2015. – 307 с.
ISBN 978-5-9273-2221-3

В сборник включены научные работы студентов факультета компьютерных наук ВГУ, выполненные в 2014-2015 гг. под руководством преподавателей факультета, представленные в виде докладов и рекомендованные к опубликованию оргкомитетами студенческих научных конференций.

УДК 004.65 + 004.438.5

ББК Ч 481 (2) 22

ISBN 978-5-9273-2221-3

© Воронежский государственный университет, 2015
© Оформление. Издательский дом ВГУ, 2015

КОМПЬЮТЕРНЫЕ АЛГОРИТМЫ ИНТЕГРИРОВАНИЯ МАТРИЧНЫХ АЛГЕБР ЛИ

Е. В. Акимова

Студент

А. В. Лобода

Профессор

Введение и постановка задачи

Целью данной статьи является разработка алгоритма решения некоторых специальных систем дифференциальных уравнений в частных производных. Эти системы связаны с задачей описания аффинно-однородных поверхностей в многомерных пространствах (см. [1]-[4]).

Одна из систем такого типа в упрощенной форме выглядит так:

$$\frac{\partial \Phi}{\partial x_1} + (2 \cdot x_2) \frac{\partial \Phi}{\partial u} + (2 \cdot y_2) \frac{\partial \Phi}{\partial v} = 0,$$

$$\frac{\partial \Phi}{\partial y_1} + (2 \cdot y_2) \frac{\partial \Phi}{\partial u} - (2 \cdot x_2) \frac{\partial \Phi}{\partial v} = 0,$$

$$-\mu \cdot x_2 \frac{\partial \Phi}{\partial x_1} - \mu \cdot y_2 \frac{\partial \Phi}{\partial y_1} + \frac{\partial \Phi}{\partial x_2} - 2(x_1 - y_2) \frac{\partial \Phi}{\partial u} + 2(-x_2 - y_1) \frac{\partial \Phi}{\partial v} = 0,$$

$$\beta \cdot x_2 \frac{\partial \Phi}{\partial x_1} + \beta \cdot y_2 \frac{\partial \Phi}{\partial y_1} + \frac{\partial \Phi}{\partial y_2} + 2(3 \cdot x_2 - y_1) \frac{\partial \Phi}{\partial u} + 2(x_1 + 3 \cdot y_2) \frac{\partial \Phi}{\partial v} = 0, \quad \frac{\partial \Phi}{\partial u} = 0$$

Решением этой системы является функция

$$\Phi = -v + F(x_1, x_2, y_1, y_2, u),$$

определяющая некоторую аффинно-однородную поверхность в трехмерном комплексном пространстве, где

$$x_1 = \operatorname{Re}(z_1), y_1 = \operatorname{Im}(z_1), x_2 = \operatorname{Re}(z_2), y_2 = \operatorname{Im}(z_2), u = \operatorname{Re}(w), v = \operatorname{Im}(w).$$

Сама такая система соответствует описанию алгебры Ли касательных векторных полей к искомой (изначально неизвестной) однородной поверхности.

Алгебры Ли появляются из рассмотрения различных случаев аффинной однородности (см. [1]-[3]) и рассматриваются как матричные алгебры Ли.

1. Примеры алгебр Ли

Ниже представлены своими базисами несколько матричных алгебр Ли, полученных в связи с задачами об однородности.

Пример 1 (Атанов).

$$E_1 = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, E_2 = \begin{pmatrix} 0 & 0 & 0 & i \\ 0 & 0 & 0 & 0 \\ 0 & -2i & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix},$$

$$E_3 = \begin{pmatrix} 0 & -\mu & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -2 & -2i & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, E_4 = \begin{pmatrix} 0 & \beta & 0 & 0 \\ 0 & 0 & 0 & i \\ 2i & 6 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix},$$

$$E_5 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

Пример 2 (Атанов).

$$E_1 = \begin{pmatrix} -i & 0 & 0 & 1 \\ 0 & -i & 0 & 0 \\ 6i & 4i & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, E_2 = \begin{pmatrix} 3 & 2 & 0 & i \\ -2 & -1 & 0 & 0 \\ -2 & -4 & 2 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix},$$

$$E_3 = \begin{pmatrix} -i & 0 & 0 & 0 \\ 0 & -i & 0 & 1 \\ 4i & 2i & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, E_4 = \begin{pmatrix} 3 & 2 & 0 & 0 \\ -2 & -1 & 0 & i \\ -4 & -6 & 2 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix},$$

$$E_5 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}, E_6 = \begin{pmatrix} i & i & 0 & 0 \\ -i & -i & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

Пример 3 (Павлова). $A, m, \varepsilon \in \mathfrak{R}, 0 < \varepsilon \neq \frac{1}{2}$

$$E_1 = \begin{pmatrix} A & 0 & 0 & 1 \\ 0 & m & 0 & 0 \\ 4i & 0 & 2m & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, E_2 = \begin{pmatrix} 0 & 0 & 0 & i \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, E_3 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$E_4 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & i \\ 0 & 2(1-2\varepsilon) & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, E_5 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 2i(1+2\varepsilon) & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Напомним, что в матричной алгебре Ли скобка (коммутатор) двух произвольных элементов этой алгебры

$$[M_i, M_k] = M_i \cdot M_k - M_k \cdot M_i$$

также является элементом алгебры, т. е. разлагается с некоторыми вещественными коэффициентами по ее базису.

2. Интегрирование

Интегрирование подобных алгебр, т.е. нахождение их интегральных поверхностей, можно реализовать посредством длительной трудоемкой процедуры. Одним из ее этапов является построение всех решений системы уравнений в частных производных типа приведенной во введении.

С помощью пакета символьной математики Maple был разработан алгоритм, способный выполнить все этапы процедуры решения нашей задачи. По заданному базису алгебры этот алгоритм выдает уравнение искомой однородной поверхности.

Алгоритм использует операции с комплексными выражениями, включающие выделение из них вещественных и мнимых частей, выделение и упрощение систем дифференциальных уравнений, дифференцирование выражений. Ключевым его фрагментом является Maple-команда `pdsolve` ([5, с.76]), решающая уравнение в частных производных. Обычно это делается через решение системы обыкновенных дифференциальных уравнений, что является самым трудоемким действием при «ручном режиме» исследования исходной задачи (см., например, [1]).

Краткое описание отдельных шагов алгоритма:

1. Требуется представить базисные элементы матричной алгебры в виде аффинных векторных полей.

В примере 1 векторные поля выглядят так:

$$E_1 = 1 \frac{\partial}{\partial z_1} + (2 \cdot z_2) \frac{\partial}{\partial w},$$

$$E_2 = i \frac{\partial}{\partial z_1} - (2i \cdot z_2) \frac{\partial}{\partial w},$$

$$E_3 = (-\mu \cdot z_2) \frac{\partial}{\partial z_1} + 1 \frac{\partial}{\partial z_2} - (2 \cdot z_1 + 2i \cdot z_2) \frac{\partial}{\partial w},$$

$$E_4 = (\beta \cdot z_2) \frac{\partial}{\partial z_1} + i \frac{\partial}{\partial z_2} + (2i \cdot z_1 + 6 \cdot z_2) \frac{\partial}{\partial w},$$

$$E_5 = 1 \frac{\partial}{\partial w}.$$

2. Переводим базисные векторные поля алгебры в вещественные координаты и переходим к системе дифференциальных уравнений в частных производных.

Система ДУ для примера 1 приведена во введении, а ее упрощенный вариант имеет вид:

$$\frac{\partial F}{\partial x_1} = 2 \cdot y_2, \quad \frac{\partial F}{\partial y_1} = -2 \cdot x_2,$$

$$-\mu \cdot x_2 \frac{\partial F}{\partial x_1} - \mu \cdot y_2 \frac{\partial F}{\partial y_1} + \frac{\partial F}{\partial x_2} = 2 \cdot (x_2 + y_1),$$

$$\beta \cdot x_2 \frac{\partial F}{\partial x_1} + \beta \cdot y_2 \frac{\partial F}{\partial y_1} + \frac{\partial F}{\partial y_2} = 2 \cdot (x_1 + 3 \cdot y_2).$$

3. Переходим к пошаговому решению отдельных уравнений системы, полученной на предыдущем этапе.

Применение алгоритма к трем описанным выше примерам алгебр Ли приводит к следующим аффинно-однородным поверхностям (интегральные поверхности соответствующих алгебр Ли):

Пример 1.

$$v = -x_2^2 + 3 \cdot y_2^2 + 2 \cdot x_1 \cdot y_2 - 2 \cdot x_2 \cdot y_1.$$

Замечание. Параметры μ и β сокращаются в процессе интегрирования.

Пример 2.

$$v = x_2^2 + y_2^2 - 2 \cdot (x_2^2 + y_2^2 + 2 \cdot x_2 + 1) \cdot \ln(x_2^2 + y_2^2 + 2 \cdot x_2 + 1) + 2 \cdot x_1 \cdot y_2 - 2 \cdot x_2 \cdot y_1 - 4 \cdot x_2 - 4.$$

Пример 3.

Уравнение поверхности, полученное применением Maple-алгоритма:

$$v = x_2^2 \cdot (1 + 2\varepsilon) + y_2^2 \cdot (1 - 2\varepsilon) + \frac{4 \cdot x_1}{(A - 2m)} + C(x_1 \cdot A + 1) \frac{2m}{A}.$$

Замечание. Для сравнения отметим, что при аналитическом интегрировании семейства алгебр из примера 3, зависящего от трех вещественных параметров A , m , ε , получаются 5 различных типов поверхностей:

1. $m=0, A=0$

$$v = x_2^2 \cdot (1 + 2\varepsilon) + y_2^2 \cdot (1 - 2\varepsilon) + 2 \cdot x_1;$$

2. $m=0, A \neq 0$

$$v = x_2^2 \cdot (1 + 2\varepsilon) + y_2^2 \cdot (1 - 2\varepsilon) + \frac{4}{A} \cdot x_1 - \frac{4}{A^2} \cdot \ln(x_1 + \frac{1}{A});$$

3. $m \neq 0, A=0$

$$v = x_2^2 \cdot (1 + 2\varepsilon) + y_2^2 \cdot (1 - 2\varepsilon) - \frac{1}{m^2} - \frac{2}{m} \cdot x_1 + C \cdot e^{2mx_1};$$

4. $m \neq 0, A \neq 0$

4.1 $\frac{2m}{A} = 1$

$$v = x_2^2 \cdot (1 + 2\varepsilon) + y_2^2 \cdot (1 - 2\varepsilon) + \frac{4}{A} \cdot (x_1 + \frac{1}{A}) \cdot \ln(x_1 + \frac{1}{A}) + C \cdot (x_1 + \frac{1}{A});$$

4.2 $\frac{2m}{A} \neq 1$

$$v = x_2^2 \cdot (1 + 2\varepsilon) + y_2^2 \cdot (1 - 2\varepsilon) + \frac{4 \cdot x_1}{(A - 2m)} + C(x_1 \cdot A + 1) \frac{2m}{A}.$$

Заключение

Все три примера, описанные выше, были проинтегрированы аналитически и с помощью программы, написанной в пакете символьной математики Maple.

Сравнение интегральных поверхностей, полученных алгоритмом и «в ручном режиме», показывает, что часто результаты совпадают. Однако на примере 3 мы видим, что иногда необходимо дополнительное рассмотрение отдельных частных случаев (связанных с критическими значениями параметров).

При этом алгоритм, написанный в Maple, выполняет большую часть описанной процедуры, существенно ускоряя решение задачи и освобождая его от естественных ошибок «ручного режима». Отметим, впрочем, что на некоторых примерах алгебр алгоритм оказывается работоспособным лишь после предварительного упрощения базисов этих алгебр (см. также [6]).

Список литературы

1. Лобода, А. В. Об одном семействе аффинно-однородных вещественных гиперповерхностей 3-мерного комплексного пространства / А. В. Лобода, А. С. Ходарев // Известия ВУЗов. Математика. – 2003. – № 10. – С. 38–50.
2. Лобода, А. В. О различных способах представления матричных алгебр Ли, связанных с однородными поверхностями / А. В. Лобода, В. К. Евченко // Изв. вузов. Матем. – 2013. – № 4. – 42–60.
3. Лобода, А. В. О полном списке аффинно-однородных поверхностей $(\epsilon, 0)$ – типов в пространстве S^3 / А. В. Лобода, А. В. Шиповская // Изв. вузов. Матем.. – 2015. – № 6. – 75–81.
4. Атанов, А. В. Аффинно-однородные поверхности типа $(0, 0)$ в пространстве S^3 / А. В. Атанов, А. В. Лобода // Матем. заметки. – 2015. – №2. – С. 309–313.
5. Савотченко, С. Е. Методы решения математических задач в Maple: Учебное пособие / С. Е. Савотченко, Т. Г. Кузьмичева – Белгород : Изд. Белаудит, 2001. – 115 с.
6. Кочурова, Е. Д. Интегрирование матричных алгебр Ли / Е. Д. Кочурова, А. В. Лобода // Сборник студенческих научных работ ФКН ВГУ. – Вып. 7. – Воронеж, 2013. – С. 68–76.

РАЗРАБОТКА КОМПЬЮТЕРНОЙ СИСТЕМЫ МОНИТОРИНГА ДВИЖЕНИЯ ГЛАЗ

А. В. Алексеев

Студент

Я. А. Туровский

Доцент

Введение

Анализ движения глаз на данный момент является крайне важной задачей в таких направлениях как медицина, маркетинговые исследования, исследования usability. Крайне перспективной задачей является управления различными устройствами при помощи глаз – это очень полезно в современном мире, где все больше в обычную жизнь входят электронные системы (например: системы типа “умный дом”) и так же это поможет людям с ограниченными физическими возможностями [1].

За последние 20 лет появилось большое количество различных систем и способов анализа движения глаз, но большинство из них осталось не

реализованными для достаточно широкого применения даже в рамках лабораторных исследований. Самыми распространенными сейчас являются системы, которые используют метод электроокулографии и видеоокулографии. Данные методы имеют свои достоинства и недостатки.

Вместе с окулографическими в последнее время большую актуальность имеют системы, которые используют нейрокомпьютерные интерфейсы. Методов, по которым работают нейрокомпьютерные интерфейсы достаточно много, однако, в силу определенных особенностей, они имеют ряд недостатков, связанных с низкой, по сравнению с другими типами интерфейсов, скоростью и точностью работы.

Для того что бы повысить устойчивость и качество работы нейрокомпьютерного и окулографического интерфейса часто применяется решение об их объединении. Это позволяет намного увеличить поток данных, получаемых от пользователя на компьютер и устранить наиболее частые проблемы при использовании данных интерфейсов по отдельности: ошибочные данные и неустойчивую работу. Так же это позволяет использовать такую систему при различных внешних условиях, что приводит к ее еще более устойчивой работе. Использование данной системы позволяет достигнуть создания беспроводной системы по управлению различными внешними устройствами и системы “виртуального мира” в полном объеме.

После проведения анализа рынка окулографических систем и устройств, стало понятно, что данные устройства в странах СНГ и России единичны, несмотря на то, что первые окулографические системы появились именно в СССР [2]. В мире так же немного фирм, которые занимались данной тематикой. Исходя из этого, было принято решение о создании своей модульной окулографической системы, которая могла бы работать одновременно с нейрокомпьютерными интерфейсами.

1. Анализ аналогов

Для анализа подобных систем были выбраны лидирующие на рынке системы фирм *SensoMotoric Instruments (SMI)* и *Tobii*.

Компания *SMI* была основана Доктором Винфридом Тайвайсом в 1991 году и являлась первым производителем коммерческих систем трекинга глаз на основе видеоокулографии. Первый продукт *SMI*, *3D VOG*, использовался на бортах Европейского космического агентства, НАСА и орбитального комплекса «Мир» для анализа воздействия космических миссий на вращательные движения глаз космонавта, вызванных гравитацией. Со временем компания сместила акцент с космонавтики в сторону офтальмологии и научных исследований.

Шведская фирма Tobii производит устройства видеоокулографии для регистрации движений глаз (eye tracking), которые встроены в различные другие устройства для реабилитации, например, специальные планшеты для обучения детей с синдромом Дауна. Она была образована уже после SMI, но, несмотря на отставание по времени, стала достаточно быстро выпускать свою продукцию, заняв существенную долю рынка.

Устройства данных фирм весьма дороги, так как используют комплектующие, которые производятся на заказ. Они достаточно сложны в управлении и требуют специального обучения, что делает их использование затруднительным для потребителей. Среди минусов надо отметить невозможность использования данных систем совместно с другими системами без специального программного обеспечения, которое поставляются в составе более дорогих комплектаций комплекса.

2. Цель работы

Исходя из анализа других систем и устройств, было принято решение о создании своей модульной системы мониторинга движения глаз. Для достижения цели работы должны быть решены следующие задачи:

- 1) создание новых методов обработки изображений, отражающих особенности данной системы, и соответствующих алгоритмов;
- 2) разработка достаточно дешевого и простого устройства для обнаружения зрачка;
- 3) создание компьютерной системы, позволяющей определить направление взгляда, на основе данных, получаемых от видеокамер

Исходя из указанных задач, был проведен ряд исследований, которые позволили определить оптимальные характеристики системы, а также направления создания алгоритмов обработки изображений и обнаружения на нём зрачка.

3. Реализация

Для решения первой задачи было выбрано направление создания алгоритма, который использует фильтрацию изображения и, учитывая освещенность участка глаза, выбирает метод детектирования зрачка: «яркий» или «темный». Такой алгоритм был реализован, а также дополнен системой подавления шума, который появляется из-за различных факторов (например, из-за вспышки света) и может помешать правильной обработке изображения глаза [3]. Результат работы системы подавления шума показан на рис. 1 и рис. 2. Также алгоритм регулирует степень чувствительности ИК – камеры к свету в зависимости от освещенности изображения.



Рис. 1. Изображение до фильтрации

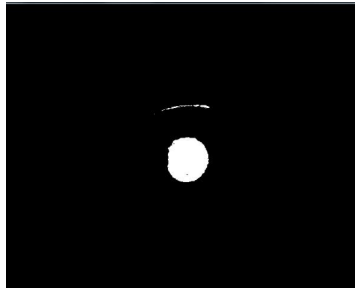


Рис. 2. Изображение после фильтрации

Для выполнения второй задачи была использована обычная видеокамера с извлеченным ИК фильтром, что позволило получить более четкое изображение области глаза. Это заметно упростило поиск зрачка на изображении. Для данной камеры был изготовлен специальный держатель, который крепится на голове и обеспечивает постоянный угол между оптическими осями камеры и глаза. В ряде случаев, для улучшения освещенности области глаза был использован вариант установки маломощного ИК диода для подсветки [4]. Держатель предоставляет возможность изменения положения камеры, что позволяет настроить изображение с камеры вне зависимости от формы головы, а также расположить камеру так, чтобы она не закрывала обзор человеку.

Для решения третьей задачи был создан программно-аппаратный комплекс, который состоит из одной ИК камеры, двух обычных видеокамер и программного обеспечения, осуществляющего обработку информации. Для определения направления взгляда были использованы две видеокамеры, которые фиксируют изображения объектов, расположенных перед пользователем, и составляют панорамную картину этих объектов. Эти камеры закреплены на держателе сверху и не мешают обзору человека. Комплекс работает следующим образом: после получения изображения зрачка и панорамного изображения объектов, которые

находятся перед человеком, программа определяет координаты положения зрачка и переносит их на панорамное изображение. При этом учитывается область видимости глаза и перед запуском обработки данных происходит калибровка всех камер. В качестве дополнения в данную систему был добавлен нейрокомпьютерный интерфейс, использующий частотные компоненты электроэнцефалограммы (ЭЭГ) в качестве дополнительных данных для управления внешними устройствами [5]. Это позволяет более гибко использовать данную систему, “разгрузив” интерфейс мозг-компьютер и передав, тем самым, ряд важных её функций окулографическому интерфейсу. Для использования данного нейрокомпьютерного интерфейса необходима обработка большого количества данных в виде ЭЭГ. Для быстрой обработки этих данных была использована технология CUDA [6].

4. Использование системы

Основными потребителями данной системы являются предприятия и учреждения, специализирующиеся на реабилитации больных. Она может быть использована и для проведения маркетинговых исследований. Предлагаемая система, в отличие от аналогов, крайне проста в производстве и управлении. Надо отметить её низкую себестоимость и возможность простого включения в более масштабные проекты. Стабильно развивающимся является рынок AR-устройств, в которые может быть интегрирована разрабатываемая система. AR – это среда с прямым или косвенным дополнением физического мира цифровыми данными в режиме реального времени при помощи компьютерных устройств — планшетов, смартфонов и инновационных гаджетов вроде Google Glass или Vuzix Wrap 1200AR, а также программного обеспечения. С учётом вышеизложенного, на данный момент у предлагаемой системы нет конкурентов в категории систем мониторинга движения глаз.

Заключение

При реализации данного проекта были выполнены все поставленные задачи. Для проведения тестирования окулографического интерфейса глаз-мозг-компьютер была создана фокус группа из 32 человек, состоящая из 28 мужчин и 4 женщин. Перед ними была поставлена задача проезда по замкнутому маршруту, управляя самодвижущимся шасси в виде модели танка. Управление осуществлялось при помощи окулографического интерфейса. Единственным источником информации, которым располагали пользователи, было изображение, получаемое с камеры, закрепленной на модели танка. Каждому из группы было предоставлено три попытки, которые длились 3 минуты. За это время необходимо было объехать два препятствия и вернуться на исходную позицию. Важно

отметить, что все участники тестирования, до момента начала теста, не управляли какими – либо устройствами при помощи подобных интерфейсов, а задание и расположение препятствий держались в секрете. После каждого проезда, данные с участников собирались и подвергались анализу.

По результатам проведения такого тестирования 28 человек из 32 успешно справились с такой задачей.

Исходя из данных полученных в ходе заездов, были выявлены сильные и слабые стороны информационной системы для управления внешними устройствами и намечены пути её совершенствования нейрокомпьютерного интерфейса глаз-мозг-компьютер.

Список литературы

1. Wedel, M. Eye fixations on advertisements and memory for brands: a model and findings / M. Wedel, R. Pieters // *Marketing Science*. – 2000. – № 19. – С. 297–312.

2. Duchowski, A.T. A Breadth-First Survey of Eye Tracking Applications/ A.T. Duchowski // *Behavior Research Methods, Instruments, & Computers*. – 2002. – №34. – С. 455–470.

3. Posner, M. I. Orienting of attention / M. I. Posner // *Quarterly Journal of Experimental Psychology*. – 1980. – № 32. – С. 3-25.

4. Bulling, A. Wearable EOG goggles: Seamless sensing and context-awareness in everyday environments / A.Bulling, D.Roggen, G.Tröster // *Journal of Ambient Intelligence and Smart Environments*. – 2009. – №1. – С. 157-171.

5. Туровский, Я. А. Моделирование нейрокомпьютерного интерфейса на основе активной парадигмы / Я. А. Туровский, С. Д. Кургалин, А. В. Максимов // *Системы управления и информационные технологии*. – 2012. – Т.47. – № 1. – С. 99-103.

6. Вахтин, А. А. Реализация численного вейвлетного преобразования на графических адаптерах архитектуры NVIDIA CUDA / А. А. Вахтин, Я. А. Туровский // *Вестник Воронежского государственного университета. Сер. Системный анализ и информационные технологии*. – 2012. – №1. – С. 69-72.

ОБУЧАЮЩЕЕ ПРИЛОЖЕНИЕ «ПОШАГОВОЕ ВЫПОЛНЕНИЕ SQL-ЗАПРОСОВ»

С. С. Батищева

Студент

Д. И. Соломатин

Старший преподаватель

Введение

На сегодняшний день реляционные базы данных (БД) применяются во многих компьютерных системах и пользовательских приложениях, что побуждает программистов к изучению языка запросов SQL. В университетах по всему миру в обязательную программу включен курс изучения SQL, однако у многих студентов поначалу возникают трудности с его освоением.

В данной работе рассматривается создание программного средства, которое упростило бы понимание основ работы Select-операторов для студентов и дало возможность на примерах понять логику их выполнения.

1. Постановка задачи

В рамках описываемой работы были сформулированы следующие требования к разрабатываемому программному продукту:

- ввод пользователем запроса;
- вывод результатов выполнения следующих блоков запроса: from, join, where, group by, order by, подзапросы, групповые, строковые и числовые функции;
- добавление таблиц БД с возможностью их использования;
- отображение результатов пошагового выполнения запроса в файле HTML. Использование HTML обусловлено наглядностью представления информации в файлах данного типа.

В качестве метода получения промежуточных данных были рассмотрены два варианта: обращение к реальной СУБД и написание собственного интерпретатора Select-запросов языка SQL, удовлетворяющего всем поставленным ранее требованиям. В современных СУБД осуществляется оптимизация каждого отдельно взятого оператора языка, в результате чего первоначальный запрос может быть коренным образом изменён в целях ускорения выдачи результата пользователю. Кроме того, СУБД не предоставляют сторонним разработчикам возможности получения промежуточных данных во время выполнения

запроса. Исходя из минусов первой альтернативы, возможность её реализации была исключена из последующего рассмотрения.

2. Правила выполнения SQL-запроса

Существуют логические правила выполнения запросов, вытекающие из семантики языка SQL (см. таблицу). В процессе обучения студентов основам SQL именно эти правила разбираются в литературе, на обучающих сайтах и на лекциях в вузах. Однако физическое исполнение запросов в реальных СУБД сильно отличается от данных логических правил из-за оптимизации. Но людям для успешного написания запросов не требуется понимать внутреннюю работу СУБД, т. к. для пользователей СУБД работает по принципу чёрного ящика: на вход СУБД поступает запрос, на выходе пользователь получает результирующую таблицу. Цель разрабатываемого приложения – показать, как происходит выполнение запроса на логическом уровне по шагам, т. е. так, как человек должен представлять себе последовательность выполнения запроса при его написании.

Таблица

Правила выполнения оператора select SQL-запроса

| Очередность | Название блока | Пояснение |
|-------------|----------------|--|
| 1 | From | Выполняется выборка указанных в блоке таблиц и формируется рабочая таблица |
| 2 | Join | Указанные таблицы из блока From объединяются одним из возможных способов в зависимости от вида Join |
| 3 | Where | Из рабочей таблицы выбираются строки, для которых предикаты данного блока являются истинными |
| 4 | Group by | Строки группируются по условиям или столбцам, указанным в этом блоке |
| 5 | Select | Из рабочей таблицы выбираются нужные столбцы и, если указаны какие-либо функции, добавляются столбцы с результатами выполнения функций |
| 6 | Order by | Результирующая таблица сортируется по указанным столбцам |

Также в блоках Select, From, Where могут быть указаны подзапросы. В этом случае сначала выполняется подзапрос, подчиняющийся указанным логическим правилам, а затем выполняется внешний запрос.

3. Реализация приложения

Процесс разбора файла, поступающего на вход программы и содержащего SQL-запрос, состоит из двух частей: лексического и синтаксического анализа [1]. Лексический и синтаксический анализаторы описываются с помощью формальных грамматик. Грамматика состоит из правил преобразования последовательностей символов в другие последовательности [2].

Основным языком разработки является C# и среда Visual Studio 2013.

В качестве генератора нисходящих анализаторов для формальных языков был выбран ANTLR. Работа с ANTLR начинается с построения файла грамматики Sql3.g. Исходя из файла грамматики, ANTLR генерирует три файла [1]:

- Sql3Lexer.cs – класс лексического анализатора, разбивающий поток символов из входного файла Input.txt на поток лексем;
- Sql3Parser.cs – класс синтаксического анализатора, обрабатывающий поток токенов и строящий AST-дерево;
- файл Sql3.tokens – файл, содержащий все типы токенов.

Полученные файлы вместе с остальными классами приложения компилируются в исполняемую программу Sql3.exe.

Основным классом разработанной программы является Program – класс, в котором происходит работа с лексером и парсером, вызов интерпретатора, заполнение списка промежуточных таблиц результатами выполнения текущего запроса и отправка полученных данных на печать.

Реализованная программа состоит из следующих частей (рис. 1):

- часть, отвечающая за выполнение запросов – пакет Interpretation. Созданные ANTLR классы Sql3Lexer и Sql3Parser отвечают за генерацию AST-дерева. Класс ColumnDescr описывает столбцы в таблицах базы данных. Класс Table служит для отображения таблиц базы данных и состоит из списка столбцов и списка строк. Класс Interpreter содержит логику выполнения SQL-запросов. Выполнение запросов производится при последовательной обработке AST-дерева.

- часть, имитирующая базу данных программы – пакет DataBase. Эта часть представлена классом DemoDB. Он отвечает за загрузку .txt-файлов из специальной директории в качестве таблиц базы данных и предоставляет интерпретатору экземпляры класса Table.

- часть, осуществляющая вывод данных в HTML. Класс HTMLData отвечает за представление данных в виде, удобном для последующей обработки в классе HTMLPrinter. HTMLPrinter осуществляет отображение экземпляров HTMLData в файл HTML.

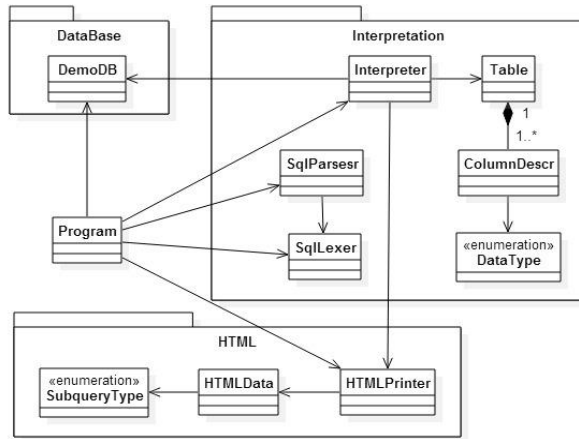


Рис. 1. Диаграмма классов программы

4. Представление пошагового выполнения запроса в HTML

Строение генерируемого программой HTML-файла имеет следующий вид. Вверху сайта выводится написанный пользователем запрос (рис. 2).

Выполнение SQL-запроса

```

select S.id as Student_id,
       concat(S.name, ' ', S.surname) as Student_name,
       U.name as University_name
from Students as S
left join (
  select *
  from Universities
  where lower(name) like '%воронеж%') as U
on S.univ_id=U.id
where S.entry_year=2014
order by U.name, S.name

```

Рис. 2. Заголовок сайта

Далее в соответствии с логическими правилами Select-запроса выводятся промежуточные таблицы (рис. 3, 4, 5).

В первом блоке выводятся таблицы из базы данных. Одна из них является результатом выполнения подзапроса, пошаговый просмотр которого возможен, если нажать на кнопку «Показать» (рис. 3). Свернуть подзапрос можно, нажав на кнопку «Скрыть».

1. Блок from

Выбираются указанные в блоке from таблицы из базы данных. Кроме того, при необходимости таблицам присваиваются псевдонимы.

| # | S.id | S.univ_id | S.name | S.surname | S.entry_year | S.birthday |
|---|------|-----------|--------|-----------|--------------|--------------------|
| 1 | 3 | 71 | Петя | Абдулов | 2015 | 05.08.1995 0:00:00 |
| 2 | 2 | 36 | Ира | Парфенова | 2014 | 10.12.1993 0:00:00 |
| 3 | 6 | 23 | Вася | Федин | 2013 | 23.03.1994 0:00:00 |
| 4 | 10 | 67 | Катя | Петрова | 2014 | 14.11.1995 0:00:00 |
| 5 | 8 | 95 | Аня | Иванова | 2014 | 30.07.1992 0:00:00 |
| 6 | 5 | 36 | Аня | Кошкина | 2010 | 14.11.1992 0:00:00 |
| 7 | 7 | 82 | Ира | Якова | 2014 | 10.02.1995 0:00:00 |

| # | U.id | U.name | U.year | U.director |
|---|------|---|--------|------------|
| 1 | 23 | Воронежский политехнический институт | 1948 | Петров |
| 2 | 71 | Воронежский государственный архитектурно-строительный университет | 1931 | Иванов |
| 3 | 82 | Воронежский государственный университет | 1924 | Ендовицкий |
| 4 | 36 | Воронежская государственная медицинская академия | 1901 | Хомов |

Просмотр пошагового выполнения подзапроса Показать Скрыть

Рис. 3. Блок From

В блоке join объединяются таблицы с помощью Left outer join по заданному условию. В блоке where отбрасываются строки, не удовлетворяющие условию (рис. 4).

2. Объединение таблиц из блока from с помощью left (outer) join `on S.univ_id=U.id`

Таблицы объединяются в одну сводную таблицу путём присоединения строк одной таблицы к строкам другой на основе заданного в блоке on условия с отбрасыванием строк из правой (т.е. второй) таблицы, для которых нет соответствия между столбцами левой таблицы и столбцами правой. Полученная таблица является результирующей для блока from.

| # | S.id | S.univ_id | S.name | S.surname | S.entry_year | S.birthday | U.id | U.name | U.year | U.director |
|---|------|-----------|--------|-----------|--------------|--------------------|------|---|--------|------------|
| 1 | 3 | 71 | Петя | Абдулов | 2015 | 05.08.1995 0:00:00 | 71 | Воронежский государственный архитектурно-строительный университет | 1931 | Иванов |
| 2 | 2 | 36 | Ира | Парфенова | 2014 | 10.12.1993 0:00:00 | 36 | Воронежская государственная медицинская академия | 1901 | Хомов |

3. Блок where `S.entry_year=2014`

Из таблицы, полученной в блоке from, выбираются записи, отвечающие условию поиска, а все остальные строки отбрасываются. В таблице зелёным цветом отмечены строки, вошедшие в результирующий набор, а серым цветом отмечены строки, не вошедшие по какому-либо условию поиска.

| # | S.id | S.univ_id | S.name | S.surname | S.entry_year | S.birthday | U.id | U.name | U.year | U.director |
|---|------|-----------|--------|-----------|--------------|--------------------|------|--|--------|------------|
| 1 | 2 | 36 | Ира | Парфенова | 2014 | 10.12.1993 0:00:00 | 36 | Воронежская государственная медицинская академия | 1901 | Хомов |
| 2 | 7 | 82 | Ира | Якова | 2014 | 10.02.1995 0:00:00 | 82 | Воронежский государственный университет | 1924 | Ендовицкий |
| 3 | 11 | 71 | Костя | Пупкин | 2014 | 15.05.1994 0:00:00 | 71 | Воронежский государственный университет | 1931 | Иванов |

Рис. 4. Блоки Left outer join и Where

В четвертом блоке выбираются только указанные в Select столбцы. В пятом блоке строки сортируются по некоторому условию (рис. 5).

4. Блок select

Из результирующей таблицы предыдущего блока выбираются только указанные столбцы, а также добавляются результаты выполнения функций, если таковые указаны в запросе.

| # | Student_id | Student_name | University_name |
|---|------------|---------------|---|
| 1 | 2 | Ира Парфенова | Воронежская государственная медицинская академия |
| 2 | 7 | Ира Якова | Воронежский государственный университет |
| 3 | 11 | Костя Пупкин | Воронежский государственный архитектурно-строительный университет |
| 4 | 10 | Катя Петрова | null |
| 5 | 8 | Аня Иванова | null |

5. Блок order by U.name, S.name

Результирующая таблица блока select сортируется по указанным условиям, столбцам. Если в запросе присутствует group by, на данном шаге из каждой группы останется только первая строка, все остальные будут удалены из результирующей таблицы.

| # | Student_id | Student_name | University_name |
|---|------------|---------------|---|
| 1 | 2 | Ира Парфенова | Воронежская государственная медицинская академия |
| 2 | 11 | Костя Пупкин | Воронежский государственный архитектурно-строительный университет |
| 3 | 7 | Ира Якова | Воронежский государственный университет |
| 4 | 8 | Аня Иванова | null |
| 5 | 10 | Катя Петрова | null |

Рис. 5. Блоки Select и Order by

Заключение

В результате проделанной работы было создано обучающее приложение, отвечающее всем предъявленным требованиям.

На текущей стадии внедрения приложение передано научному руководителю для использования в целях обучения студентов.

У данного программного средства существуют перспективы развития:

- усовершенствование и развитие грамматики до стандарта SQL : 2011. Внедрение предложения Having, добавление нереализованных конструкций и тонкостей существующих блоков;
- дополнение грамматики операторами Update, Insert, Delete и демонстрация правил работы этих операторов.

Список литературы

1. Соломатин, Д. И. Основы синтаксического разбора, построение синтаксических анализаторов / Д. И. Соломатин, А. В. Копытин, А. И. Другалев. – Воронеж : ВГУ, 2014. – 57 с.
2. Молдованова, О. В. Языки программирования и методы трансляции: учебное пособие / О. В. Молдованова. – Новосибирск : СибГУТИ, 2012. – 134 с.

ИССЛЕДОВАНИЕ И ПРИМЕНЕНИЕ МЕТОДОВ ШУМОПОДАВЛЕНИЯ В ЗАДАЧЕ АНАЛИЗА МОРФОЛОГИИ ПОВЕРХНОСТИ ИОНООБМЕННЫХ МЕМБРАН

С. В. Беленов

Студент

Е. А. Сирота

Доцент

Введение

Ввиду того, что зрение – это основной источник получения информации человека, разработка аналогов зрительно-мыслительного комплекса человека является одним из важных направлений развития информационных технологий.

Обработка изображений – существенный компонент в системах машинного зрения, применение которых позволяет автоматизировать многие процессы.

Обработка изображений отвечает за преобразование изображений. Как пример: повышение контраста, резкости, коррекция цветов, сглаживание. Задачей обработки изображения может быть как улучшение (восстановление, реставрация) изображения по какому-то определенному критерию, так и специальное преобразование, кардинально меняющее изображение [1].

1. Теоретический анализ

На химическом факультете ВГУ на кафедре аналитической химии проводятся эксперименты с нанопористыми мембранами. Для оценки вклада зерен на поверхности таких мембран использовались программные средства, такие как photoshop. Это довольно рутинная работа, которая требует больших затрат времени. Для автоматизации анализа морфологии поверхности ионообменных мембран было решено реализовать программный комплекс. Таким образом, целью настоящей работы явилось:

- анализ и исследование методов шумоподавления;
- применение методов фильтраций, основанных на порядковых статистиках;
- реализация пространственных фильтров повышения резкости;
- разработка и реализации новых методов с целью подсчета ионообменных зерен;
- анализ полученных результатов;

Объектами исследования являются гетерогенные катионообменные мембраны МК-40 и МК-41 и анионообменные мембраны МА-40 и МА-41. Морфологические исследования поверхности ионообменных мембран проводятся методом СЭМ. В связи с тем, что изображения ионообменных мембран были получены в результате цифровой обработки, на фотографиях имелись черные и белые пятна, так называемый эффект «соли и перца». На рис. 1 показана микрофотография поверхности химически кондиционированного образца анионообменной мембраны МА-40, полученная при 500-кратном увеличении, а также так называемый эффект «соли и перца», черные и белые точки на изображении, являющиеся шумом.

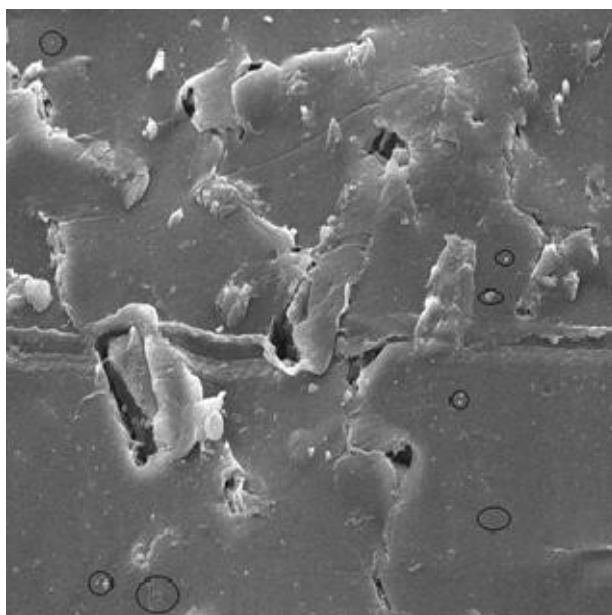


Рис. 1. Эффект «Соли и перца» черные и белые точки

Для того чтобы посчитать площадь мембран необходимо уменьшить шум на полученном изображении. Для это использовались статистические фильтры подавления шума: медианный, минимума, максимума.

На этапе предварительной обработки СЭМ-изображений применяется медианный фильтр с целью шумоподавления [1-3]. Основная функция медианного фильтра заключается в замене отличающегося от фона значения пикселя на другое, более близкое его соседям. Изолированные темные или светлые (по сравнению с окружающим фоном) кластеры,

имеющие площадь не более чем, будут удалены (заменены на значения медиан по окрестностям) медианным фильтром с маской размерами 3x3, 5x5, 7x7 [3, 4].

Использование 0-го перцентиля приводит к фильтру, основанному на выборе минимального значения – фильтру минимума. Такой фильтр полезен при обнаружении наиболее темных точек на изображении – это приводит к уменьшению белого униполярного импульсного шума.

Медианный фильтр, рассмотренный выше является 50-ый перцентиль упорядочного набора чисел. Использование 100-го перцентиля приводит к фильтру, основанному на выборе максимального значения – фильтру максимума.

После уменьшения импульсного шума на изображении требуется определение сегментов ровной поверхности ионообменников. Для этого были использованы три алгоритма сегментаций: пороговый, статистический и смешанный. Для того чтобы воспользоваться любым из трех алгоритмов необходимо получить гистограмму яркости изображения (рис. 2).

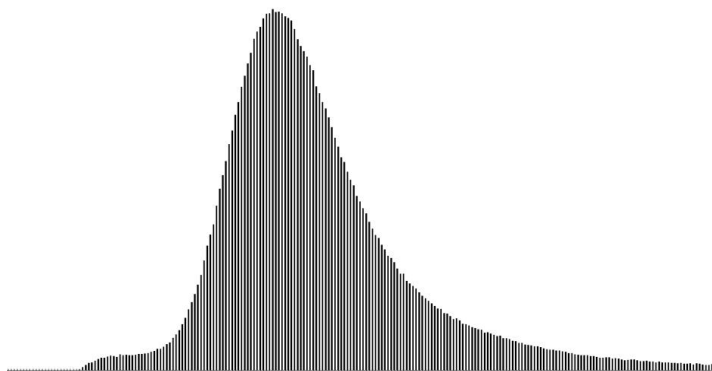


Рис. 2. Гистограмма изображения ионообменной мембраны

В пороговом методе для сравнения используется диапазон значений "яркости" пикселя. Если пиксель попадает в выбранный диапазон, то он присоединяется к группе. С помощью значения порога можно задавать этот диапазон. Чем больше значение порога, тем шире диапазон и тем большего размера вырастет область (при прочих равных условиях). Например, при значении порога 0 диапазон также будет нулевым и в область смогут объединиться только пиксели с одинаковой яркостью. Минус порогового метода в том, что результат очень сильно зависит от начальных условий (в первую очередь от яркости точки, на которую кликнул мышкой оператор).

Статистический метод тоже использует диапазон, но этот диапазон корректируется с учётом статистических характеристик отобранных пикселей. Фактически диапазон здесь постоянно вычисляется заново. Поэтому значение порога, заданное оператором, в этом методе никак не используется. Статистический метод лучше всего работает на предварительно обработанных изображениях. Например, можно выполнить медианную фильтрацию или сглаживание, а потом искать области ионнообменников статистическим методом. Статистический метод гораздо меньше зависит от начальных условий.

Смешанный метод комбинирует оба указанных подхода. Он корректирует диапазон допустимых значений яркости, но при этом учитывает пороговое значение, выбранное оператором.

2. Экспериментальная часть

Для реализации методов шумоподавления, была использована среда программирования Delphi. Она позволяет, создавать подходящие классы, в полную меру, отразить возможности работы с изображением и достаточно быстро выполнять большие объемы вычислений [2].

Для реализации медианного фильтра необходимо определить размер маски яркости изображения. Для вычисления яркости каждого пикселя, программно была реализована формула $Y = 0.3 * R + 0.6 * G + 0.1 * B$, где R,G,B – соответствующая яркость этих цветов в конкретном, дискретном пикселе.

| | | | |
|----|----|----|----|
| 97 | 97 | 97 | 97 |
| 97 | 30 | 97 | 97 |
| 97 | 97 | 97 | 97 |
| 97 | 97 | 97 | 97 |

Рис. 3. Матрица яркостей исходного изображения.

При подсчете яркости получает матрица, по размерности совпадающая с размерностью исходного изображения, но в каждой точке (пиксель), теперь стоит не цвет, а значение яркости. На рис. 3 показан пример получившейся матрицы.

При использовании медианного фильтра был создан массив размерности 3*3, в центре которого была точка (пиксель), яркость которого необходимо проверить. Матрица сортируется, и центральный элемент меняется местами с элементом, который был выделен. Тем самым если у нас есть поле одного цвета, одной интенсивности и в нем есть точка наиболее яркая или наиболее темная, которая будет выделяться из своего окружения, то медианный фильтр ее заменит под цвет, которого больше в

его окружности. Это подходит для уменьшения эффекта «соли и перца». Для дальнейшей работы были реализованы маски размерности 5*5, 7*7. Результат работы масок представлен ниже на рис. 4 и 5 соответственно.

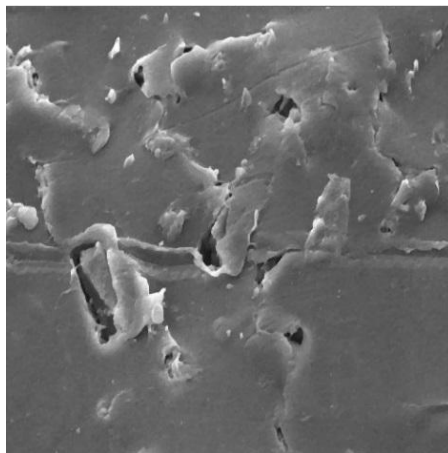


Рис. 4. Результат работы медианного фильтра размерности 3*3

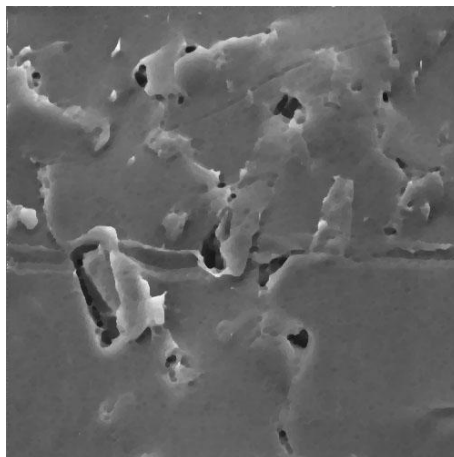


Рис. 5. Результат работы медианного фильтра размерности 5*5

Использование порогового алгоритма позволяет определить площадь ионообменных зерен мембраны. На рис. 6 изображен участок площади, который удалось выделить.

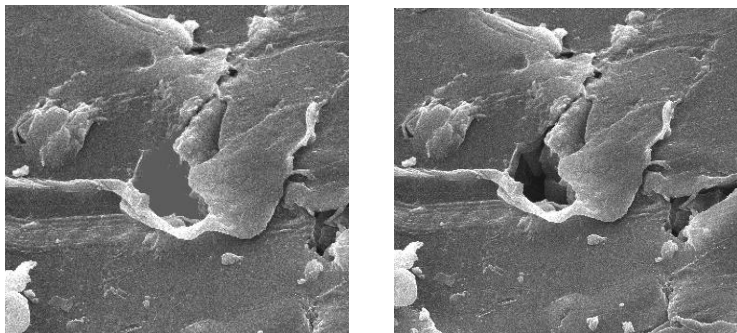
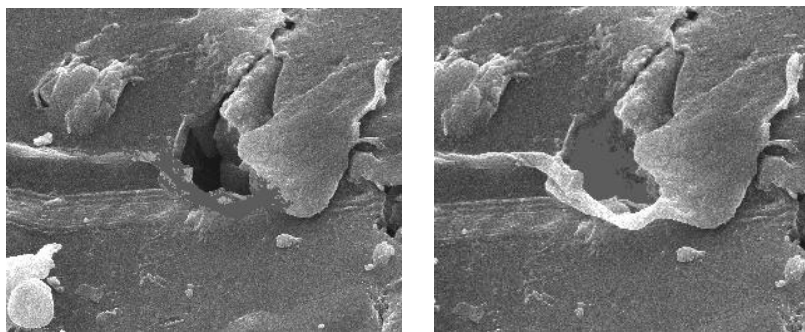


Рис. 6. Справа исходное изображение

Использование статистического алгоритма дает более полное выделение площади ионообменника, как результат будет хорошо посчитана площадь мембраны. К перепадам яркости и контрастности статистический алгоритм более чувствителен, что позволяет получать хороший результат. Смешанный алгоритм позволяет определять белые пространства ионообменных зерен мембраны, там где мембраны имеют «вдутье» или накладываются друг на друга. В комплексе, все методы позволяют с большой точностью определить площадь поверхности ионообменных зерен.



а

б

(а) результат работы статистического алгоритма, (б) результат работы смешанного алгоритма

Рис. 7. Работа алгоритмов

Заключение

Таким образом, в рамках разрабатываемого программного комплекса было реализовано несколько методов шумоподавления. Было установлено, что для дальнейшей работы удобно использовать медианный фильтр размерностью 5x5. Реализована возможность построения гистограммы изображения, для более удобного подсчета площади ионообменника. Реализованы алгоритмы подсчета площади: пороговый, статистический и смешанный. Установлено, что минусом порогового алгоритма является зависимость алгоритма от начальных данных. Статистический алгоритм лучше всего работает на предварительно обработанных изображениях. Смешанный метод совмещает два остальных и должен работать лучше всего. С помощью пространственных фильтров, мы смогли увеличить четкость изображения и получить четкие контуры разрывов. В некоторых случаях появилась возможность вычислить площадь ионообменных мембран. Использование алгоритмов сегментации позволило определить площадь ионообменных зерен мембраны, пригодных для дальнейшей работы.

Список литературы

1. Гонсалес, Р. Цифровая обработка изображений / Р. Гонсалес, Р. Вудс. – Москва : Техносфера, 2005. – 1072 с.
2. Леонтьев, В. Большая энциклопедия компьютера и Интернета / В. Леонтьев. – Москва : ОЛМА Медиа Групп, 2007. – 1084 с.
3. Миронов, А. Д. Обработка цифровых изображений [Электронный ресурс]. – Режим доступа : www.msu.ru/resources/msu-publ.html
4. Грузман, И. Цифровая обработка изображений в информационных системах / И. Грузман, В. Киричук, В. Косых. – Новосибирск : Изд-во НГТУ, 2002. – 352 с.

ПРОГРАММНЫЙ МОДУЛЬ ДЛЯ РАСПОЗНАВАНИЯ СЛАБОКОНТРАСТНЫХ ОБЪЕКТОВ НА ТЕМАТИЧЕСКИХ РАСТРОВЫХ ИЗОБРАЖЕНИЯХ

Б. Бендо

Студент

М. А. Дрюченко

Доцент

Введение

На сегодняшний день задача распознавания образов и извлечения информации из растровых изображений является одной из основных в области компьютерного зрения. В данной статье на примере изображений, содержащих смеси зерновых объектов, рассматривается задача распознавания типов объектов, составляющих смесь. Используется комплекс методов и алгоритмов предварительной обработки растровых изображений, локализации интересующих объектов, выделения их информативных признаков. Для построения распознающей системы на основе обучения с «учителем» по «прецедентам» (при отсутствии статистической модели данных) использовались два базовых подхода – на основе нейронных сетей и метода SVM. Далее рассмотрим каждый из этапов обработки изображений более подробно.

1. Этапы обработки изображений

При обработке растровых изображений можно выделить следующие базовые этапы [1]:

1) предварительная обработка и улучшение качества растрового представления изображения с последующей его бинарной сегментацией и локализацией интересующих объектов на основе морфологических преобразований изображения;

2) определение признаков, описывающих геометрические характеристики формы локализованных объектов, а также статистические характеристики распределения яркости изображений объектов с изъятым сложным фоном;

3) настройка, обучение и тестирование классификаторов многомерных данных, реализованных на основе многослойной нейронной сети (НС) прямого распространения и на основе метода опорных векторов (SVM).

Укрупненная блок-схема, отражающая основные этапы процесса обработки приведена на рис. 1. Рассмотрим каждый из этапов более подробно.

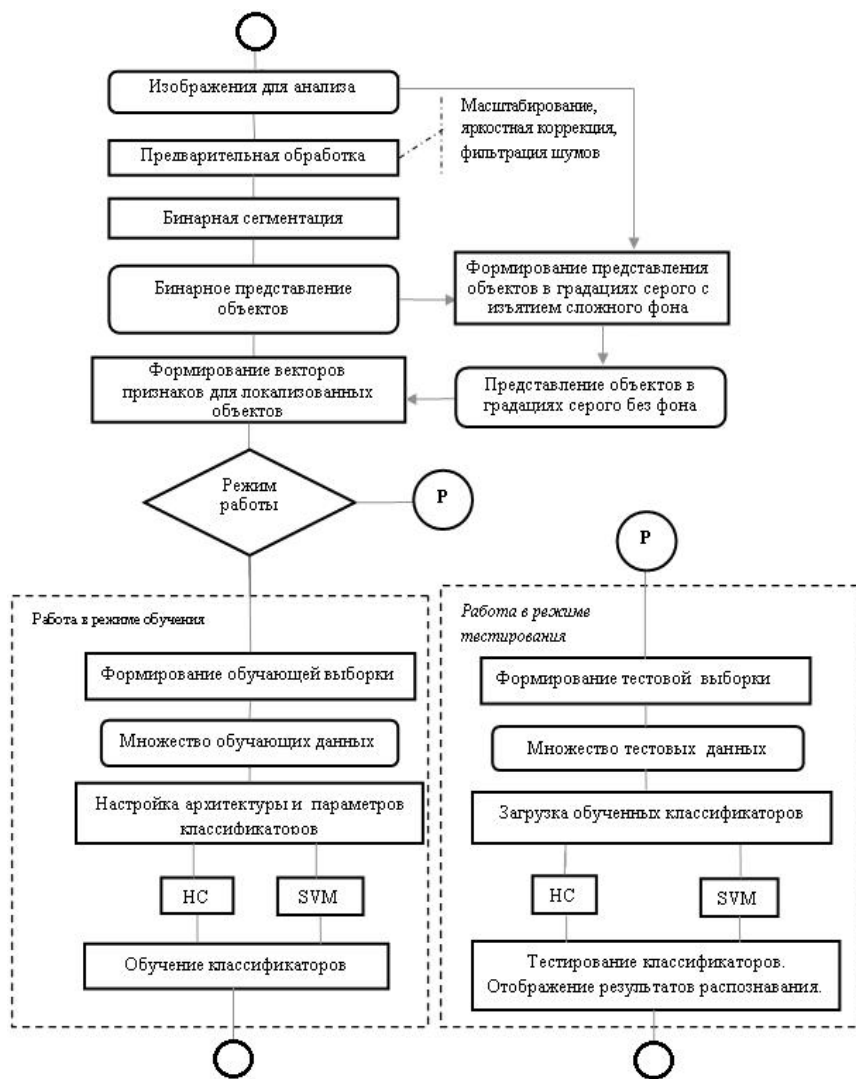


Рис. 1. Укрупненная блок-схема процесса обработки

2. Предварительная обработка и сегментация

С целью улучшения качества, представленных на изображениях слабоконтрастных объектов, к ним могут применяться различные алгоритмы первичной растровой обработки. В качестве базовых алгоритмов в работе рассматривались: яркостная коррекция (нормализация, инвертирование), подавление шума (Despeckle-удаление пятен, медианная, билатеральная фильтрация, усиление резкости), а также морфологические преобразования (эрозии, дилатации, изоляции ярких регионов) [2]. Важным этапом перед применением алгоритмов обработки изображений высокого уровня (распознавания/классификации) является сегментация изображений. Сегментация – это процесс разбиения изображения на «сегменты», которые представляют собой множества пикселей, объединенные по тем или иным признакам. Процедура сегментации позволяет значительно уменьшить объем информации на изображении и облегчить его дальнейший анализ. Для отделения слабоконтрастных объектов от сложного фона применялись алгоритмы глобальной и локальной бинарной сегментации, в результате работы которых исходные полутоновые изображения, имеющее $N = 256$ уровней яркости, преобразовывались в свое черно-белое представление.

3. Выделение признаков объектов

Признаком изображения объекта называется его простейшая отличительная характеристика или свойство, в простейшем случае описываемое одним числовым значением [3]. Для решения задачи распознавания тематических объектов рассматривалась система признаков, состоящая из двух групп:

- признаков формы, инвариантных к сдвигу, масштабу, повороту.
- признаков, основанных на статистических характеристиках яркости объектов.

Среди реализованных признаков описания формы отметим следующие [4]:

- компактность – характеристика, определяемая отношением площади контура к квадрату периметра $d = s/p^2$, где $s = \mu_{0,0}$ – площадь контура (количество пикселей с яркостью отличной от фоновой $I(x, y) > 0$), p – количество пикселей принадлежащих границе области;

- эксцентриситет – числовая характеристика конического сечения контура объекта, показывающая степень отклонения его формы от окружности. Вычислялся по формуле $e = \sqrt{1 - r_{\min}^2 / r_{\max}^2}$, где r_{\min} , r_{\max} – значения малой и большой полуосей контура;

- дисперсионная характеристика контура – характеристика, определяющая степень «изрезанности» контура локализованного объекта.

Вычисляется по формуле $\sigma_r = \sqrt{k^{-1} \sum_{i=1}^k (r_i - m_r)^2}$, где $r = (r_1, r_2, \dots, r_k)$ – вектор расстояний от центра масс объекта до точек, составляющих его границу, $r_i = \sqrt{(x_c - x_i)^2 + (y_c - y_i)^2}$, $i = \overline{1, k}$, (x_i, y_i) – координаты точек границы контура, k – их число, $m_r = k^{-1} \sum_{i=1}^k r_i$ – среднее вектора расстояний.

Среди реализованных «яркостных» признаков, отметим следующие:

- энтропия $L_1 = - \sum_{i=0}^{N-1} F(i) \log_{10} F(i)$, где N – количество градаций

яркости изображения локализованного объекта;

- энергия $L_2 = \sum_{i=0}^{N-1} [F(i)]^2$;

- вариация $L_3 = \sum_{i=0}^{N-1} (i - m_F)^2 F(i)$, где m_F – среднее арифметическое

по гистограмме.

- энергия, вариация, однородность и контраст на основе матрицы смежности, содержащей относительные частоты наличия на изображении соседних элементов, расположенных на расстоянии d друг от друга под углами $0^\circ, 90^\circ$ (горизонтально-вертикальные) или $45^\circ, 135^\circ$ (диагональные пары), с яркостями $i, j \in G$, $G = \{0, 1, \dots, N-1\}$ [3].

4. Обучение и тестирование классификаторов

Для решения задачи распознавания объектов (на примере изображений зерновых смесей) в работе использовались два типа классификаторов: двухслойные НС класса многослойный перцептрон, подробно рассмотренные в работе, а также машины опорных векторов SVM. В обоих случаях задача распознавания решалась как бинарная – классификаторы обучались относить каждый распознаваемый объект к одному из двух классов.

При использовании НС рассматривалась сеть следующей архитектуры. Количество входных контактов сети n равнялось числу

используемых признаков распознавания, размер скрытого слоя m_1 выбирался в диапазоне $n \leq m_1 \leq 2n + 1$, выходной слой сети состоял из одного нейрона, на выходе которого формировалось решение в виде $rec = sign(y - 0.5)$, где значения $+1$ и -1 переменной rec соответствуют первому и второму классам объектов. Матрица обучающих данных НС определялась в виде $X = \{x_1^{(a)}, \dots, x_k^{(a)}, x_1^{(b)}, \dots, x_{m-k}^{(b)}\}$, где $x_i^{(a)}, x_j^{(b)}$ – векторы измеренных значений признаков для распознаваемых объектов первого и второго класса, $i = \overline{1, k}$, $j = \overline{1, m-k}$, k – количество векторов признаков для объектов первого класса, m – объем обучающей выборки. Ожидаемые на выходе НС значения определялись компонентами целевой матрицы $Z = \{z_1^{(a)}, \dots, z_k^{(a)}, z_1^{(b)}, \dots, z_{m-k}^{(b)}\}$, где $z_i^{(a)} = (+1.0, \dots, +1.0)$, $i = \overline{1, k}$, $z_j^{(b)} = (-1.0, \dots, -1.0)$, $j = \overline{1, m-k}$. Матрица входных данных для тестирования обученной НС формировалась в виде $\bar{X} = \{\bar{x}_1, \bar{x}_2, \dots, \bar{x}_m\}$, где \bar{x}_i – вектор вычисленных значений признаков распознаваемого объекта, $i = \overline{1, m}$. Матрица \bar{X} нормировалась аналогично X . Получаемые на выходе НС данные обозначались как $Y = \{y_1, y_2, \dots, y_m\}$.

Подробная математическая формулировка задачи классификации с SVM описана в работе [5]. В качестве массивов обучающих/тестирующих данных рассматривались аналогичные описанным выше для НС наборы измеренных признаков для объектов различных классов. При расчетах в методе SVM в качестве ядра классификатора рассматривалась радиальная базисная функция.

При тестировании классификаторов качество распознавания объектов определялось путем оценки вероятности суммарной ошибки распознавания, вычисляемой как $P_{ош} = m_{ош}/m$, где $m_{ош}$ – визуально оцененное число ошибочно классифицированных объектов. В таблице 1 приведены результаты оценки вероятности ошибочного распознавания для изображений смесей зерновых объектов четырех типов, полученные при использовании НС и SVM. Для каждого типа исследуемых зерновых смесей первоначальное обучение классификаторов проводилось с использованием нескольких изображений, содержащих объекты, принадлежащие каждому из двух классов зерен в отдельности (рис. 2).

Таблица 1

Результаты экспериментальной проверки алгоритмов анализа смешанного потока объектов для зерновых смесей различных типов

| Тип распознаваемых объектов | Вероятность ошибочного распознавания объектов | |
|------------------------------|---|--------|
| | НС | SVM |
| «Подсолнечник – соя» | 1,64 % | 1,81 % |
| «Гречиха – овес» | 1,52 % | 1,29 % |
| «Пшеница – ячмень» | 5,07 % | 8,13 % |
| «Фундук хороший – дефектный» | 3,55 % | 4,77 % |

Дальнейшее тестирование проводилось на изображениях, содержащих смеси разнотипных объектов. При обучении и тестировании классификаторов использовалась полная система признаков.

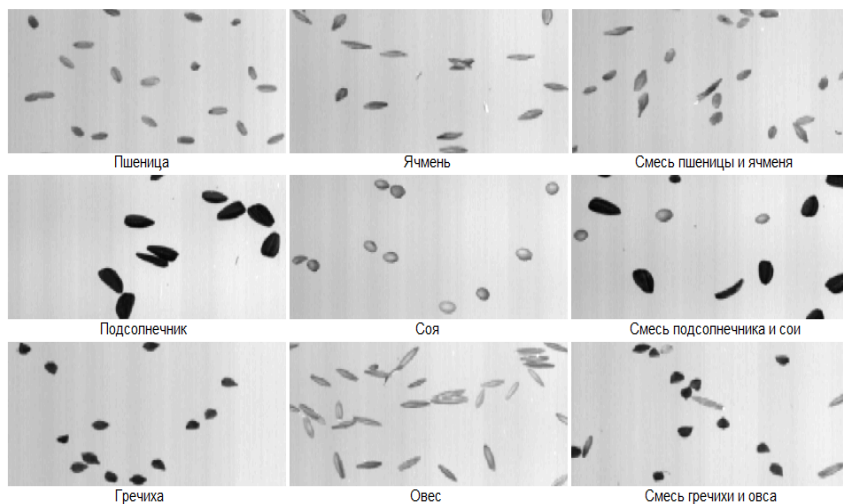


Рис. 1. Фрагменты типовых образцов обрабатываемых изображений

Заключение

Полученные результаты показывают, что алгоритм SVM в среднем показал несколько худшие результаты по сравнению с НС. Следует также отметить, что на результаты распознавания оказывали влияние посторонние объекты в виде примесей (они отдельно не классифицировались и не отсеивались при распознавании). Также следует отметить, что на практике очень часто встречаются случаи, когда рассматриваемые объекты на изображении расположены вплотную друг к

другу, образуя т.н. «склеенные» объекты, анализ последних должен предполагать предварительное их разделение.

Практическая реализация алгоритмов распознавания слабоконтрастных объектов выполнена на языке C++ с использованием библиотек Qt-4.7.4, OpenCV-2.2, Fast Artificial Neural Network Library [6] и LIBSVM [7]. При разработке использовались отдельные процедуры программного модуля анализа изображений потока объектов [8]. Среди функциональных возможностей реализованного программного модуля, дополняющих [8] можно отметить возможность настройки и использования альтернативного классификатора SVM, а также возможность использования более широкого набора признаковых описателей формы и яркости распознаваемых объектов. Разрабатываемый программный модуль может быть использован для распознавания и анализа объектов произвольной формы на тематических растровых изображениях произвольного качества.

Список литературы

1. Аппаратно-программный комплекс для анализа неоднородного потока объектов в системах фотосепарации реального времени / Э.К. Алгаинов [и др.] // Измерительная техника: оптико-физические измерения. – 2014. – №5. – С. 23-29.
2. Гонсалес, Р. Цифровая обработка изображений / Р. Гонсалес, Р. Вудс. – Москва : Техносфера, 2005. – 1072 с.
3. Напряшкин, А. А. Алгоритмическое и программное обеспечение системы интерпретации аэрокосмических изображений для решения задач картирования ландшафтных объектов / А. А. Напряшкин. – Дис. канд. техн. наук. – Томск, 2002. – 183 с.
4. Потапов, А. А. Новые информационные технологии на основе вероятностных текстурных и фрактальных признаков в радиолокационном обнаружении малоконтрастных целей / А. А. Потапов // Радиотехника и электроника. – 2003. – Т. 48. – № 9. – С. 1101-1119.
5. Методы измерения спектральных характеристик и распознавания элементов зерновых смесей в системах сепарации реального времени / Э. К. Алгаинов [и др.] // Измерительная техника. – 2014. – №1. – С. 36-41.
6. Fast Artificial Neural Network Library [Electronic resource]. – URL : <http://leenissen.dk/fann/wp>
7. Library for Support Vector Machines [Electronic resource]. – URL : <http://www.csie.ntu.edu.tw/~cjlin/libsvm>
8. Программный модуль для анализа изображений неоднородного потока объектов / Э.К. Алгаинов [и др.] // Свидетельство о государственной регистрации программы для ЭВМ №2014617546 от 28.07.2014.

АДАПТИВНОЕ СВЕТОФОРНОЕ РЕГУЛИРОВАНИЕ НА МОДЕЛИ ИЗОЛИРОВАННОГО ПЕРЕКРЕСТКА

Н. А. Болдинова

Студент

А. С. Беляев

Старший преподаватель

Введение

Транспортные пробки являются основным источником не только загрязнения воздуха, но и колоссальных экономических издержек, связанных с потерянными в заторах часами и задержкой доставок. Поэтому оптимальная организация дорожного движения транспортных средств, в частности эффективное светофорное регулирование является одной из наиболее актуальных проблем в настоящее время.

Существует два основных подхода к светофорному регулированию:

- постоянное регулирование;
- адаптивное регулирование [1].

Первый способ является традиционным и имеет широкое распространение [2]. Однако этот подход, основанный на использовании фиксированных по длительности фаз светофора, рассчитанных на основе статистических данных, не учитывает текущую обстановку на перекрестке, которая зачастую носит нестабильный характер. Такое упущение учитывает адаптивный подход.

1. Цель эксперимента

В данной статье рассматривается реализация адаптивного способа светофорного регулирования.

В качестве входных параметров, описывающих текущую ситуацию были выбраны:

- число транспортных средств, ожидающих на направлении с зеленым сигналом перед переключением сигнала;
- число транспортных средств, ожидающих на направлении с красным сигналом перед переключением сигнала;
- число транспортных средств, прибывающих на направлении с зеленым сигналом.

На основе входных параметров принималось решение о продлении текущей фазы светофора.

На разрабатываемый способ светофорного регулирования накладывались следующие ограничения:

- применим для светофоров без дополнительных секций;
- не учитывает пешеходную фазу;
- применим для изолированного перекрестка.

Сравнение результатов реализованного способа управления проводилось с результатами работы светофорного регулирования, основанном на использовании фиксированных циклов на примере модели конкретного перекрестка. При этом в критерии оценки его эффективности реализованного способа учитывались:

- среднее число транспортных средств, ожидающих на перекрестке;
- среднее время ожидания;
- среднее время проезда через перекресток.

2. Нечеткие нейронные сети

Основная проблема при реализации алгоритма светофорного регулирования заключается в том, что на сегодняшний день нет единой математической теории управления транспортными потоками и на данный момент нет общепринятого подхода, описывающего поведение движения автотранспорта [3]. Именно поэтому наиболее подходящим вариантом решения поставленной задачи является представление светофорного регулирования в виде «черного ящика». Для решения этой задачи можно применить искусственные нейронные сети, поскольку именно этот подход обладает свойством обобщения [4]. При этом какую-либо априорную информацию (знание эксперта) для ускорения процесса ее обучения ввести невозможно [5]. Эту проблему позволяет решить использование концепции нечетких нейронных сетей, которая объединяет в себе достоинства аппарата нейронных сетей и теории нечетких множеств. В таком подходе выводы делаются на основе аппарата нечеткой логики, а соответствующие функции принадлежности подстраиваются с использованием алгоритмов обучения нейронных сетей. Такая концепция позволяет не только использовать априорную информацию, но и приобретать новые знания [5].

3. Нечеткий логический вывод

В основе топологии используемой нечеткой нейронной сети лежат элементы нечеткого логического вывода, согласно которому входные и выходные переменные необходимо представить в виде лингвистических переменных, которые описываются соответствующим набором нечетких множеств. Нечеткие множества характеризуются функциями принадлежности [6]. На рис. 1 представлен пример набора функций принадлежности для выходной переменной времени продления фазы.

Аналогичным образом определены функции принадлежности для каждой из входных переменных.

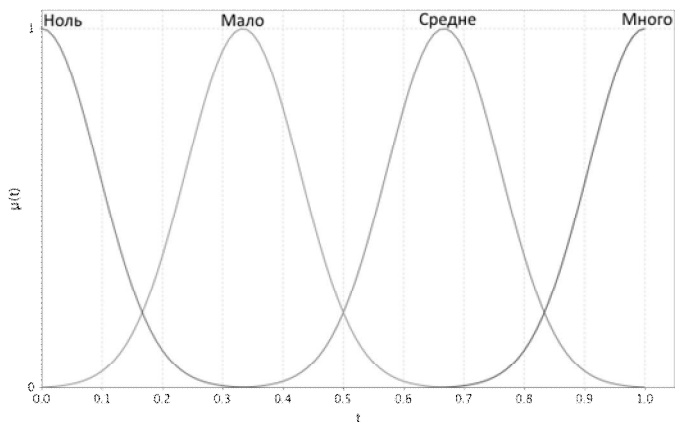


Рис. 1. Функции принадлежности для лингвистической переменной t

Часть используемой базы правил, построенной на определенных таким образом нечетких множествах, представлена на рис. 2.

| | Предпосылка | Заключение |
|-----|---|-------------------|
| 1 | Если q_f это Мало и q_g это Мало и a_g это Мало | то t это Ноль |
| 2 | Если q_f это Мало и q_g это Средне и a_g это Мало | то t это Мало |
| 3 | Если q_f это Мало и q_g это Много и a_g это Много | то t это Средне |
| 4 | Если q_f это Мало и q_g это Очень Много и a_g это Много | то t это Много |
| 5 | Если q_f это Средне и q_g это Много и a_g это Много | то t это Средне |
| 6 | Если q_f это Много и q_g это Средне и a_g это Мало | то t это Ноль |
| 7 | Если q_f это Много и q_g это Очень Много и a_g это Средне | то t это Мало |
| 8 | Если q_f это Много и q_g это Очень Много и a_g это Много | то t это Средне |
| ... | ... | ... |
| 48 | Если q_f это Очень Много и q_g это Много и a_g это Много | то t это Мало |

Рис. 2. Фрагмент базы правил

4. Нечеткая нейронная сеть

Топология используемой нечеткой нейронной сети представлена на рис. 3.

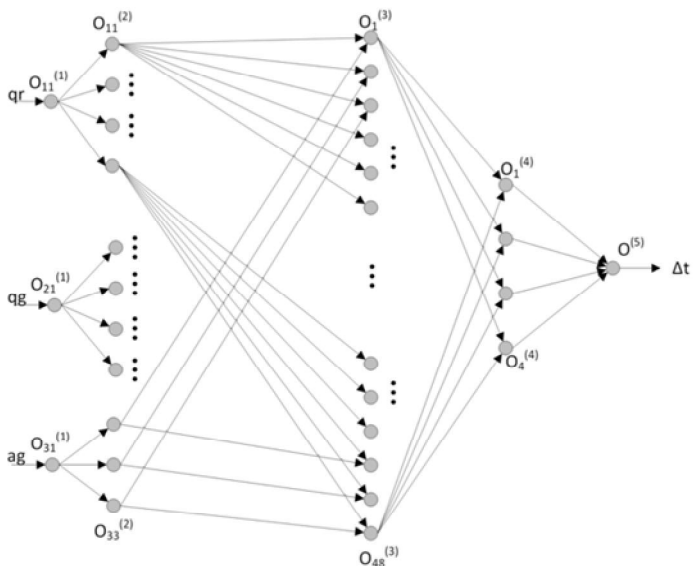


Рис. 3. Нечеткая нейронная сеть

Каждому слою соответствует определенный этап нечеткого вывода Мамдани [7]. Количество нейронов на каждом слое, порядок их соединения также определяются отмеченными элементами нечеткого вывода [7]. В данной сети параметры функции активации на втором слое и веса на четвертом слое настраиваются в ходе обучения с учителем на основе генетического алгоритма.

5. Реализация и эксперимент

Представленный подход был реализован с использованием языка программирования Scala и следующих библиотек:

- Neugorph (реализация нейронной сети);
- Watchmaker (генетический алгоритм обучения сети).

Для создания модели перекрестка использовались средства, реализованные на языке Scala.

Для проведения эксперимента была построена модель четырехстороннего изолированного перекрестка с тремя полосами движения на каждом из направлений (рис. 4).

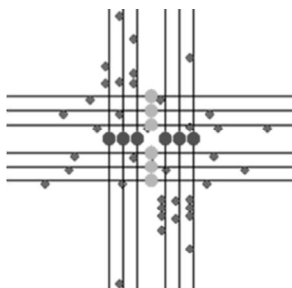


Рис. 4. Модель перекрестка

Параметры модели, необходимые для проведения эксперимента были измерены на одном из загруженных перекрестков города Воронежа (таблица).

Таблица

Параметры перекрестка

| Название | Значение |
|-------------------------------------|-----------------|
| Максимальная интенсивность | 0,84 авт./сек. |
| Умеренная интенсивность | 0,51 авт./сек. |
| Минимальная продолжительность фазы | 40 сек. |
| Максимальная продолжительность фазы | 60 сек. |

В ходе эксперимента были рассмотрены различные ситуации загруженности двух улиц перекрестка и получены следующие результаты. При использовании адаптивного подхода наблюдалось сокращение:

- среднего размера очереди на 10,23%;
- среднего времени ожидания на 9,57%;
- среднего времени ввоза через перекресток на 9,08%.

Заключение

В данной статье рассматривается реализация адаптивного способа светофорного регулирования на основе нечеткой нейронной сети. Описывается ее структура, входные и выходные переменные. В ходе проведения эксперимента на модели наблюдалось улучшение показателей эффективности по сравнению с использованием постоянного светофорного регулирования, что доказывает его преимущество в рамках построенной модели.

Список литературы

1. Расчет временных параметров сигналов светофора // Транспортная инженерия [Электронный ресурс]. – Режим доступа : <http://www.againc.net/ru/education/transport-engineering>

2. Traffic Control Systems Handbook // FHWA Office of Operations [Электронный ресурс]. – Режим доступа : http://ops.fhwa.dot.gov/publications/fhwahop06006/chapter_3p1.htm
3. Введение в математическое моделирование транспортных потоков / А. В. Гасников [и др.]. – Москва : Изд. МФТИ, 2010. – 362 с.
4. Осовский, С. Нейронные сети для обработки информации / С. Осовский пер. с польского И. Д. Рудинского. – Москва : Финансы и статистика, 2002. – 344 с.
5. Круглов, В. В. Искусственные нейронные сети. Теория и практика / В. В. Круглов, В. В. Борисов. – Москва : Горячая линия-Телеком, 2002. – 382 с.
6. Рутковская, Д. Нейронные сети, генетические алгоритмы и нечеткие системы / Д. Рутковская, М. Пилиньский, Л. Рутковский; пер. с польск. И. Д. Рудинского. – Москва : Горячая линия – Телеком, 2006. – 452 с.
7. Alodat, M. Using Polygamy technology with FI, GA, and NN on traffic lights / M. Alodat, I. Alodat // The International Journal of Engineering and Science. – 2013. – Vol. 2. – № 7. – P. 39-45.

ОЦЕНКА ВЕРОЯТНОСТНО-ВРЕМЕННЫХ ХАРАКТЕРИСТИК БЕСПРОВОДНЫХ СЕТЕЙ СВЯЗИ НА ОСНОВЕ МОДЕЛИ В СИСТЕМЕ NS2

Д. Ю. Большаков

Студент

Н. Н. Винокурова

Доцент

Введение

Сегодняшний стремительный скачок в развитии телекоммуникационной области однозначно обязан своим появлением компьютерной революции произошедшей буквально 10 лет назад. Сейчас же мы являемся свидетелями сетевой революции, ибо именно в сети мы находим необходимую информацию, именно в сеть помещаются огромные репозитории информации. Одной из быстро развивающихся областей в компьютерных сетях являются беспроводные сети, обладающие одним из важных достоинств – они позволяют передавать данные на расстоянии движущимся объектам.

Network Simulator 2 позволяет смоделировать сеть, провести ее анализ, выявить ту которая является наиболее оптимальной для дальнейшего использования [1].

1. NS2 (Network Simulator 2)

Проведение экспериментов на базе ns2 и изучение его архитектуры влечет за собой получение знаний в ряде областей (программирование, телекоммуникации, математика) без которых невозможно современное существование высококвалифицированного IT-специалиста. На базе ns2/nam возможна организация наглядной демонстрации функционирования протоколов и сетевых механизмов, например, влияния дисциплины обслуживания очереди на вероятность потери пакета трафика с разными приоритетами, или в чем заключается различие алгоритмов протокола TCP (slowstart, sliding window, SACK и т. д.) [2].

2. Цель эксперимента

Цель эксперимента в данной статье – оценка вероятностно-временных характеристик беспроводных сетей связи на основе модели в системе ns2 (таблица).

Сети, которые будут рассматриваться, могут использоваться в качестве сенсорных для мониторинга различных параметров (температура, влажность, давление), передачи данных между мобильными абонентами в условиях отсутствия подготовленной инфраструктуры (базовых станций). В качестве области использования, например, сеть связи МЧС при ликвидации последствий чрезвычайных ситуаций, сенсорная сеть мониторинга состояния урожая в сельском хозяйстве и т. п.

3. Определение, моделирование беспроводных сетей

Для начала было необходимо построить в ns2 модели беспроводных сетей с площадями от 310x310 до 350x350 метров. После этого было проведено имитационное моделирование, с целью увидеть, как работает данная сеть.

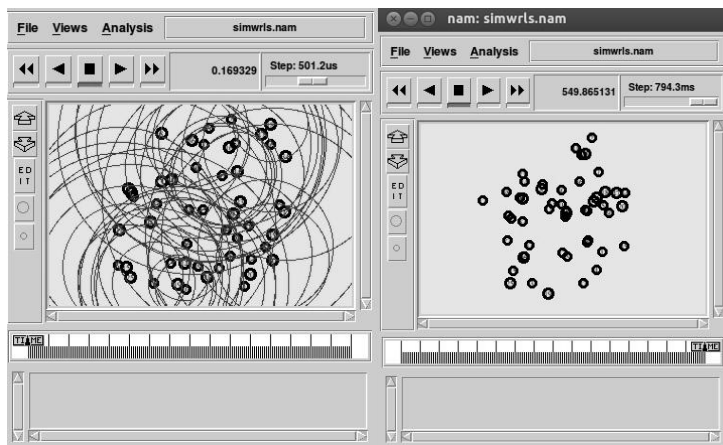
Моделировались сети, состоящие из 50 узлов. В заданный момент времени 25 узлов сети создавали соединения с 25 узлами.

Сети расположены на территории от 310x310 (рис. 1) до 350x350 (рис. 2) метров, в качестве антенн использовались антенны с круговой диаграммой направленности в горизонтальной плоскости (omnidirectional). В случайно выбранный момент времени каждый из узлов случайным образом выбирал точку на заданной территории и начинал к ней движение. Станции моделировались подвижными со случайными скоростями, лежащими в диапазоне от 0 до 20 м/с.

Параметры моделируемой сети

| | |
|------------------------------------|--------------------|
| Протокол транспортного уровня | UDP |
| Протокол маршрутизации | AODV |
| Протокол MAC уровня | CSMA/CA |
| Модель распространения радиоволн | TwoRayGround |
| Тип трафика | cbr |
| Тип антенны | omnidirectional |
| Высота антенны, м | 0,5 |
| Частота, МГц | 914 |
| Ширина полосы пропускания, МГц | 2 |
| Мощность передатчика, Вт | 0,2818 |
| Пороговая мощность приема, Вт | $3,652 * 10^{-10}$ |
| Пороговая мощность обнаружения, Вт | $1,559 * 10^{-11}$ |

В качестве модели трафика сети использовался cbr (constantbitrate), трафик с постоянной интенсивностью. В процессе моделирования каждым из передающих узлов сети отправлялось 51200 байт пользовательской информации. В качестве модели распространения радиоволн используется двулучевая модель сигнала с отражением одного из лучей от поверхности (TwoRayGround).

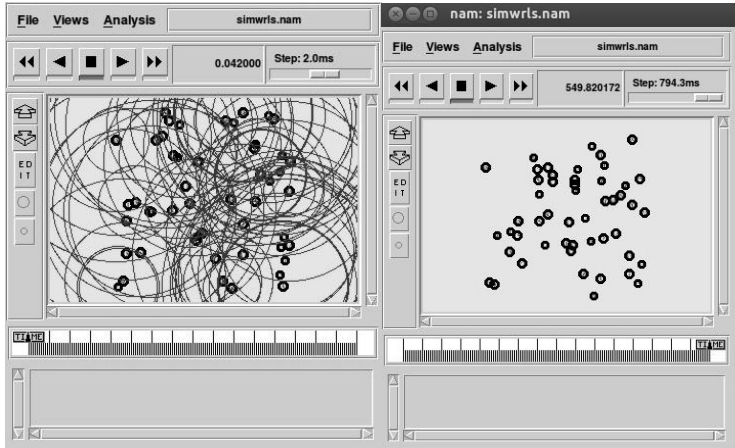


а

б

а – начальное расположение, б – конечное расположение

Рис. 1. Сеть с территорией 310x310



а

б

а – начальное расположение, б – конечное расположение

Рис. 2. Сеть с территорией 350x350

4. Статистика смоделированных сетей

После того как были смоделированы модели беспроводных сетей, при помощи программы tracelograph был проведен анализ с использованием трафиков cbr-пользовательский трафик с постоянной интенсивностью аналогичный передаче речи и потоковому видео и aodv-служебный трафик протокола маршрутизации AODV, отвечающего за построение маршрутов между узлами сети, для получения статистики (рис. 3 – рис. 6).

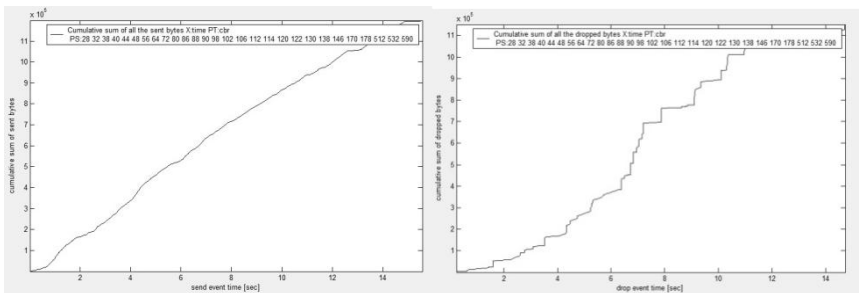


Рис. 3. Количество переданных и потерянных байт (cbr 310-310)

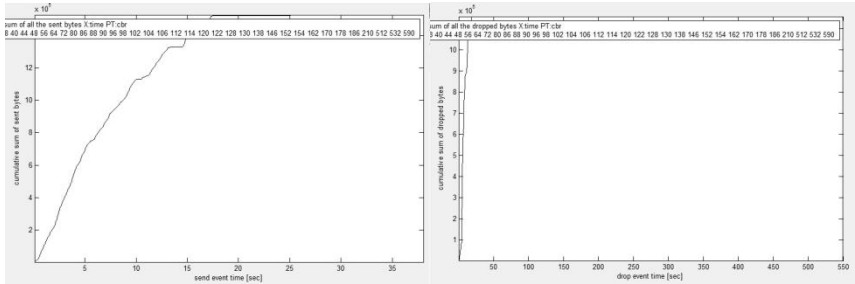


Рис. 4. Количество переданных и потерянных байт (cbr 350-350)

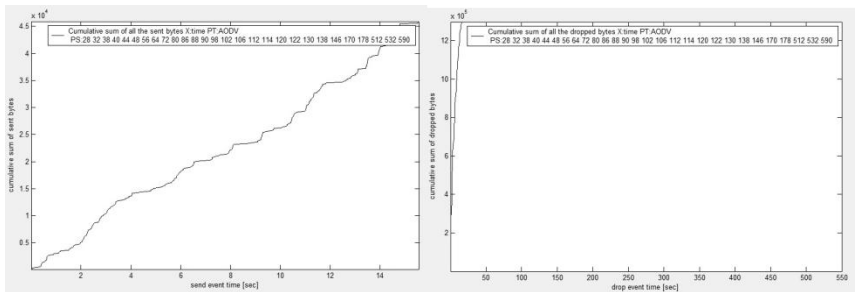


Рис. 5. Количество переданных и потерянных байт (aodv 310-310)

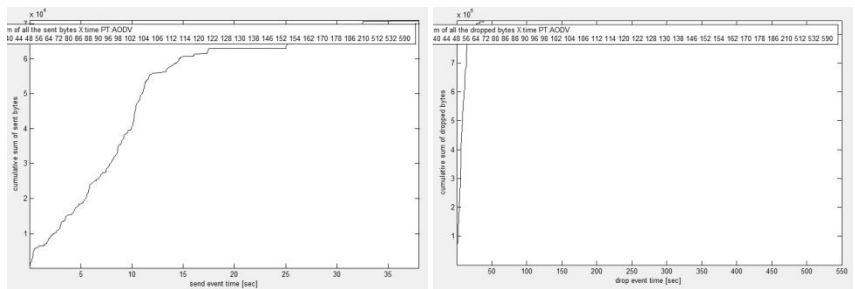


Рис. 6. Количество переданных и потерянных байт (aodv 350-350)

Заклучение

Данная статья посвящена моделированию беспроводных сетей в NS2, с последующим анализом. В итоге данного эксперимента, исходя из полученных анализов, наиболее оптимальными сетями являются модели с площадью 330-330 и 350-350 с использованием трафика (cbr). Количество переданных байт намного превышает количество потерянных. По критерию минимизации потерянной информации может быть больше, чем

переданной так как при потерях происходит повторная передача пакетов, но повторно передаваемые пакеты уже не учитываются при подсчете передаваемой информации

Потеря информации обусловлена мобильностью узлов, что приводит к разрыву соединений и коллизиям при конкурентном доступе к разделяемой среде передачи информации.

Использование NS2 позволяет решать задачи с минимальными временными и умственными затратами, что при условии постоянного роста конкуренции является несомненным преимуществом.

Список литературы

1. NS-2 для исследования телекоммуникационных сетей [Электронный ресурс]. – Режим доступа : <http://simulation.su/uploads/files/default/2007-hrapov-1.pdf>

2. О пакете Network Simulator [Электронный ресурс]. – Режим доступа : <http://www.networksimulation.ru>

ОПИСАНИЕ ОДНОГО КЛАССА 6-МЕРНЫХ МАТРИЧНЫХ АЛГЕБР ЛИ

В. С. Бондаренко

Студент

А. П. Бубнов

Студент

А. В. Лобода

Профессор

Введение и постановка задачи

В работе изучаются 6-мерные алгебры Ли, состоящие из комплекснозначных квадратных матриц 4-ого порядка, имеющих специальную структуру. Вопрос о таких алгебрах возникает в связи с задачей описания аффинно-однородных поверхностей в многомерных пространствах (см., например, [1]-[3]). Отметим без детализации, что обсуждаемые ниже алгебры относятся к случаю аффинно-однородных поверхностей $(1/2, \varepsilon)$ -типов.

Напомним, что матричной алгеброй Ли называется линейное пространство матриц, замкнутое относительно коммутатора (скобки)

$$[A, B] = A \cdot B - B \cdot A \quad (1)$$

В силу квадратичного характера выражения (1) относительно А и В и достаточно больших размеров массивов неизвестных, используемых в данной задаче, ее приходится решать на основе алгоритмических подходов (см. [4]).

Везде далее рассматриваются 6-мерные матричные алгебры Ли с базисами вида

$$\begin{aligned}
 E_1 &= \begin{pmatrix} A1_1 & A2_1 & A3_1 & 1 \\ B1_1 & B2_1 & B3_1 & 0 \\ 2i\mu_1 & 0 & 2m_1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, & E_2 &= \begin{pmatrix} A1_2 & A2_2 & A3_2 & i \\ B1_2 & B2_2 & B3_2 & 0 \\ 2\nu_1 & 0 & 2(m_2 - i\mu) & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \\
 E_3 &= \begin{pmatrix} A1_3 & A2_3 & A3_3 & 0 \\ B1_3 & B2_3 & B3_3 & 1 \\ 0 & 2i\mu_2 & 2m_3 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, & E_4 &= \begin{pmatrix} A1_4 & A2_4 & A3_4 & 0 \\ B1_4 & B2_4 & B3_4 & 1 \\ 0 & 2\nu_2 & 2m_4 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, & (2) \\
 E_5 &= \begin{pmatrix} A1_5 & A2_5 & A3_5 & 0 \\ B1_5 & B2_5 & B3_5 & 0 \\ -\mu & 0 & 2(m_5 + i\lambda) & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}, & E_6 &= \begin{pmatrix} A1_6 & A2_6 & A3_6 & 0 \\ B1_6 & B2_6 & B3_6 & 0 \\ 0 & 0 & 2m_6 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}.
 \end{aligned}$$

Необходимо найти все такие алгебры, удовлетворяющие следующим ограничениям

$$\mu_1 = 2; \nu_1 = 0; \mu_2 = (1 + 2\xi); \nu_2 = (1 - 2\xi); \mu = 0. \quad (3)$$

Здесь подразумевается, что параметры $m_1, m_2, m_3, m_4, m_5, m_6, \lambda$, входящие в матрицы (2), являются вещественными, $m_6 \neq 0$, а параметр ε удовлетворяет условиям

$$0 < \varepsilon \neq \frac{1}{2} \quad (4)$$

(связанным с понятием типа поверхности).

Основной результат

ТЕОРЕМА. Всякая 6-мерная алгебра Ли, удовлетворяющая описанным выше условиям, имеет базис вида ($t_7 \in R$):

$$\begin{aligned}
E_1 &= \begin{pmatrix} 3t_7 & 0 & \frac{1}{2}it_7 & 1 \\ 0 & t_7 & 0 & 0 \\ 4i & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, E_2 = \begin{pmatrix} 0 & 0 & 0 & i \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, E_3 = \begin{pmatrix} 0 & t_7(1+2\varepsilon) & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 2i(1+2\varepsilon) & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \\
E_4 &= \begin{pmatrix} 0 & -i(1-2\varepsilon)t_7 & 0 & 0 \\ 0 & 0 & 0 & i \\ 0 & 2-4\varepsilon & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, E_5 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}, E_6 = \begin{pmatrix} 2 & 0 & -it_7 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}. \quad (5)
\end{aligned}$$

Доказательство этого утверждения получается за счет изучения большой системы матричных уравнений, представляющих необходимые условия для того, чтобы набор матриц вида (2) был базисом некоторой алгебры Ли. Все необходимые вычисления реализованы в пакете Maple (см. [5]).

Описание вычислений

Переходя к таким вычислениям, заметим, прежде всего, что за счет рассмотрения вместо исходных матриц (2) их линейных комбинаций можно считать выполненными следующие равенства

$$m_1 = 0; m_2 = 0; m_3 = 0; m_4 = 0; m_5 = 0; m_6 = 1. \quad (6)$$

Далее уточним, что для построения отдельной алгебры достаточно подобрать элементы шести базисных матриц таким образом, чтобы коммутатор (скобка)

$$[E_k, E_l] = E_k \cdot E_l + E_l \cdot E_k \quad (7)$$

любых двух таких матриц разлагался по исходному базису, т. е.

$$[E_k, E_l] = r_1 \cdot E_1 + r_2 \cdot E_2 + r_3 \cdot E_3 + r_4 \cdot E_4 + r_5 \cdot E_5 + r_6 \cdot E_6. \quad (8)$$

при некоторых вещественных r_1, \dots, r_6 . Для удобства будем обозначать скобку как

$$W_{kl} = [E_k, E_l].$$

Необходимо рассмотреть $C_6^2 = 15$ скобок, и для каждой из них должно выполняться равенство вида (8).

Пятая матрица рассматриваться не будет, т. к. ее можно выразить через все остальные матрицы, например, из скобки $[E_1, E_2]$. Пять скобок, связанные с матрицей E_5 , ниже также не обсуждаются.

Совокупность 10 оставшихся таких матричных равенств типа (8) представляет собой набор необходимых (но, вообще говоря, не достаточных) условий для того, чтобы вещественная линейная оболочка

$\langle E_1, E_2, E_3, E_4, E_5, E_6 \rangle$ была алгеброй Ли.

У элементов левых верхних 2×2 -блоков базисных матриц выделим вещественные и мнимые части, вводя для этого 48 вещественных параметров. Тогда любой обсуждаемый базис примет вид:

$$E_1 = \begin{pmatrix} t_1 + it_2 & t_3 + it_4 & A3_1 & 1 \\ t_5 + it_6 & t_7 + it_8 & B3_1 & 0 \\ 2i\mu_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, E_2 = \begin{pmatrix} t_9 + it_{10} & t_{11} + it_{12} & A3_2 & i \\ t_{13} + it_{14} & t_{15} + it_{16} & B3_2 & 0 \\ 2\nu_1 & 0 & 2i\mu & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \quad (9)$$

$$E_3 = \begin{pmatrix} t_{17} + it_{18} & t_{19} + it_{20} & A3_3 & 0 \\ t_{21} + it_{22} & t_{23} + it_{24} & B3_3 & 1 \\ 2i\mu_1 & 0 & 2i\mu_2 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, E_4 = \begin{pmatrix} t_{25} + it_{26} & t_{27} + it_{28} & A3_4 & 0 \\ t_{29} + it_{30} & t_{31} + it_{32} & B3_4 & i \\ 0 & 2\nu_2 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix},$$

$$E_5 = \begin{pmatrix} t_{33} + it_{34} & t_{35} + it_{36} & A3_5 & 0 \\ t_{37} + it_{38} & t_{39} + it_{40} & B3_5 & 0 \\ -\mu & 0 & 2i\lambda & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}, E_6 = \begin{pmatrix} t_{41} + it_{42} & t_{43} + it_{44} & A3_6 & 0 \\ t_{45} + it_{46} & t_{47} + it_{48} & B3_6 & 0 \\ 0 & 0 & 2m_6 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

Для произвольной матрицы M , принадлежащей линейной оболочке матриц E_1, \dots, E_6 , коэффициенты разложения M по базису определяются видом ее четвертого столбца и следующими формулами:

$$r_1 = \text{Re}(M_{14}), r_2 = \text{Im}(M_{14}), r_3 = \text{Re}(M_{24}), r_4 = \text{Im}(M_{24}), r_5 = \text{Re}(M_{34}).$$

Используя полученные формулы, можно перейти от рассмотрения скобок к их исправленным вариантам. Подставляя для этого в формулы (2) в качестве матрицы M скобки W_{kl} , вычислим для каждой из таких скобок набор коэффициентов r_1, \dots, r_5 . Исправленной скобкой R_{kl} будем называть сумму матриц

$$R_{kl} = W_{kl} - (r_1 \cdot E_1 + r_2 \cdot E_2 + r_3 \cdot E_3 + r_4 \cdot E_4 + r_5 \cdot E_5)$$

именно с такими коэффициентами. Ясно, что четвертый столбец каждой из 10 обсуждаемых исправленных скобок – нулевой, а потому любая исправленная скобка R_{kl} должна быть пропорциональна матрице E_6 . Ниже приводится фрагмент Maple-программы, связанный с вычислением исправленных скобок:

```
for i from 1 to 5 do
for j from i + 1 to 6 do if ((i ≠ 5) and (j ≠ 5) then
r[1] := Re(W[i, j][1,4]); r[2] := Im(W[i, j][1,4]);
r[3] := Re(W[i, j][2,4]); r[4] := Im(W[i, j][2,4]);
```

```

r[5] := Re(W[i, j][3,4]);
R[i, j] := evalm(W[i, j] - (r[1]E[1] + r[2]E[2] + r[3]E[3] + r[4]E[4] + r[5]E[5]));
endif; end do; end do;

```

Получаем большое количество (а именно, 10) матричных уравнений. Отдельные матричные уравнения можно разбить на скалярные компоненты в количестве $9 \cdot 10 = 90$ (по числу элементов и скобок) уравнений. Даже с учетом использования пакета символьных вычислений исследование системы из 90 комплексных уравнений является чрезмерно трудоемким. При этом большим является и количество параметров в базисных матрицах, которые являются неизвестными в обсуждаемой системе.

Уменьшение количества неизвестных в системе можно связать с (3,1)- и (3,2)-элементами исправленных скобок. Эти элементы зависят ЛИНЕЙНЫМ образом от неизвестных t_k и достаточно просто вычисляются. При этом в силу пропорциональности исправленных скобок матрице E_6 эти элементы должны быть нулевыми. Таким образом, мы получим систему из 20 комплексных линейных соотношений. В вещественной форме это дает $20 \cdot 2 = 40$ линейных уравнений.

Подбором находим 28 уравнений и переменных, относительно которых эти уравнения maple может решить однозначно:

$$\begin{aligned}
S = \{ & t_2 = -2t_9, t_3 = -t_5\varepsilon - \frac{1}{2}t_5 + 2t_{17}, t_4 = -t_6\varepsilon + \frac{1}{2}t_6 - 2t_{25}, t_{10} = 0, \\
& t_{11} = -t_6\varepsilon - \frac{1}{2}t_6, t_{12} = t_5\varepsilon - \frac{1}{2}t_5, t_{13} = 0, t_{14} = 0, t_{16} = \frac{t_{21}(-1+2\varepsilon)}{4\varepsilon}, \\
& t_{18} = t_6\xi + \frac{1}{2}t_6, t_{20} = -t_{22}\xi + \frac{1}{2}t_{22} + 2t_8\xi - t_{22}, t_{24} = -\frac{1}{2} \frac{t_{31}(1+2\xi)}{\xi}, \\
& t_{27} = -2\varepsilon t_8 - 2\varepsilon t_{15} - t_{15}, t_{28} = \frac{1}{2} \frac{8\varepsilon^2 t_7 - 4\varepsilon t_{21} + 4\varepsilon t_{21} - 2t_7 - t_{21}}{1+2\varepsilon}, \\
& t_{29} = 2t_{15}, t_{30} = \frac{t_{21}(-1+2\varepsilon)}{1+2\varepsilon}, t_{32} = \frac{1}{2} \frac{t_{23}(-1+2\varepsilon)}{\varepsilon}, t_{41} = 2, t_{43} = 0, \\
& t_{44} = 0, t_{45} = 0, t_{46} = 0, t_{47} = 2, t_{48} = 0, t_{26} = t_5\varepsilon - \frac{1}{2}t_5 \}.
\end{aligned}$$

Подстановка решений этих 28 уравнений в оставшуюся часть линейной системы обнуляет все ее уравнения. Это означает, что ранг системы из 40 уравнений равен в точности 28.

С подставленными значениями считаем исправленные скобки каждой из первоначальных матриц с 6-ой матрицей. Как и ранее, вычитаем из каждой такой матрицы-скобки исходные базисные матрицы так, чтобы в

последнем столбце остались нули. Тогда можно считать, что полученная матрица пропорциональна матрице E_6 .

Из этого можно сделать выводы о значениях еще некоторых параметров. Например, многие такие исправленные матричные скобки пропорциональны E_6 с нулевым коэффициентом. Значит можно приравнять к нулю все элементы такой матрицы и решить полученные комплексные уравнения, разбив их на вещественные и мнимые части (обе будут равны нулю). Например:

$$RR_{4,6} = \begin{pmatrix} t_{25} & -A3_6(2-4\varepsilon) + 4\varepsilon A3_6 + 2A3_6 + \frac{i(8\varepsilon^2 t_7 - 2t_7)}{1+2\varepsilon} & A3_6 t_{25} + A3_4 & 0 \\ -4A3_6 & \frac{i(-1+2\varepsilon)t_{23}}{1+2\varepsilon} & -2A3_6^2 + 2B3_4 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

Заметим, что в силу полученных выше (из линейной системы) равенств $t_{47} = 2$ и $t_{48} = 0$ [2,2]-элемент матрицы E_6 не содержит мнимой части. Значит, и весь [2,2]-элемент матрицы $RR_{4,6}$ равен нулю. Тогда $RR_{4,6}$ пропорциональна матрице E_6 с нулевым коэффициентом, т. е. все элементы $RR_{4,6}$ являются нулевыми.

Таким образом, вручную сравнив каждую из полученных выше скобок с матрицей E_6 , получили, что ещё несколько параметров t_k , а также несколько элементов $A3_k$ и $B3_k$ базисных матриц равны нулю. Одно из промежуточных состояний базиса искомым алгебр Ли приведено ниже

$$E_1 = \begin{pmatrix} t_7 & -3it_{25} & A3_1 & 1 \\ 0 & t_7 & B3_1 & 0 \\ 4i & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, E_2 = \begin{pmatrix} 0 & 0 & A3_1 & i \\ 0 & 0 & B3_1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix},$$

$$E_3 = \begin{pmatrix} 0 & 2\varepsilon t_7 + t_7 + i(-\varepsilon t_{22} + \frac{1}{2}t_{22}) & A3_3 & 0 \\ it_{22} & 0 & B3_3 & 1 \\ 0 & 2i(1+2\varepsilon) & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, E_5 = \begin{pmatrix} t_{39} & 0 & 0 & 0 \\ 0 & t_{39} & 0 & 0 \\ it_{39} & 0 & 2i\lambda & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix},$$

$$E_4 = \begin{pmatrix} t_{25} & \frac{i(8\varepsilon^2 t_7 - 2t_7)}{2i(1+2\varepsilon)} & A3_4 & 0 \\ 0 & 0 & B3_4 & i \\ 0 & 2-4\varepsilon & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, E_6 = \begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

Можно заметить, что в матрице E_6 осталось «мало» элементов, что существенно облегчает исследование матричных скобок.

Выводы

Признаком того, что алгебра Ли найдена, служит тот факт, что каждая из матричных скобок (вычисленных по элементам базиса) разлагается по исходному базису, составленному из матриц $E_1 \dots E_6$. Такому требованию, как показывают вычисления, удовлетворяют лишь базисы вида (5), приведенные выше.

Список литературы

1. Лобода, А. В. Об одном семействе аффинно-однородных вещественных гиперповерхностей 3-мерного комплексного пространства / А.В. Лобода, А.С. Ходарев. – Известия ВУЗов. Математика. – 2003. – №10. – С. 38-50.
2. Лобода, А. В. О полном списке аффинно-однородных поверхностей $(\varepsilon, 0)$ -типов в пространстве C^3 / А. В. Лобода, А. В. Шиповская. – Известия ВУЗов. Математика. – 2015. – №6. – С. 75-82.
3. Atanov, A. V. Affine homogeneous strictly pseudoconvex hypersurfaces of the type $(1/2, 0)$ in C^3 / A. V. Atanov, A. V. Loboda, A. V. Shipovskaya [Электронный ресурс]. – Режим доступа : [ArXiv.org/abs/1401.2252](http://arXiv.org/abs/1401.2252)
4. Кокс, Д. Идеалы, многообразия и алгоритмы / Д. Кокс, Дж. Литтл, Д. О'Ши. – Москва : Мир, 2000. – 688 с.
5. Кирсанов, М. Н. Практика программирования в системе Maple / М. Н. Кирсанов. – Москва : Издательский дом МЭИ, 2011. – 208 с.

ИНТЕГРАЦИЯ ОБРАЗОВАТЕЛЬНОГО ПОРТАЛА MOODLE С ИНФОРМАЦИОННОЙ СИСТЕМОЙ УНИВЕРСИТЕТА

А. В. Васильев

Студент

А. П. Толстобров

Доцент

Введение

В Воронежском государственном университете в качестве платформы для организации электронного обучения используется образовательный портал «Электронный университет ВГУ» на базе популярной системы

© Васильев А. В., Толстобров А. П., 2015

e-learning Moodle [1]. Эта электронная образовательная среда предоставляет широкие возможности для создания и использования электронных курсов по дисциплинам образовательных программ ВГУ, она содержит большое количество разработанных и используемых преподавателями ВГУ электронных курсов, в ней зарегистрировано и подписано к курсам несколько тысяч студентов.

В связи с все возрастающей активностью использования этой системы, чрезвычайно актуальной стала необходимость автоматизации процессов поддержки согласованности электронных учебных курсов с дисциплинами учебных планов образовательных программ университета, а также выполнения массовых операций по групповой регистрации студентов в портале и их подписке на конкретные электронные курсы.

В настоящее время в университете бизнес-процессы разработки и поддержки использования учебных планов образовательных программ, расчета учебной нагрузки по учебным подразделениям, управления контингентом студентов обеспечиваются Информационной системой ВГУ, в базе данных которой, реализованной на СУБД Oracle, содержится вся необходимая информация об образовательных программах ВГУ и обучающихся по ним студентам.

Таким образом, решение проблем обеспечения более эффективной работы образовательного портала потребовало решения задачи интеграции образовательного портала и Информационной системы ВГУ. Решение этой комплексной задачи рассматривается в настоящей работе.

Модульная структура и открытый код программного обеспечения портала на базе системы Moodle позволяют решать задачи расширения функционала этой системы путем разработки и подключения к ней дополнительных программных модулей-плагинов без необходимости модификации программного кода ядра системы.

Поэтому, для решения поставленной задачи были разработаны на языке PHP несколько модулей, решающих конкретные подзадачи, рассмотренные ниже.

1. Модуль связи электронных курсов с учебными планами

Этот модуль реализует сервис, позволяющего осуществлять привязку электронных курсов портала Moodle к конкретным дисциплинам учебных планов образовательных программ ВГУ, по которым рассчитывается учебная нагрузка.

В общем случае, создаваемый в образовательном портале ВГУ электронный курс обычно предназначен для обеспечения учебного процесса по нескольким одинаковым или схожим учебным дисциплинам, принадлежащих разным учебным планам нескольких образовательных программ. Поэтому задачей этого сервиса является предоставление

возможности доступа оператора к базе данных учебных планов для выборки из нее необходимых данных о соответствующих конкретному электронному курсу учебных дисциплинах и привязки этих дисциплин к конкретному электронному курсу.

Сервис рассчитан на то, чтобы все манипуляции по привязке электронного курса к дисциплинам учебных планов осуществлялись без участия администратора портала самим преподавателем, разрабатывающим и использующим конкретный электронный курс.

Для выполнения этих действий преподаватель, находясь на странице курса, используя интерфейс разработанного и размещенного на странице курса блока «Уч. планы и студенты», может обратиться к экранной форме для выбора соответствующих образовательных программ.

В этой форме он должен указать атрибуты соответствующего учебного плана (рис. 1):

- Факультет;
- Степень образования – бакалавр/специалист/магистр;
- Специальность/направление – код и наименование образовательной программы);
- Специализация/профиль – в этом пункте указывается раздел учебного плана, к которому относится дисциплина, – базовая часть учебного плана или наименование соответствующей специализации (профиля);
- Форма обучения.

Карточка электронного курса - Рабочий курс

[Вернуться к курсу](#)

[Изменить тип образовательной программы](#)

Дисциплина Виды учебной работы по учебному плану

▼ Привязка электронного курса к дисциплинам учебных планов основных образовательных программ

[Вернуться к курсу](#)

Факультет:

Степень:

Специальность:

Специализация / Профиль:

Форма обучения:

Дисциплина:

[Добавить дисциплину](#)

*Если дисциплина относится к базовой части учебного плана, то в строке "Специализация/Профиль" следует выбрать пункт "Базовая часть учебного плана."

Рис. 1. Форма привязки курса к дисциплинам учебных планов

Устанавливаемые в форме параметры выбираются непосредственно из базы данных Oracle информационной системы поддержки учебных планов и расчета нагрузки, что обеспечивает актуальность этих данных.

В последнем пункте формы, в соответствии с установленными параметрами учебного плана, выбирается код и наименование дисциплины, к которой привязывается новый электронный курс.

После нажатия кнопки «Добавить дисциплину», выбранная дисциплина с ее атрибутами из учебного плана включается в Карточку электронного курса. В этой Карточке отображается также выбираемая из базы данных учебных планов дополнительная информация о семестре и количестве часов по видам учебных работ по дисциплине (рис. 2).



Рис. 2. Вид Карточки электронного курса

Аналогичным образом преподаватель осуществляет выборку и включение в Карточку курса соответствующих дисциплин из других учебных планов.

2. Модуль автоматизированной групповой регистрации студентов в портале и их записи на курсы

Разработан модуль-плагин, который реализует сервисы по автоматизации создания учетных записей для студентов еще не зарегистрированных в портале и подписки студентов на курс, в соответствии с относящимися к этому курсу дисциплинами образовательных программ, по которым эти студенты в настоящее время обучаются.

Сервис также рассчитан на то, чтобы все манипуляции по формированию заявки на регистрацию и подписку студентов

осуществлялись самим преподавателем, разрабатывающим и использующим конкретный электронный курс, без привлечения администратора портала.

При формировании электронной заявки на регистрацию студентов и их запись на курс, список записываемых студентов формируется путем выборки информации о них из базы данных Oracle информационной системы управления контингентом студентов ВГУ.

Сформированный список студентов может быть отредактирован преподавателем и автоматически включен в формируемую электронную заявку, которая затем также автоматически поступает на обработку администратору портала.

Последующая автоматизированная обработка записей о студентах, включенных в электронную заявку, состоит в следующем:

- если включенный в заявку студент уже зарегистрирован в портале, то есть он уже имеет учетную запись с персональным аккаунтом (логин и паролем), то этому студенту добавляются права доступа к данному курсу, определяемые пользовательской ролью «Студент»;

- если у включенного в заявку студента учетная запись отсутствует, то вначале осуществляется его регистрация в портале как нового пользователя. При этом ему присваивается персональный аккаунт (логин и пароль), по которым он может авторизовываться в портале. Затем ему предоставляются права доступа к указанному в заявке электронному курсу, определяемые пользовательской ролью «Студент».

Созданный модуль позволяет реализовывать два способа формирования электронной заявки для записи студентов на электронный курс.

1. Автоматическое формирование электронной заявки путем выборки списка студентов из базы данных информационной системы «Контингент» в соответствии с контингентом, обучающимся по образовательными программам, к дисциплинам учебных планов которых привязан данный электронный курс. Такая привязка осуществляется с помощью модуля, рассмотренного выше.

2. Ручное формирование электронной заявки для записи студентов на курс без использования привязки курса к учебным планам образовательных программ, используемая в случаях, когда в базе данных системы «Контингент» по каким-либо причинам отсутствует привязка студентов к соответствующей образовательной программе.

Ниже рассматриваются оба этих способа.

При выборе первой процедуры автоматизированного формирования списка подписываемых на курс студентов в соответствии с образовательными программами, к дисциплинам которых привязан

данный электронный курс, в интерфейсе модуля выделяются пункты-ссылки на дисциплины учебных планов образовательных программ, к которым привязан данный электронный курс.

При выборе в этой форме наименования конкретной дисциплины курса открывается относящаяся к этой дисциплине информация о дисциплине, выбираемая из соответствующего учебного плана, и ссылки на академические группы студентов, которые в соответствии с учебным планом должны в данном учебном году изучать эту дисциплину. У каждой группы в скобках указывается количество входящих в нее студентов.

Щелчок по наименованию группы открывает список студентов этой группы, выбранный из базы данных ИС «Контингент» (см. рис. 3).

► Б3.Б.4 Управление данными

▼ Б3.Б.13 Базы данных

Дисциплина: Б3.Б.13 Базы данных
Ступень: Бакалавр
Направление: 010200 Математика и компьютерные науки
Профиль: -
Форма обучения: Очная
Кафедра ответственная за дисциплину: Кафедра информационных систем
Год учебного плана: 2014-2015
Семестры: 3
Год поступления студентов: 2013

Отметить Снять всех студентов.
Группа: '5' (12 студентов)
Отметить Снять

| | | |
|--------------------------|-------|--|
| <input type="checkbox"/> | 1. / | имов Сергей Павлович (16130112) |
| <input type="checkbox"/> | 2. / | рев Сергей Владимирович (16130080) |
| <input type="checkbox"/> | 3. / | Окола Еришел Бюрел (16135001) |
| <input type="checkbox"/> | 4. / | ухов Александр Сергеевич (16130085) |
| <input type="checkbox"/> | 5. / | ников Алексей Александрович (16130097) |
| <input type="checkbox"/> | 6. / | лин Антон Витальевич (16130107) |
| <input type="checkbox"/> | 7. / | Никита Сергеевич (16130102) |
| <input type="checkbox"/> | 8. / | ва Дарья Александровна (16130109) |
| <input type="checkbox"/> | 9. / | Сергей Дмитриевич (16130095) |
| <input type="checkbox"/> | 10. / | в Николай Александрович (16130113) |
| <input type="checkbox"/> | 11. / | ов Дмитрий Валерьевич (05110045) |
| <input type="checkbox"/> | 12. / | ько Алексей Юрьевич (16130162) |

Рис. 3. Работа со списком регистрируемых студентов

В открывшемся списке студентов группы преподаватель должен отметить галочками тех студентов, которые должны быть включены в электронную заявку для записи на курс.

Автоматическая обработка указанного в заявке списка студентов состоит в проверке учетной записи студента, его регистрации в портале, если до этого он еще не был зарегистрирован, и последующей записи студента на курс с правами доступа, определяемыми пользовательской ролью «Студент».

После того, как заявка будет обработана, у записанных студентов появится значок, свидетельствующий о том, что студент подписан на курс.

В ряде случаев возникает необходимость записи на курс студентов, обучающихся по образовательным программам, не привязанным к данному курсу. Например, при обучении студента по индивидуальному плану или ином случае, когда в базе данных системы «Контингент» по каким-либо причинам отсутствует привязка студентов к соответствующей образовательной программе.

Для этих целей предназначен сервис записи студентов, доступ к которому осуществляется выбором соответствующего пункта в разработанном блоке «Уч. планы и студенты».

При выборе этого пункта открывается форма для ввода параметров для выборки студентов из базы данных системы «Контингент». Вводятся следующие параметры:

- наименование факультета,
- год поступления,
- форма обучения,
- степень обучения,
- наименование направления/специальности.

После установки параметров выборки студентов и нажатия кнопки «Загрузить» под формой ввода появляется список выбранных групп студентов. Щелчок мышкой по пункту с названием группы открывает список входящих в группу студентов.

Обязательным условием записи студентов на курс является наличие у них номера студенческого билета. Этот номер становится логином студента. Номер студенческого билета выбирается из базы данных ИС «Контингент» и указывается в списке студентов в скобках рядом с фамилией, именем и отчеством студента. В случае отсутствия в базе данных информации о номере студенческого билета, рядом с ФИО студента отображается окно для ручного ввода этого номера при записи студентов на курс.

Студенты, которых необходимо записать на курс, отмечаются в списке галочками. После этого электронная заявка отправляется администратору нажатием кнопки «Отправить заявку».

В секции «Поданные заявки», расположенной в нижней части формы, отображается номер заявки, ее статус, и количество записываемых студентов.

3. Модуль отписывания студентов от курса

Важным сервисом, реализуемым данным модулем, является также возможность отписывания студентов от электронного учебного курса. Эта операция обычно используется после того, как студенты заканчивают обучение по данному курсу, для того, чтобы записи этих студентов не перегружали текущие отчеты по выполнению студентами учебных

поручений. Важно отметить, что отписывание студента от курса не означает удаления из базы данных системы информации об учебных достижениях этих студентов при прохождении курса. Вся эта информация сохраняется в базе данных портала. При необходимости, можно получить к ней доступ вновь подписав студента на данный курс.

Для того чтобы перейти к сервису отписывания студентов от курса, необходимо в блоке «Уч. планы и студенты» выбрать ссылку «Отписывание студентов от курса». После этого на странице курса появляются пункты-ссылки с наименованиями групп подписанных на курс студентов, при выборе которых открывается список студентов, входящих в соответствующую группу.

В открывшемся списке следует отметить галочками тех студентов, которые должны быть отписаны от курса.

После нажатия кнопки «Отписать» отмеченные студенты будут исключены из числа участников данного курса.

Заключение

Программные модули-плагины, обеспечивающие интеграцию образовательного портала ВГУ на базе системы Moodle с Информационной системой университета, были реализованы на языке PHP, подключены к программному обеспечению портала Moodle, и успешно апробированы в 2014-2015 учебном году.

Реализуемые ими сервисы реализованы таким образом, что управление ими осуществляется самим преподавателем, курирующим конкретный электронный курс, что существенным образом улучшает оперативность массового выполнения этих операций в начале учебных семестров. Важно подчеркнуть, также, что использование реализованных сервисов обеспечивает актуальный состав подписанных на электронные курсы студентов в соответствии информацией о контингенте студентов, выбираемой из базы данных Информационной системы управления контингентом студентов.

Список литературы

1. Образовательный портал «Электронный университет ВГУ». [Электронный ресурс] / Воронежский государственный университет. – Режим доступа : <http://www.moodle.vsu.ru>

РАЗРАБОТКА SMART TV ПРИЛОЖЕНИЯ ДЛЯ ПРОСМОТРА IPTV-ТЕЛЕВИДЕНИЯ

Г. Н. Вознесенская

Студент

Д. И. Соломатин

Старший преподаватель

Введение

На сегодняшний день сетевые технологии очень широко распространены. Они используются для подключения к интернету, для IP-телефонии и, в том числе, для IP-телевидения (IPTV). Технология IPTV позволяет транслировать цифровое телевидение через IP-сети с видеопотоками HD-разрешения и многоканальным звуком. Благодаря этой передовой технологии общение с телевизором превратилось в диалог [1].

Для того чтобы пользователь мог в полной мере использовать возможности IPTV, производители телевизоров усовершенствовали возможности своей продукции и появилось новое поколение телевизоров под названием «Smart TV». Такие телеприемники называют еще «умные телевизоры». Используя технологию Smart TV, пользователи могут запускать на экране телевизора различные Smart TV приложения.

В данной работе рассматривается реализация Smart TV приложения для просмотра IPTV для конкретного провайдера.

1. Обзор аналогов

Многие операторы в стране, предоставляющие услуги доступа в Интернет, также предоставляют услуги IPTV-телевидения. Для каждого провайдера важно, чтобы пользователь мог в полной мере воспользоваться всеми сервисами, в том числе и новой технологией. При этом существует два варианта: создать свое приложение или просить пользователя использовать существующее.

На большинстве моделей телевизоров для просмотра IPTV конкретного провайдера, требуется только правильно сформировать плейлист. Данное решение имеет ряд неудобств. Во-первых, пользователь должен сам формировать плейлист и не всегда это является простым действием. Во-вторых, не на всех моделях телевизоров существуют такие приложения. Например, у Philips для просмотра IPTV имеется буквально одно приложение, предоставляемое компанией Trinity своим абонентам. В-третьих, не покрывается весь спектр предоставляемых услуг провайдера. Пользователь не сможет формировать списки любимых каналов,

использовать архив телепередач и родительский контроль, получать справочную информацию (о расчётных периодах, балансе) и т. д., поэтому многие провайдеры прибегают к разработке собственного продукта.

2. Общая структура Smart TV приложения

Приложение Smart TV – это веб-страница определенного типа, которая выполняется веб-браузером, встроенным в Smart TV телевизор. Просмотр такой страницы аналогичен просмотру веб-страниц на обычном компьютере. Отличия заключаются в разрешении экрана, аппаратном обеспечении и в интерфейсе пользователя, который представляет собой пульт дистанционного управления [2].

Таким образом, клиентская часть Smart TV приложений разрабатывается на HTML + CSS + JavaScript. Серверная часть в рассматриваемом приложении разработана на PHP.

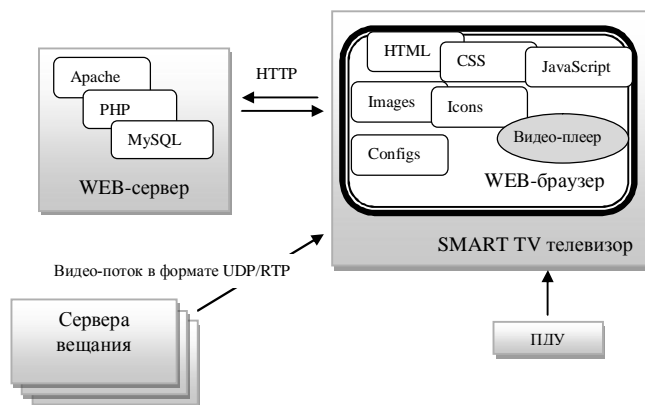


Рис. 1. Общая структура Smart TV приложения для просмотра IPTV

Компонент «Видео-плеер» – это платформо-зависимое расширение браузера. Работа с ним осуществляется при помощи API, предоставляемого вендорами.

3. Библиотека SmartBox

Изучение особенностей разработки Smart TV приложений показало, что реализация кроссплатформенного приложения сразу для нескольких вендоров – непростая задача. Имеются проблемы даже при реализации приложения для телевизоров одного производителя. Так, например, платформы Samsung 2011 и Samsung 2104 сильно отличаются друг от друга. Поэтому разработка клиентской части приложения производилась

на базе библиотеки Smartbox, которая позволяет создавать кроссплатформенные Smart TV приложения и предоставляет удобный функционал для их реализации.

4. Клиентская часть приложения

Клиентская часть приложения включает:

- файлы конфигурации – для телевизоров Samsung создаются два специальных конфигурационных файла: config.xml и widget.info. Данные файлы содержат всю необходимую информацию для работы виджета. Также создается специальный файл-обертка для телевизоров Philips, который обеспечивает запуск приложения в браузере телевизора;
- файлы стилей и различных ресурсов;
- файлы, отвечающие за логику приложения (JavaScript) – обработка нажатий клавиш пульта, обращение к серверу за видеопотоками, списками каналов, анонсами и т.д.;
- главный файл входа в приложения – index.html.

В приложении может быть только один файл index.html. С учетом этой особенности вся структура данной страницы была разбита на div-блоки, каждый из которых представляет сложный элемент, также состоящий из блоков (см. рис. 2).

Верхнее меню предназначено для переключения страниц. Каждый элемент является div-блоком, который содержит атрибут page – название страницы. При щелчке на этот элемент считывается содержимое атрибута и отображается соответствующая страница.

Страницы – это объекты одной структуры, хранящиеся в глобальном объекте window.App.pages.*. Они показываются и скрываются в зависимости от того, какой элемент основного меню был выбран. Каждая страница имеет одинаковую структуру: два основных блока – меню и контейнер со сценами. В блоке Меню каждый пункт – название сцены для показа. Переключение по сценам осуществляется аналогично верхнему меню для показа страниц. Контейнер для сцен содержит структуру сцен. Сцены, как и страницы, являются объектами одной структуры. Каждая сцена хранится в глобальном массиве, принадлежащем определенной странице. Сцена отвечает за выполнение какой-либо функции из набора, предоставляемого страницей.

Взаимодействие с серверной частью приложения осуществляется с помощью асинхронных AJAX-запросов по протоколу HTTP. Данные возвращаются в формате JSON. В реализации активно использовалась библиотека JQuery.

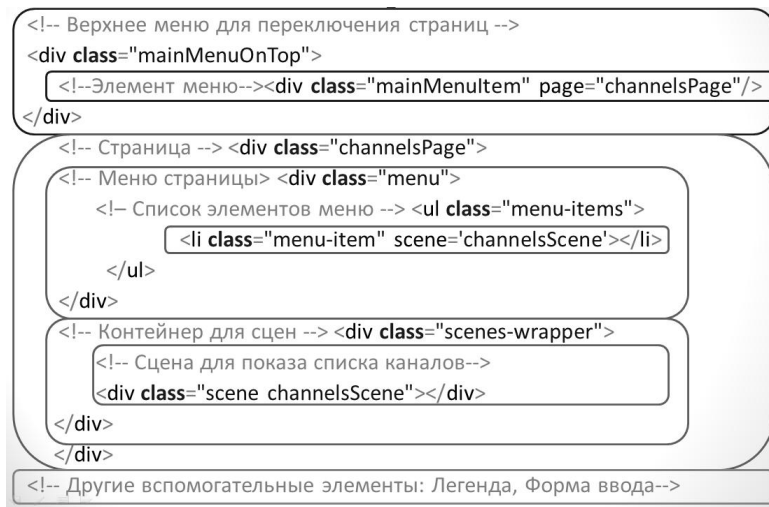


Рис. 2. Фрагмент страницы index.html

7. Тестирование

В разработке приложений одним из самых важным этапов является тестирование. Тестирование Smart TV приложений осуществляется в несколько этапов:

- тестирование в браузере;
- тестирование в эмуляторах вендоров;
- тестирование на конечных устройствах.

Первый пункт не является специфичным, так как требуется просто отлаживать программу в веб-браузере.

Для запуска приложения в эмуляторе Samsung последней версии необходим сам эмулятор, который является образом для VirtualBox. После импорта образа, необходимо создать общую папку, указав путь до директории с проектами. Название директории в виртуальной машине должно быть Apps с соблюдением регистра.

Производители Smart TV телевизоров LG и Philips также предоставляют разработчикам эмуляторы своих платформ.

Тестирование приложение на конечных устройствах также имеет свои особенности. Для этого необходимо четко следовать всем инструкциям, написанным специально для разработчиков.

8. Интерфейс пользователя

При входе в приложение открывается главная страница. На ней находится форма авторизации, новостная информация, экран с последней просмотренной передачей и меню навигации по приложению (рис. 3).



Рис. 3. Главная страница приложения

При переходе на пункт меню «Каналы», осуществляется переход на страницу с каналами. На данной странице есть меню с категориями «Все каналы» и «Списки избранного», сформированные самим пользователем (рис. 4). При переводе фокуса на название канала открывается сцена с телепрограммой по данному каналу на ближайшее время. Текущая программа выделена и отображается индикатор длительности. При фокусе на название телепрограммы открывается сцена с анонсом или описанием. Весь интерфейс приложения полупрозрачный, поэтому пользователь может делать настройки и смотреть телепередачу одновременно.

На вкладке «ТВ Программа» можно просмотреть ТВ программу по конкретному каналу на неделю вперед или по списку каналов на текущий день. На вкладке «Параметры» можно посмотреть справочную информацию о текущем балансе и расчетных периодах, задать учетное имя пользователя и пароль. Также имеется возможность сформировать списки избранного и ограничить доступ к некоторым каналам паролем.

В режиме просмотра пользователь может просмотреть информацию о текущей передаче, посмотреть ТВ программу на ближайшее время, увидеть индикатор длительности и произвести все необходимые настройки (рис. 5).

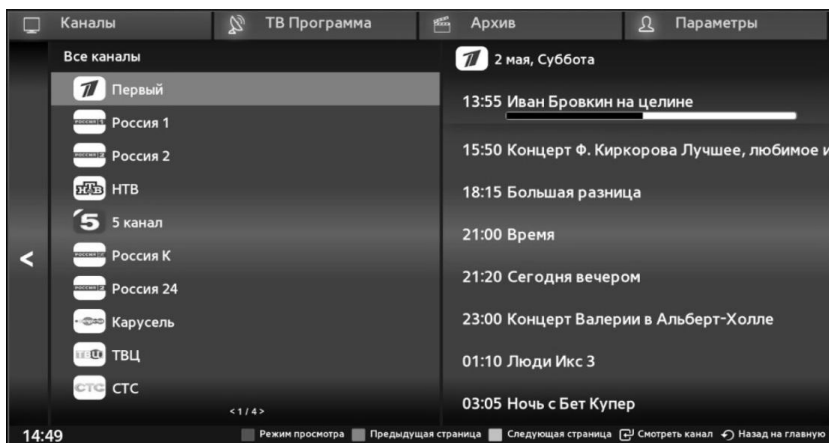


Рис. 4. Страница с каналами



Рис. 5. Режим просмотра

Стоит сказать, что все сцены в приложении автоматически скрываются через 30 секунд после отсутствия действий со стороны пользователя.

Заключение

В результате проделанной работы было реализовано кроссплатформенное Smart TV приложение для просмотра IPTV конкретного провайдера со следующими возможностями:

- поддерживаются три платформы: Samsung, LG, Philips;

- поддерживаются разные виды представления списка доступных каналов, режим просмотра каналов, а также возможность формирования списков избранных каналов;
- имеется несколько вариантов просмотра программы передач с различными опциями на время и на канал;
- в случае если пользователь не успел посмотреть телепрограмму во время, он может воспользоваться архивом;
- поддерживается функции родительского контроля и установка/изменение пароля для приложения;
- имеется страница со справочной информацией о расчетных периодах и балансе;
- удобный и понятный функционал для облегчения работы с приложением, в том числе и легенда.

Список литературы

1. Технология IPTV [Электронный ресурс]. – Режим доступа : <http://itmicro.ru/iptv-technology.html>
2. SS IPTV – телевидение через интернет [Электронный ресурс]. – Режим доступа : <http://televizor-info.ru/eto-interesno/ss-iptv-televidenie-cherez-internet.html>
3. Что такое телевизоры Smart TV? [Электронный ресурс]. – Режим доступа : <http://mylcd.info/model/smart.html>
4. Что такое Smart TV? [Электронный ресурс]. – Режим доступа : <http://smarttv.ru>
5. Создание кроссплатформенного Smart TV приложения на базе библиотеки SmartBox для Samsung, LG, Philips [Электронный ресурс]. – Режим доступа : <http://habrahabr.ru/post/188294>

ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ ФУНКЦИОНАЛЬНОГО МОДУЛЯ ДЛЯ РЕШЕНИЯ ОПТИМИЗАЦИОННЫХ ЗАДАЧ В SAP ERP

Д. С. Дедов

студент

И. В. Илларионов

доцент

Введение

Системы управления предприятием на базе решений SAP уже много лет задают стандарты развития корпоративных технологий. Решения SAP – это комплекс программных продуктов для реализации бизнес-задач, предлагающий широкую функциональность, полную интеграцию и неограниченную масштабируемость. Продукты позволяют реализовать обширный спектр управленческих технологий, таких как: логистика, финансы, управление отношениями с клиентами, производство, управление проектами и т. д. Основной продукт корпоративной линейки – SAP ERP ориентирован на решение следующих задач:

- планирование финансовых, человеческих и материальных ресурсов.
- оперативный контроль выполнения производственных планов.
- учет и анализ результатов хозяйственной деятельности.
- обработка больших объемов данных в реальном масштабе времени.

Одна из наиболее востребованных задач в общей схеме бизнес-процессов предприятия – планирование [1, 2] различных аспектов его функционирования. Здесь активно используются различные математические алгоритмы оптимизации, одним из которых являются методы линейного программирования. Линейное программирование представляет собой математический аппарат, разработанный для решения оптимизационных задач с линейными выражениями. Такие задачи обычно встречаются при решении вопросов оптимального планирования производства с ограниченным количеством ресурсов, при определении оптимального плана перевозки и т. д.

1. Планирование в SAP ERP

Для всех предприятий, работающих на рынке, очень важно спланировать свою деятельность так, чтобы минимизировать свои издержки и получить максимальную прибыль. SAP ERP содержит

программный компонент ориентированный на решение этой задачи – модуль планирования производства (PP). Основное его предназначение – ресурсное обеспечение общего объема промышленного производства с целью наилучшего удовлетворения плановых уровней продаж. Модуль представляет сотрудникам плановых подразделений предприятия полный набор инструментов для создания и оптимизации календарных планов производства. Дальнейшим развитием модуля PP является модуль Advanced Planning & Organization (APO), включающий следующие компоненты:

- Demand Planning (DM) – набор высокопроизводительных приложений для разработки планов и прогнозов, высокой степени достоверности. Компонент Demand Planning позволяет специалистам по планированию разрабатывать прогнозы с использованием различных источников: статистики продаж, информации о клиентах и партнерах, кривыми жизненного цикла товара и т. д.

- Supply Network Planning (SNP) – служит для планирования процессов закупки, производства, распределения в рамках глобальной логистической модели. Главная задача компонента – это построение плана товародвижения по всех логистической цепи. Качественным результатом от использования компонента предполагается получение экономического решения по продажам и распределению.

2. Цель работы

Целью работы является разработка и реализация программного функционального модуля для решения оптимизационных задач в SAP ERP. Данный модуль может использоваться в любых функциональных модулях системы (например, DM, SNP, PP). Модуль должен поддерживать реализацию различных методов оптимизации, в том числе линейного программирования [3]. Данная работа является реализацией базового функционала модуля и будет составлять основу функционирования системы, то есть будет являться некоторой инфраструктурой, к которой в последующем могут быть подключены любые дополнительные методы оптимизации. Языком разработки данного модуля является внутренний язык SAP – Advanced Business Application Programming (ABAP) [4,5].

3. Линейное программирование

Задачей линейного программирования называется задача минимизации (максимизации) значения целевой функции, с учетом ограничений на значения этих переменных, где все зависимости являются линейными [6].

Задача линейного программирования (ЛП) возникает из необходимости оптимально использовать имеющиеся ресурсы. Это задачи,

связанные с целеобразованием и анализом целей и функций; задачи разработки или совершенствования структур (производственных структур предприятия, организованных структур объединения); задачи проектирования (проектирование сложных робототехнических комплексов, гибких производственных систем).

Далее мы будем исходить из следующей формулировки задачи ЛП:

Необходимо определить вектор $(x_1, x_2, \dots, x_n)^T$ максимизирующий /минимизирующий линейную целевую функцию

$$z = c_1 x_1 + c_2 x_2 + \dots + c_j x_j + \dots + c_n x_n \quad (1)$$

при следующих линейных ограничениях

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n \leq b_1, \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n \leq b_2, \\ \dots \\ \dots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n \leq b_m. \end{cases}, \quad (2)$$

$$x_1 \geq 0, x_2 \geq 0, \dots, x_n \geq 0 \quad (3)$$

4. Архитектура исполнения программы

Архитектура (рис. 1) приложения состоит из трех уровней:

- Самым нижним уровнем является уровень базы данных. На этом уровне управление данными осуществляется при помощи системы управления реляционной базой данных (RDBMS).

- АВАР-программы выполняются на уровне сервера приложений, что относится как и приложениям, поставляемым SAP, так и приложениям, разработанным пользователем. АВАР-программы осуществляют взаимодействие с информацией, хранящейся в базах данных и обработку этой информации.

- Третьим уровнем является уровень клиента. Он формирует интерфейс программы и обеспечивает взаимодействие с пользователем. При этом каждый пользователь может вызвать программу, ввести новые данные и получить результат рабочего процесса.

Проектируемый функциональный модуль должен содержать набор оптимизационных алгоритмов, один из которых – алгоритм линейного программирования. Интерфейс модуля включает экран для ввода данных, глобальные объекты из базы данных и подпрограммы, работающие с конкретным алгоритмом.

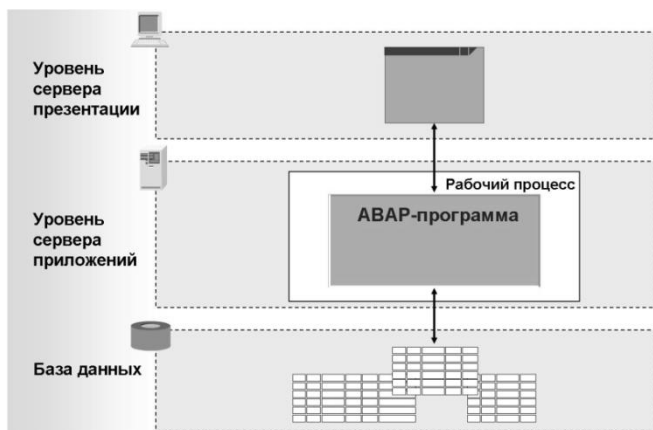


Рис. 1. Архитектура исполнения программы


5. Описание работы программы


Чтобы запустить данную программу нужно ввести в поле транзакции «ZTEST_OPT» (рис. 2) или выбрать во вкладке «Favorites» на главном экране «Methods optimization», или перейти в транзакцию SE80 и ввести имя программы «ZTEST_METHODS» и из подпункта Transaction запустить программу. После этого пользователь попадает в главное окно, на котором задаются необходимые данные для выбранного метода оптимизации (в нашем случае – метод линейного программирования).



Рис. 2. Главное окно модуля

Далее, с помощью первого выпадающего списка выбирается количество переменных для алгоритма, а с помощью второго – количество ограничений. Затем указывается метод для решения задачи (симплекс-метод, метод транспортной задачи или задачи о назначениях). Если поставить галочку (check-box) в поле «EXCEL», то данные можно подгрузить из внешнего файла с расширением .XLS. После заполнения

главного экрана мы нажимаем на клавиатуре кнопку F8 или на экране иконку .

После этого, на новом экране вводим в сформированную таблицу исходные данные решаемой задачи и нажимаем кнопку сохранить , после чего можем видеть полученный результат (рис. 3).

| | | | | |
|------|------|------|------|-------|
| 0,00 | 0,00 | 0,00 | 1,00 | 1,33 |
| 0,00 | 0,00 | 1,00 | 0,00 | 3,33 |
| 1,00 | 0,00 | 0,00 | 0,00 | 3,00 |
| 0,00 | 1,00 | 0,00 | 0,00 | 0,67 |
| 0,00 | 0,00 | 0,00 | 0,00 | 3,33 |
| 0,00 | 0,00 | 0,00 | 0,00 | 1,33 |
| 0,00 | 0,00 | 0,00 | 0,00 | 12,67 |

Рис. 3. Отображение результатов оптимизации

6. Тестирование модуля

Все три алгоритма программирования были проверены на работоспособность конкретными задачами.

Задача симплекс-метода была протестирована задачей планирования и закупки, на примере, производства красок. В качестве исходных данных были заданы продукты, из которых состоят краски, их суточные запасы, а также некоторые критерии и ограничения при изучении рынка.

При проверке транспортной задачи была решена задача транспортных перевозок, при которых общие затраты на перевозку всей продукции были бы минимальны. Изначально были даны склады, содержащие продукцию, потребители, нуждающиеся в определенном количестве продукции, и издержки на перевозку товара конкретным потребителям.

Задача о назначении состояла в назначении работников на определенные работы таким образом, чтобы затраты труда были минимальны. В качестве начальных данных была приведена таблица издержек каждого работника по каждой должности.

Заключение

В ходе работы был реализован функциональный модуль, содержащий реализацию методов линейного программирования, как одного из инструментов решения задачи планирования в ERP системах на базе средств разработки платформы SAP. Этот модуль представляет собой инфраструктуру, к которой в последующем возможно подключение дополнительных алгоритмов оптимизации и использование их в любых функциональных модулях системы.

Список литературы

1. Эддоус, М. Методы принятия решений / М. Эддоус, Р. Стэнфилд // пер. с англ. под ред. член-корр. РАН И. И. Елисеевой. – Москва : ЮНИТИ, 1997. – 590 с.
2. Исследование операций в экономике / Н. Ш. Кремер, Б. А. Путко, И. М. Тришин, М. Н. Фридман : учеб. пособие для вузов / под ред. проф. Н. Ш. Кремера. – Москва : ЮНИТИ, 2002. – 407 с.
3. Банди, Б. Основы линейного программирования / Б. Банди; пер с англ. О. В. Шихеевой; под ред. В. А. Волынского. – Москва : Радио и связь, 1989. – 176 с.
4. Keller, H. ABAP Objects. Introduction to Programming SAP Applications / H. Keller, S. Kruger. – New York : Addison-Wesley Professional, 2002. – 556 p.
5. Кречмер, Р. Разработка приложений SAP R/3 на языке ABAP/4 / Р. Кречмер, В. Вейс. – Москва : издательство Лори, 1998. – 348 с.
6. Коротков, М. Основы линейного программирования / М. Коротков, М. Гаврилов / Санкт-Петербург : «ВНУ-Петербург», 2003. – 154 с.

ПРОГНОЗИРОВАНИЕ СТОИМОСТИ ОБЪЕКТОВ НЕДВИЖИМОСТИ С ПРИМЕНЕНИЕМ НЕЙРОННЫХ СЕТЕЙ

Э. Джумари
Магистрант
А. А. Сирота
Профессор

Введение

Нейронные сети – это одно из направлений исследований в области искусственного интеллекта, основанное на попытках воспроизвести нервную систему человека. А именно: способность нервной системы обучаться и исправлять ошибки, что должно позволить смоделировать, хотя и достаточно грубо, работу человеческого мозга [1]. Важнейшая особенность сети, свидетельствующая о ее широких возможностях и огромном потенциале, состоит в параллельной обработке информации всеми звеньями, что позволяет значительно ускорить процесс обработки информации. Кроме того, при большом числе межнейронных соединений сеть приобретает устойчивость к ошибкам, возникающим на некоторых

линиях. В настоящее время нейронные сети используются для решения целого ряда задач, одной из которых является задача прогнозирования.

Прогнозирование – это предсказание будущих событий. Пусть заданы n дискретных отсчетов $\{y(t_1), y(t_2), \dots, y(t_n)\}$ в последовательные моменты времени t_1, t_2, \dots, t_n . Тогда задача прогнозирования состоит в предсказании значения $y(t_{n+1})$ в некоторый будущий момент времени t_{n+1} .

Целью прогнозирования является уменьшение риска при принятии решений. Прогноз обычно получается ошибочным, но ошибка зависит от используемой прогнозирующей системы. Предоставляя прогнозу больше ресурсов, можно увеличить точность прогноза и уменьшить убытки, связанные с неопределенностью при принятии решений. Типичными приложениями техники прогноза являются предсказание цен на фондовой бирже, прогноз погоды, прогноз потребления электроэнергии, прогноз отказов технических систем и т. д.

В настоящей работе рассматривается применение нейронной сети для решения задачи прогнозирования стоимости объектов недвижимости. Объектом исследования является рынок продажи жилой недвижимости города Воронежа.

Целью исследования является разработка методов прогнозирования модели оценки изменения цен реальных сделок на рынке продажи жилого фонда недвижимости, основанной на нейросетевых технологиях и позволяющей существенно повысить эффективность работы организации, занимающейся продажей жилья.

На начальном этапе проводился анализ известных методов прогнозирования данных с использованием искусственных нейронных сетей.

Вторая задача заключалась в разработке собственной модели и нейросетевого алгоритма стоимости недвижимости, показывающего высокую производительность при решении задач большой размерности.

Третья задача – разработка программы и исследование алгоритма стоимости недвижимости.

При решении указанных задач использовались современные методы теории искусственных нейронных сетей, градиентные методы оптимизации, возможности нейросетевого пакета среды Matlab [2].

1. Методика эксперимента

Для решения перечисленных задач проводилось несколько экспериментов. Целью экспериментов было прогнозирование стоимости недвижимости в городе Воронеж. Использовались данные по реальным объектам недвижимости (квартирам), полученные в период с 1 октября 2014 года по 15 марта 2015 года.

Каждый из экспериментов можно разбить на несколько этапов. Первым этапом было формирование обучающей выборки. На этом этапе определяется вид представления данных и происходит формирование наборов, подаваемых на входные нейроны и соответствующие им наборы снимаемых с выходов сети. На входе сети подаются данные, которые зависят от 8 факторов: район, этажность, тип, материал, количество комнат, жилая площадь, общая площадь, наличие балкона. На выходе сети в качестве целевого значения используется стоимость объекта (квартиры).

Исходные данные преобразуются к виду, в котором их можно подать на входы сети. В рассматриваемом примере есть количественные (этаж, количество комнат, жилая площадь) и неколичественные данные (район, тип). Неколичественные данные подвергаются индексации, а дальше все данные нормируются таким образом: пусть $x = \{x_1, x_2, \dots, x_n\}$ – множество каждого фактора, которое нужно пронормировать, при $n = \overline{1, i}$, i – количество элементов множества x (в нашем примере $i = 489$), тогда:

$$x_{norm} = \frac{x_i}{x_{\max}},$$

где x_{\max} – максимальное значение множество x , x_{norm} – нормируемое значение. В результате данные приводятся к диапазону значений $[0, 1]$.

В качестве архитектуры сети было решено выбрать многослойную нейронную сеть прямого распространения [3] (нейронная сеть типа «многослойной персептрон»), так как ее можно успешно применить для решения большинства задач.

2. Результаты эксперимента.

В ходе первого эксперимента проводилось обучение и тестирование сетей с различной архитектурой. Для анализа прогнозирующего алгоритма тестирование проводилось на обучающих данных. Это позволило оценить потенциально достижимую точность получаемого прогноза. В реальности тестирование ранее обученной нейронной сети необходимо проводить на примерах, которые не участвовали в процессе обучения. При этом число тестовых примеров должно быть тем больше, чем выше качество обучения. Поэтому во втором эксперименте для обоснования прогнозирующего алгоритма проводилось обучение и тестирование на данных, не используемых при обучении. Блок-схема алгоритма, реализованного в ходе второго эксперимента, представлена ниже.

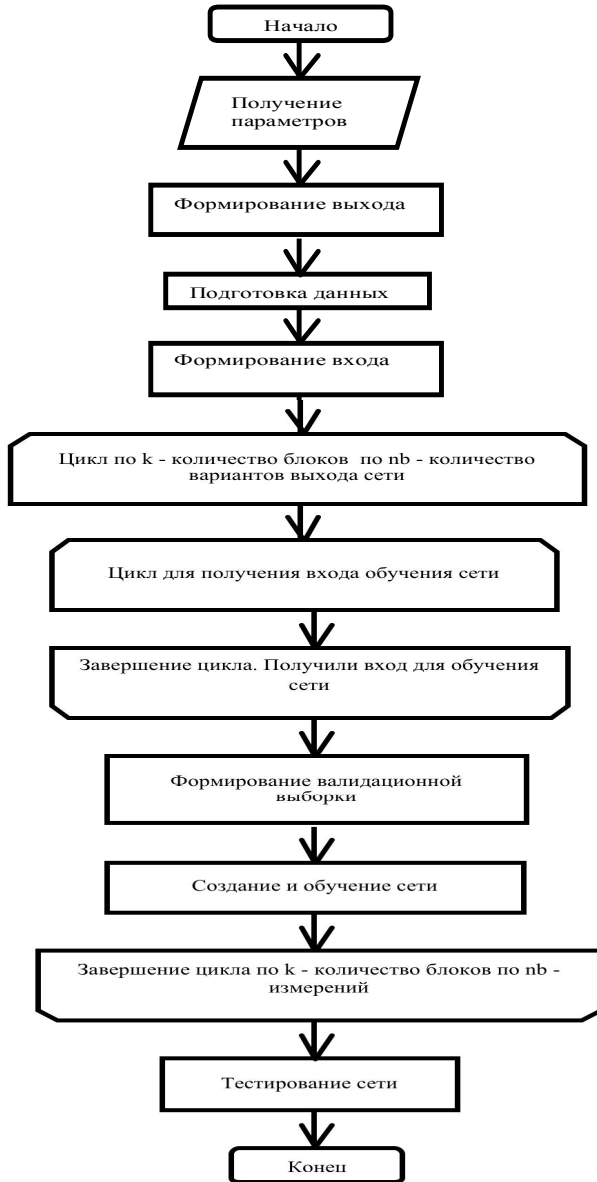


Рисунок. Блок-схема алгоритма

В приведенной ниже таблице показаны результаты экспериментов при разных параметрах архитектуры нейронной сети. В качестве анализируемого показателя используется средняя квадратичная ошибка (СКО) определения стоимости объектов недвижимости.

Таблица

Результаты экспериментов

| Параметры алгоритма | СКО для эксперимента 1 | | | СКО для эксперимента 2 | | |
|------------------------|------------------------|-------|-------|------------------------|-------|-------|
| | Количество эпох | | | Количество эпох | | |
| Нейроны в скрытом слое | 100 | 1000 | 10000 | 100 | 1000 | 10000 |
| 5 | 0.118 | 0.135 | 0.138 | 0.136 | 0.150 | 0.167 |
| 10 | 0.136 | 0.138 | 0.138 | 0.140 | 0.147 | 0.145 |
| 20 | 0.137 | 0.139 | 0.143 | 0.144 | 0.141 | 0.148 |

В левой колонке указаны параметры архитектуры – нейроны в скрытом слое. В верхней строке – средняя квадратичная ошибка для каждого эксперимента при разном количестве эпох, выполняемых в процессе обучения, отображено в колонках. Результаты экспериментов показывают, что для эффективного обучения сети достаточно 100...1000 эпох. При этом количество нейронов в скрытом слое достаточно выбрать порядка 5...10. Некоторое увеличение ошибки при увеличении числа нейронов и количества эпох объясняется недостаточным объемом обучающих данных, что приводит к возникновению эффекта переобучения. В целом общая средняя квадратичная ошибка прогнозирования стоимости недвижимости составила 10-15%.

Заключение

По результатам выполненной работы можно сделать следующие выводы:

1) Нейронные сети являются очень мощным инструментом для работы на финансовых рынках, но для освоения технологии нужно потратить времени уже точно не меньше, чем на освоение технического анализа.

2) Основные проблемы, возникающие при работе с этой технологией – правильная предобработка данных, этот этап играет решающую роль для прогнозирования данных.

3) Среднеквадратическая относительная ошибка прогноза по модели составляет примерно 10-15%. Построенная модель позволяет повысить эффективность управления недвижимостью в масштабах города или крупной корпорации и сделать этот механизм более прозрачным.

Список литературы

1. Осовский, С. Нейронные сети для обработки информации / С. Осовский. – Москва : Финансы и статистика, 2002. – 343 с.
2. Дьяконов, В. Математические пакеты расширения MATLAB. Специальный справочник / В. Дьяконов, В. Круглов. – Санкт-Петербург : Питер, 2001. – 480 с.
3. Кирсанов, Э. А. Обработка информации в пространственно-распределенных системах радиомониторинга: статистический и нейросетевой подходы / Э. А. Кирсанов, А. А. Сирота. – Москва : Физматлит, 2012. – 343 с.

МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ УПРУГОПЛАСТИЧЕСКОГО ГИСТЕРЕЗИСА В НАНОРАЗМЕРНЫХ ФУЛЛЕРЕНОВЫХ ПЛЕНКАХ

А. В. Ещенко

Студент

М. Е. Семенов

Профессор

Введение

Гистерезисные явления возникают в различных областях материаловедения, как на макро-, так и на микроуровне. В зависимости от целей исследования могут использоваться различные модели описания подобных явлений, такие как, например, конструктивные и феноменологические. Как правило, конструктивные модели описывают входно-выходные соотношения системы, при которых не учитываются динамические свойства модели. Однако широко известно, что механические свойства большинства материалов, в том числе рассматриваемое явление упругопластического гистерезиса, сохраняются неизменными, (например, возникновение гистерезиса в упругопластических материалах не зависит от скорости механического воздействия на материал.). Тем не менее, результаты проведенного эксперимента в работе [1] с использованием наноразмерных фуллереновых пленок показывают, что для гистерезисной кривой, построенной в координатах смещения, сила зависит от скорости применяемого воздействия.

В данной работе была рассмотрена и построена математическая модель, а так же были идентифицированы параметры модели,

описывающей возникновение гистерезиса, который наблюдался в работе [1].

1. Фуллерен

В начале XX века впервые был получен фуллерен как вещество. Фуллерен – новая форма существования углерода, одного из самых распространенных элементов на Земле. Фуллерен причислен одновременно и к неорганическим материалам как новая аллотропная модификация углерода, так и к органической химии, поскольку в химических превращениях проявляет свойства непредельных углеродов. Несмотря на давно известную уникальную способность атомов углерода связываться в сложные разветвленные и объемные молекулярные структуры, составляющие основу всей органической химии, возможность образования только из одного углерода стабильных каркасных молекул все равно неожиданная. В ходе одного из экспериментов группа ученых (1986 г. Гарольд Крото, Роберт Керл, Ричард Смолли) облучала при помощи лазера кусок графита, и датчики обнаружили в испарениях признаки специфических молекулярных структур. Ученые предположили, что это молекулы C₆₀ и C₇₀, и что C₆₀ имеет форму усеченного симметричного икосаэдра (рис.1).

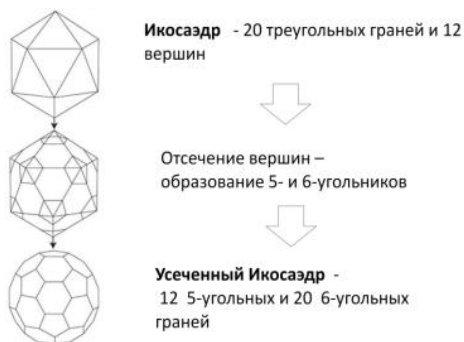


Рис. 1. Образование формы молекулы C₆₀

Фуллерены нашли свое применение в биологии и медицине, производные фуллерена используются при создании вакцин и противовирусных препаратов, ведется изучение перспектив применения фуллерен-кислород-йодного лазера для лечения рака и вирусных инфекций. В области энергетики проводятся исследования по разработке накопителей водорода на основе углеродных наноструктур, в металлургии интерес представляет синтез фуллеренов и их соединений в структуре железоуглеродистых сплавов. Особенно интересны проводимые

исследования в трибологии (научная дисциплина, занимающаяся изучением трения и износа узлов машин и механизмов в присутствии смазочных материалов). В трибологии изучаются трибологические свойства фуллеренов и фуллереносодержащих материалов для применения в качестве присадок к маслам и консистентным смазкам (например, в полимерное связующее вводят многослойную углеродную наноструктуру фуллероидного типа).

2. Экспериментальное открытие возникновения гистерезиса

Эксперимент, на результатах которого основывается данная работа, был проведен корейскими учеными [2] в процессе разработки альтернативного метода нанесения алмазоподобных пленок.

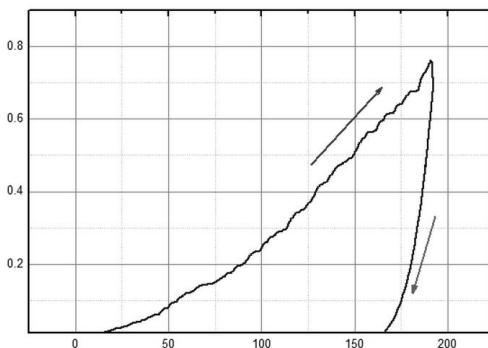
С развитием микроминиатюризации, трибология получила второе дыхание. Существует изрядное количество покрытий, которые могут быть использованы для уменьшения трения и износа — мягкие металлы, органические самоориентирующиеся молекулы, графен и алмазоподобные пленки. Материалов много, но все они имеют те или иные недостатки, и какого-либо универсального пока еще не придумали. Алмазоподобные пленки (Diamond-like coatings, DLC) — наиболее известны. Тем более, что они могут одинаково успешно применяться как на микро- так и на макроуровне.

В эксперименте рассматривался такой способ нанесения DLC, как нанесение ионным пучком, в котором в качестве материала используется не атомарный углерод, а молекулы фуллерена C_{60} . Фуллерен ионизируется, разгоняется до 5 кэВ и взаимодействует с подложкой. При этом молекулы разбиваются, и из обломков формируется новая аморфная структура. Этот метод имеет свои преимущества: пленки влагостойки, возможно нанесение покрытия на предметы произвольной формы, что несколько затруднительно в случае магнетронного распыления. Недостаток пленок — довольно высокий уровень собственных напряжений. Пленка стремится расширяться, занять больший объем, чем у нее есть. Это приводит к неприятным последствиям — если нанести такую пленку на тонкую подложку — подложка может изогнуться. Если подложка будет толще, а адгезия между пленкой и подложкой недостаточно хороша — пленка просто отслоится. Тогда в твердую массу DLC был добавлен фуллерен (поскольку он использовался в качестве основного материала), чтобы компенсировать внутренние напряжения.

Так же в ходе эксперимента было выяснено, что если параллельно с ионным пучком, на подложку направить пучок молекулярный, то в результате получается нано-композит, в котором молекулы фуллерена окружены твердым аморфным углеродом. Напряжения в полученной пленке отсутствовали. Твердость пленки тоже уменьшилась — если для

пленки, нанесенной из ионного пучка характерны значения 50–60 ГПа, то нано-композит продемонстрировал 25–30 ГПа. Но это все равно достаточно много — к примеру, твердость монокристаллического кремния ~ 10 ГПа.

В процессе измерения твердости полученной пленки методом наноиндентирования и была зафиксирована anomальная гистерезисная кривая в результате контролируемого протыкания (рис.2).

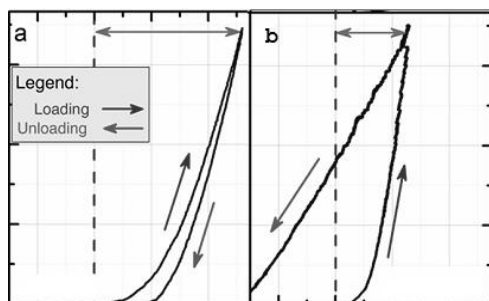


*по оси x – глубина проникновения индентора(в нанометрах),
по оси y – приложенная к индентору сила(в миллиньютонах).*

Стрелка вверх показывает направление нагрузки, стрелка вниз – разгрузки

Рис. 2. График кривой гистерезиса

Данное явление оказалось характерно для нано-композитов, состоящих из смеси молекул фуллерена и твердого аморфного углерода. В зависимости от скорости, с которой производится нагрузка и разгрузка во время эксперимента, кривая меняет свою форму. При высокой скорости (≈ 4 нН/мин) — наблюдается типичная картина для твердой пленки (рис. 3а). При низкой скорости (≈ 0.1 нН/мин) — anomальная картина (рис. 3б).



a) 4нН/мин; b) 0.1нН/мин

Рис. 3. скорость протыкания

Детальный анализ показал, что в случае «аномального» индентирования, вместо отпечатка образуется холм высотой в несколько десятков нанометров (рис.4).

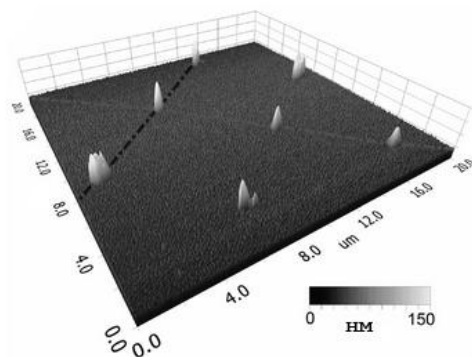


Рис. 4. Отображение образования «холмов» на плоскости

При малой скорости продавливания в пленке возникает дополнительная движущая сила, которая выталкивает индентор обратно. Под нагрузкой материал разбухает, что приводит к выталкиванию индентора и формированию холмов вместо отпечатков за счет того, что при совместном использовании ионного и молекулярного пучков, происходит полимеризация молекул фуллерепа. В обычном состоянии они связаны слабыми связями. В полимеризованном – между двумя соседними молекулами образуется намного более прочная ковалентная связь. Эти два типа связи, помимо прочности, отличаются длиной. Ковалентная связь короче, и полимеризованные молекулы упакованы плотнее. При

«протыкании», полимерные комплексы в месте контакта деформируются, и ковалентные связи разрушаются. В результате, плотно упакованные молекулы стремятся отодвинуться друг от друга, что приводит к увеличению объема, заполнению отпечатка и формированию холма.

3. Математическая модель

Как известно, физические свойства наноразмерных пленок зависят от структуры материала, и основные процессы в нанопокровениях протекают согласно явлениям полимеризации и деполимеризации, которые подчиняются вероятностным законам. Состояние покрытия можно описать совокупностью параметров ω_1, ω_2 , которые характеризуют доли полимеризованного и неполимеризованного фуллерена соответственно.

Динамика данных параметров описывается следующей системой уравнений:

$$\begin{cases} \dot{\omega}_1 = -\lambda_{12} + \lambda_{21}\omega_2, \\ \omega_1 + \omega_2 = 1, \end{cases} \quad (1)$$

с начальными условиями

$$\begin{cases} \omega_1(0) = \omega_{01}, \\ \omega_2(0) = \omega_{02}, \end{cases}$$

где $\lambda_{12}, \lambda_{21}$ – интенсивность перехода из одного состояния в другое

$$\begin{cases} \lambda_{12} = c_1 + c_2 F(t), \\ \lambda_{21} = c_1, \end{cases} \quad (2)$$

где c_1, c_2 – безразмерные произвольные постоянные, $F(t)$ – воздействующая сила следующего вида:

$$F(t) = \begin{cases} at, t \in [0; \frac{T}{2}] \\ -a(t-T), t \in [\frac{T}{2}; T] \end{cases} \quad (3)$$

где T – период.

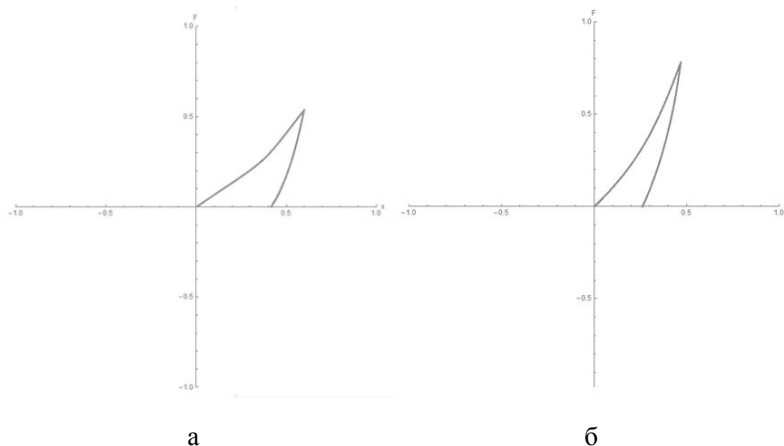
Для описания поведения гистерезиса существует несколько математических моделей. Обзор математических моделей описывающих поведение гистерезиса приведен в [3]. Рассмотрим модель Бука-Вена, которая описывает гистерезис в следующей форме[2]:

$$\begin{cases} \Phi_{bw}(x, t) = \alpha kx(t) + (1 - \alpha)Dkz(z) \\ z = \dot{D}^{-1}(A\dot{x}) - \beta|\dot{x}||z|^{n-1}z - \gamma\dot{x}|z|^n, \end{cases} \quad (4)$$

где \dot{z} – производная по времени, $n \geq 1, D > 0$ – постоянное смещение (коэффициент жесткости, $0 < \alpha < 1$), x – смещение. $\Phi_{bw}(x, t)$ –

суперпозиция закона Гука $\alpha kx(t)$ и гистерезисной компоненты $(1 - \alpha)Dkz(z)$, α, β, γ – безразмерные коэффициенты, определяющие форму и размер гистерезисной петли, n – скалярная величина.

Для каждого из состояний вещества были построены гистерезисные кривые с помощью модели Бука-Вена. На рис. 5а, 5б изображены гистерезисные кривые для обеих долей вещества.



*а – доля полимеризованного фуллерена,
б – доля неполимеризованного фуллерена*

Рис. 5. График гистерезисной кривой

4. Идентификация параметров модели

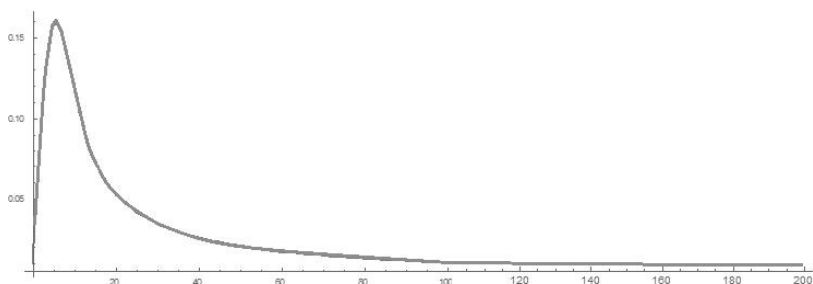
При решении уравнения (1) с учетом выражений (2), (3) было найдено следующее решение, описывающее динамику изменения долей полимеризованного (рис.6) и неполимеризованного фуллерена.

Далее графически представлена разница между гистерезисной кривой, которая является результатом моделирования описанного процесса, и гистерезисной кривой, аномально наблюдаемой в эксперименте (рис.7).

Заключение

В данной работе была рассмотрена модель упругопластического гистерезиса, возникающего в фуллереновых нанопленках, а так же были идентифицированы параметры модели, которые определяют качественное изменение кривой гистерезиса. Были определены параметры c_1, c_2 , от которых зависит интенсивность перехода из полимеризованного состояния в неполимеризованное и обратно. Кроме этого были вычислены и другие

параметры модели Бука-Вена, которые влияют на изменение поведения петли гистерезиса.



$$c_1 = 0.05, c_2 = 0.01, a = 1$$

Рис. 6. График динамики изменения далей вещества

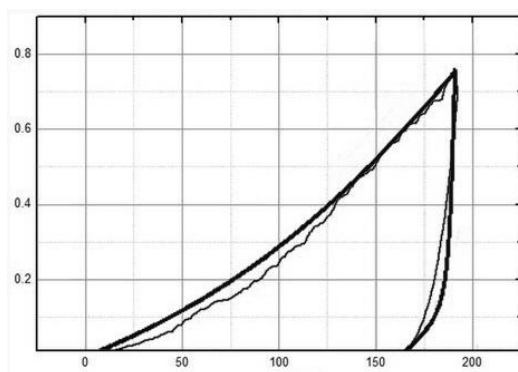


Рис. 7. Сравнение гистерезисных кривых, полученных экспериментально и посредством моделирования.

Список литературы

1. Self-healing phenomenon and dynamic hardness of C60-based nanocomposite coatings / O. V. Penkov, V. E. Pukha, A. Yu. Devizenko [et al.] // Nano Lett. – 2014. – Vol. 14. – P. 2536-2540.
2. Ikhouane, F. Systems with Hysteresis: On the Hysteretic Bouc–Wen Model Part I: Forced Limit Cycle Characterization / F. Ikhouane, T. Rodellar // New York : Springer, 2005. – 582 p.
3. Macki, J. Mathematical models for hysteresis / J. Macki, P. Nistri, P. Zecca // SIAM Review. – 1993. – V. 35(1). – P. 94-123.

ОПРЕДЕЛЕНИЕ АВТОРСТВА ТЕКСТА НА ОСНОВЕ АНАЛИЗА ПАРНОЙ ВСТРЕЧАЕМОСТИ БУКВ

О. В. Желтухина

Студент

Д. И. Соломатин

Старший преподаватель

Введение

Мы живем в информационном обществе, где знания, новые идеи, творчество и вообще интеллектуальная собственность имеют большую ценность, поэтому задача определения авторства текста является актуальной. В данной работе предпринимается попытка исследовать методы и алгоритмы определения авторства текста, проверить возможность использования для этой задачи некоторых характеристик текста, найти оптимальный набор признаков и правила их применения для определения авторства и предоставить модуль, который можно подключить и использовать в других приложениях.

Для того, чтобы эффективно определить авторство текста прежде всего необходимо выбрать уровень, на котором будет проведено исследование.

Методы атрибуции позволяют исследовать текст на пяти уровнях: пунктуационном, орфографическом, синтаксическом, лексико-фразеологическом, стилистическом.

На *пунктуационном* уровне выявляются особенности употребления автором знаков препинания и характерные ошибки.

На *орфографическом* уровне выявляются характерные ошибки в написании слов.

На *синтаксическом* уровне определяются особенности построения предложений, предпочтение тех или иных языковых конструкций, употребление времен, порядок слов и т. д.

На *лексико-фразеологическом* уровне определяются словарный запас автора, особенности использования слов и выражений, склонность к употреблению редких и иностранных слов, диалектизмов, архаизмов, неологизмов, профессионализмов, навыки употребления фразеологизмов, пословиц, поговорок и т. д.

На *стилистическом* уровне определяются жанр, общая структура текста, для литературных произведений – сюжет, характерные изобразительные средства (метафора, ирония, аллегория, гипербола,

сравнение), стилистические фигуры (градация, антитеза, риторический вопрос и т. д.), другие характерные речевые приемы [1].

Очень часто в качестве параметров для определения авторства текста выбираются статистические характеристики, например, среднее число слов в предложении, частота употребления тех или иных частей речи и др. Если такой параметр позволяет различать авторов, имеет достаточную устойчивость и его сложно контролировать сознательно, то такой параметр можно назвать авторским инвариантом.

1. Авторский инвариант

Авторский инвариант – это количественная характеристика литературных текстов или некий параметр, который однозначно характеризует своим поведением произведения одного автора или небольшого числа «близких авторов», и принимает существенно разные значения для произведений разных групп авторов.

Авторским инвариантом может быть не любой параметр, а только тот, который удовлетворяет следующим условиям:

1) *Массовость* – параметр должен опираться на характеристики, которые слабо контролируются автором, чтобы исключить сознательное искажение своего стиля и копирование стиля другого автора.

2) *Устойчивость* – параметр должен сохранять постоянное значение для произведений каждого автора, то есть, иметь небольшое отклонение от среднего значения параметра для этого автора.

3) *Различающая способность* – параметр должен уверенно различать разные группы авторов. В идеале параметр должен иметь абсолютно разные значения для разных авторов, но в реальности достаточно, чтобы колебания параметра для разных авторов превышали колебания для разных текстов одного автора.

Примерами авторских инвариантов могут быть:

- среднее число слов в предложении;
- среднее количество слогов в слове;
- процентное содержание служебных слов (предлогов, союзов, частиц);
- частота употребления существительных, глаголов, прилагательных (в процентах);
- частота употребления предлога «в» (в процентах), частота употребления частицы «не» (в процентах) [2].

Для достижения наиболее точных результатов инварианты можно комбинировать. В результате для обучающей выборки и для контрольного текста будет получен вектор признаков. Для вектора, полученного для контрольного текста, ищется наиболее близкий вектор признаков среди

векторов для обучающей выборки, и чем ближе находятся их значения, тем больше вероятность того, что данный текст принадлежит данному автору.

2. Распознавание автора текста с использованием цепей Маркова

Метод основывается на формальной математической модели последовательности букв текста как реализации цепи Маркова. Для каждого текста из обучающей выборки (то есть, выборки текстов, для которых автор известен) считается матрица переходных частот употреблений пар букв. Такая же матрица строится и для текста, автор которого неизвестен. Далее, на основе этих данных для каждого автора вычисляется вероятность того, что он написал анонимный текст. Тот автор, у которого эта оценка самая высокая, считается автором текста. Данный метод был предложен Д.В. Хмелевым в работе [3].

Для проведения эксперимента был взят набор текстов российских и зарубежных авторов (Акунин, Конан Дойл, Сэлинджер, Тургенев, Уайлд, Грин, Агата Кристи, Гоголь, Роулинг, Достоевский, Фрай): по 6 текстов для каждого автора. Из этого набора последовательно выбиралось по одному тексту, который выступал контрольным (анонимным). По оставшимся текстам проводилось обучение. На все тексты было наложено условие по размеру – не менее 100 000 символов.

Все тексты форматировались следующим образом: все заглавные буквы превращались в прописные; отбрасывались все знаки пунктуации, цифры и символы, не являющиеся буквами русского алфавита (кроме пробела); отбрасывались лишние пробелы (оставлялось ровно по одному пробелу между словами). Все тексты были представлены в формате txt, кодировка UTF-8.

Далее составлялась матрица переходных частот употреблений пар букв: индексами являлись порядковые номера букв русского алфавита минус 1 (так как массивы в Java нумеруются с нуля). Буквы «е» и «ё» считались за одну букву. Индексу 32 соответствовал символ пробела, его сочетание с буквами в тексте тоже учитывалось.

Для обучающей и контрольной выборок были вычислены следующие характеристики:

1) матрица $p_{обуч}$, где $a_{обуч}[i][j]$ – число переходов из символа i в символ j для k -того автора.

2) вектор $p_{обуч}$, где $b_{обуч}[i]$ – количество символов i во всей обучающей выборке для k -того автора.

Для обучающей выборки также была вычислена матрица переходных частот $p_{обуч}$, где

$$p[i][j] = \frac{a[i][j]}{b[i]}, \quad (1)$$

Далее, для каждого из k авторов был вычислен параметр:

$$L[k] = - \sum_i \sum_j a_{\text{контрК}}[i][j] * \ln \left(\frac{a_{\text{контрК}}[i][j]}{p_{\text{обуч}}[i][j] * b_{\text{контрК}}[i]} \right), \quad (2)$$

Таким образом, для каждого из текстов были получены значения L_1, \dots, L_k для каждого из «предполагаемых» авторов. Эти значения были отсортированы по убыванию. Первый автор в списке, т. е. автор, имеющий максимальное значение L для данного текста, полагался истинным автором текста.

После проведения вышеописанного эксперимента были получены следующие результаты: верно авторство определено для 56 из 66 текстов, т. е. точность описанного метода составила около 85% для участвующего в эксперименте набора текстов. Для текстов объемом менее 100 000 символов точность метода заметно снижается.

Заключение

Значения L имеют достаточно высокую различающую способность и устойчивость, эту характеристику автору сложно контролировать. Таким образом, характеристику L можно считать авторским инвариантом.

Список литературы

1. Батура, Т.В. Формальные методы установления авторства текстов и их реализация в программных продуктах / Т. В. Батура // Программные продукты и системы. – 2013. – № 4. – С.286-295.
2. Фоменко, А.Т. Методы. Дополнение 3. Авторский инвариант русских литературных текстов [Электронный ресурс] / А.Т. Фоменко. – Режим доступа : <http://www.chronologia.org/xpon2/dop3.html>
3. Хмелёв, Д. В. Распознавание автора текста с использованием цепей А.А. Маркова / Д. В. Хмелёв // Вестник Московского университета. Сер.9, Филология. – 2000. – № 2. – С. 115-126.
4. Mystem-morphtagger. Gate plugin for morphological analysis of Russian language [Электронный ресурс]: документация. – Режим доступа : <http://code.google.com/p/mystem-morphtagger>

РАСПОЗНАВАНИЕ МИШЕНИ, ИЗОБРАЖЕННОЙ НА ЧАСТИЧНО ЗАКРЫТОМ АЛЬБОМНОМ ЛИСТЕ

М. В. Землянухин

Студент

Д. И. Соломатин

Старший преподаватель

Введение

В настоящее время одним из динамично развивающихся направлений ИТ является робототехника. Робот – такое устройство, которое по данным датчиков способно делать выводы о состоянии окружающей среды и осуществлять действия с помощью манипуляторов. Взаимодействие с окружающей средой довольно сложно в реализации, поэтому возможности современных роботов все еще очень далеки от возможностей человека. Однако простые роботы могут помочь человеку в работе.

Одним из важнейших датчиков робота является видеочамера. Как известно, человек получает около 80% информации об окружающем мире с помощью зрения. Научив робота распознавать с помощью чамеры конкретные объекты, можно существенно расширить его возможности и границы применения. Проблемами распознавания объектов занимается раздел науки «Компьютерное зрение».

В настоящее время проблемам компьютерного зрения уделяется все большее внимание. Несколько примеров автоматизации задач, использующих компьютерное зрение: автоматическое чтение автомобильных номеров, отслеживание перемещений спортсменов в игровых видах спорта для предоставления обширной статистики для спортивных передач, продвинутые контроллеры для компьютерных игр и т. д. которые, например, реагируют на движения рук и ног.

В данной работе решается учебная задача о проверки наличия мишени (см. рис. 1, 2) в поле зрения робота, результаты решения которой можно в будущем использовать и в практических целях (например, человек сможет показывать роботу карточки, которые будут интерпретированы им как команды).

Задача усложняется допуском того, что часть границы альбомного листа закрыта другим объектом (например, пальцем человека, держащего лист).

Разделение поставленной задачи на подзадачи:

- поиск альбомного листа в кадре и его нормализация;

- анализ содержимого листа на предмет наличия мишени.

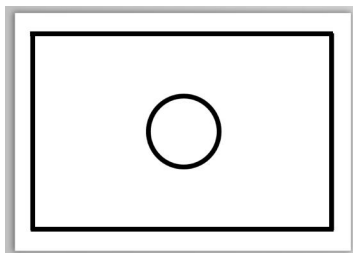


Рис. 1. Мишень на альбомном листе



Рис. 2. Поле зрения робота

Преимущество выбранного в работе подхода в том, что, нормализовав альбомный лист, в будущем возможно обнаруживать на нем не только какую-то определенную жестко заданную мишень, но и считать с него изображение произвольной формы, например, кривую, которую робот воспримет как команду на движение по заданному пути.

Алгоритм нахождения альбомного листа содержит большое количество шагов, осуществляемых последовательно. При этом каждый шаг работает с данными, возвращенными предыдущим шагом, а первый шаг получает на вход исходное изображение.

1. Нахождение альбомного листа – часть, использующая библиотеку OpenCV

Шаги алгоритма, использующие библиотеку OpenCV:

- сглаживание шума (см. рис. 3). Для сглаживания шума применяется алгоритм локального усреднения (Local averaging) цвета с размером сглаживаемой области для каждой точки 3×3 .

- приведение изображения с 3 каналами цвета к изображению с одним каналом цвета (см. рис. 4). Выбранный алгоритм поиска перепада цветов (Canny edge detector) работает только с одной компонентой цвета. Поэтому для его применения нужно преобразовать исходные данные. Для этого сначала RGB-изображение преобразуется в HLS-изображение, и затем в HLS-изображения берется компонента L (Light).

- применение детектора границ Кэнни (Canny edge detector) (см. рис. 5). Детектор границ Кэнни преобразует исходное изображение с одной компонентой цвета в бинарное изображение, на котором белым цветом отмечены перепады цвета (границы цветов). Детектор принимает на вход ряд параметров, которые были подобраны так, чтобы оставить как можно больше границ и не отфильтровать линии, которые являются границами альбомного листа.

– выделение и аппроксимация контуров (см. рис. 6). Границы освещенности, показывающие точки перепадов света, могут быть представлены в виде контуров и затем аппроксимированы. Основным параметром метода аппроксимации является максимально допустимое расстояние между любой из точек, преобразующихся в отрезок, до самого отрезка.



Рис. 3. Сглаживание шума



Рис. 4. Переход к изображению с одним каналом цвета

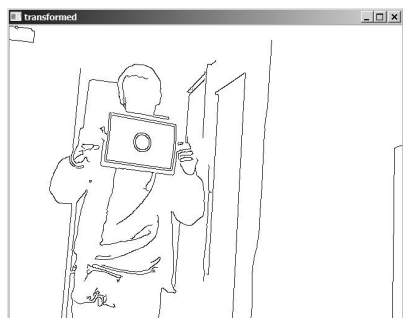


Рис. 5. Применение детектора границ Кэнни (Canny edge detector)

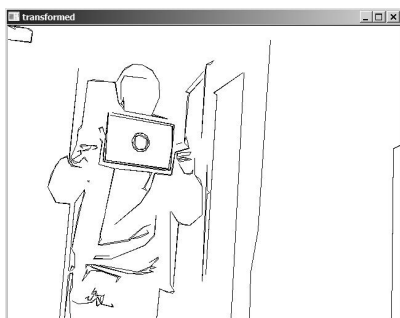


Рис. 6. Выделение и аппроксимация контуров

2. Нахождение альбомного листа – геометрическая часть

После последовательного применения нескольких алгоритмов, реализованных в библиотеке OpenCV, получен набор аппроксимированных отрезков, расположенных в местах значительного изменения компоненты Light пространства цветов HLS исходного изображения. Если альбомный лист присутствует на изображении, каждая из его черных границ будет представлена двумя отрезками (переход с

белого на черный и с черного на белый). Далее задача сводится к тому, чтобы из большого количества отрезков выделить те, которые потенциально могут быть альбомными листами, т. е. похожи на прямоугольник с заранее известным соотношением длин сторон.

Шаги алгоритма, использующие геометрические приемы:

- фильтрация отрезков малой длины (см. рис. 7).
- в условии рассматриваемой задачи задается максимальное расстояние, на котором может быть распознан альбомный лист, поэтому, лист не может быть размером меньше определенного. На этом шаге из рассмотрения удаляется большое количество отрезков. Все отрезки короче определенной (рассчитанной) длины удаляются из списка отрезков.
- фильтрация отрезков, близко расположенных друг к другу (см. рис. 8). Черная линия на белом фоне дает две линии перехода цвета – с черного на белый и с белого на черный. Нет смысла рассматривать эти линии отдельно, поэтому нужно оставить одну из них.

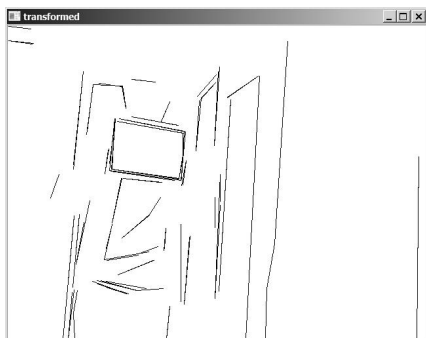


Рис. 7. Фильтрация отрезков малой длины

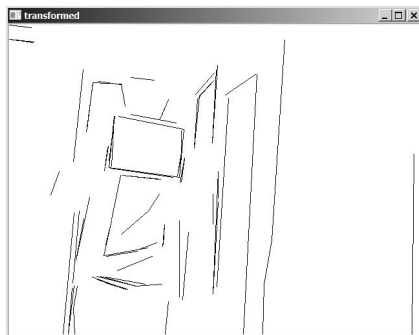


Рис. 8. Фильтрация отрезков, близко расположенных друг к другу

– получение пар примерно параллельных линий (см. рис. 9). Метод просматривает все возможные пары отрезков. Те пары отрезков, у которых углы относительно оси OX не различаются больше, чем на входной параметр, возвращаются для последующей обработки.

– получение возможных прямоугольников (см. рис. 10). Метод просматривает все возможные пары пар отрезков, полученных на предыдущем шаге, и возвращает список возможных прямоугольников. Проверяется несколько условий:

- угол между прямыми должен быть близок к 90 (максимальное отличие от 90 задается параметром).

– отношение длины прямоугольника к ширине. Это отношение для альбомного листа примерно 1.41. Максимальное отличие от 1.41 задается параметром. Здесь происходит выбор между тем, чтобы отфильтровать как можно больше ложных прямоугольников, и тем, чтобы распознать прямоугольник в условиях большого влияния перспективы (например, когда мы смотрим на лист сверху вниз, его ширина намного меньше длины).

– минимальная площадь. Минимальное значение заранее получено, при расположении альбомного листа под максимальным углом на максимально допустимом расстоянии от камеры.

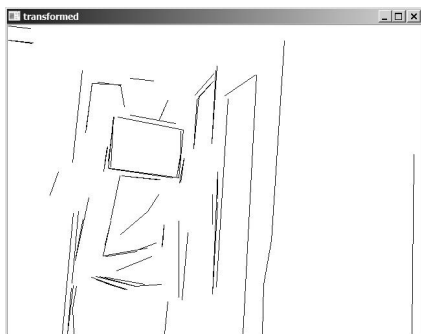


Рис. 9. Получение пар примерно параллельных линий

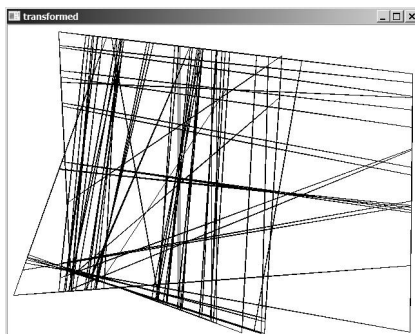


Рис. 10. Получение возможных прямоугольников

– фильтрация по сравнению периметра с суммарной длиной совпадения (см. рис. 11). Для каждого из прямоугольников находятся две величины: периметр прямоугольника и сумму длин совпадений исходных отрезков с отрезками прямоугольника. Далее находим отношение второй величины к первой (оно находится в диапазоне 0.0-1.0). Если оно больше какого-то значения (например, 0.8), прямоугольник проходит фильтрацию. Чем меньше значение коэффициента, тем большую часть окаймляющей линии альбомного листа допускается закрыть (например, рукой), но тем больше возможно ложных откликов.

– фильтрация внешних прямоугольников (см. рис. 12). На этом шаге мы проверяем вложенность и оставляем только внутренние прямоугольники.

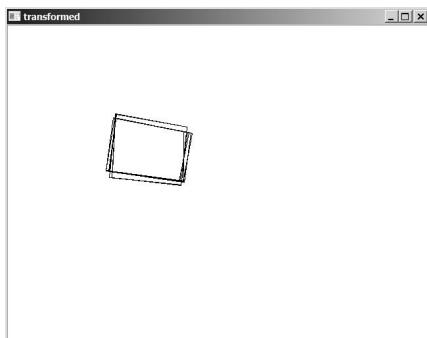


Рис. 11. Фильтрация по сравнению периметра с суммарной длиной совпадения

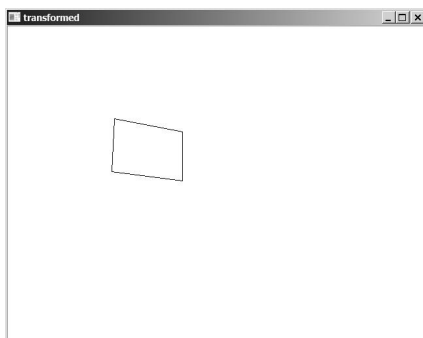


Рис. 12. Фильтрация внешних прямоугольников

– Нормализация найденных прямоугольников (см. рис. 13). На этом шаге найденные прямоугольники приводятся к горизонтальной ориентации и заданному размеру.



Рис. 13. Нормализация найденных прямоугольников



Рис. 14. Нахождение мишени

3. Поиск мишени на нормализованном альбомном листе

Мишень представляет собой круг, расположенный в центре листа. Для ее поиска используется преобразование Хафа (см. рис. 14).

Преобразование Хафа применяется к бинарному изображению. Как правило, бинарное изображение, подаваемое на вход алгоритму, получается применением к исходному изображению какого-либо детектора краев. Преобразование Хафа трансформирует исходное бинарное изображение так, что каждой точке ставится в соответствие возможность существования особенности изображения (такой, как линия, отрезок, круг или любой другой).

Библиотека OpenCV содержит алгоритм, который, используя преобразование Хафа, позволяет найти круги на изображении. Несмотря на то, что алгоритм способен находить только круги идеальной формы, его

применение в данном случае возможно. Альбомный лист нормализован, поэтому влияние перспективы устранено.

4. Результаты эксперимента

Было записано видео, состоящее из 550 кадров, в котором альбомный лист с мишенью и без мишени располагается на разном расстоянии и под разными углами от камеры. Для оценки успешности алгоритма использовалась классификация из источника [2].

– TP — the number of true positives – процент случаев, когда образец был корректно обнаружен и действительно являются искомым объектом – 94.2%.

– FP — the number of false positives – количество случаев, когда образец был обнаружен, но в действительности не является искомым объектом – 0%.

– TN — the number of true negatives – количество случаев, когда алгоритм не обнаружил искомого объекта в кадре, и кадр действительно не содержал искомого объекта – 100%.

– FN — the number of false negatives – количество случаев, когда алгоритм не обнаружил искомого объекта в кадре, а объект там был – 5.8%.

Среднее время работы алгоритма на процессоре Pentium Dual-Core CPU 2.5 GHz – 0.086 секунды, максимальное время – 0.243 секунды.

Список литературы

1. Bradski, G. Learning OpenCV / G. Bradski, A. Kaebler. – Sebastopol : Gravenstein Highway North : O'Reilly, 2008. – 557 p.
2. Dawson-Howe, K. A practical introduction to computer vision with OpenCV / K. Dawson-Howe. – United Kingdom : Wiley, 2014 – 235 p.
3. OpenCV community help wiki [Электронный ресурс] : сайт сообщества. – Режим доступа : <http://docs.opencv.org>

РАСПОЗНАВАНИЕ ЛИЦ

Д. А. Зяблов

Студент

В. В. Фертиков

Доцент

Введение

Принято считать, что задача компьютерного зрения – это задача логического вывода: имеются некоторые измерения и модель, а нужно определить, что дало такие измерения. Следует, однако, заметить, что компьютерное зрение отличается от многих иных задач логического вывода. Во-первых, имеется чрезвычайно много данных; во-вторых, неизвестно, что из имеющейся информации пригодится, а что – нет. Например, одна из основных трудностей в создании хороших программ распознавании объектов состоит в определении того, какие пиксели распознавать, а какие – игнорировать [1]. Задача идентификации и распознавания лиц – это одна из первых практических задач, которая стимулировала становление и развитие теории распознавания и идентификации объектов. Интерес к процедурам, лежащим в основе процесса узнавания и распознавания лиц, всегда был значительным, особенно в связи с возрастающими практическими потребностями. В настоящее время системы, основанные на автоматической проверки личности, внедрены или внедряются во все сферы жизни: камеры, установленные на улицах, в зданиях, помогают правоохранительным органам отслеживать преступников, вовремя фиксировать беспорядки; платежные системы, основанные на идентификации пользователя по средствам распознавания лица позволяют расплатиться за купленный товар или услугу, просто подойдя к терминалу и нажав единственную кнопку, после того как система узнает вас; проходя по улице, можно проверить любого прохожего в социальных сетях и на сайтах знакомств, узнать любую доступную о нем информацию.

На сегодняшний день существует множество систем распознавания лиц, основанные на различных подходах. Они постоянно дорабатываются, алгоритмы совершенствуются. Однако, большинство из них не доступны обычным пользователям, они либо засекречены, либо являются платными. А системы в свободном доступе дают не удовлетворительный результат. Таким образом, разработка общедоступной системы распознавания лиц является одной из актуальных проблем на сегодняшний день, для решения которой возможно применение нескольких алгоритмов.

1. Цели и задачи

Целью работы является разработка общедоступной системы распознавания лиц.

При этом решаются следующие задачи:

- изучение существующих подходов к распознаванию лиц;
- рассмотрение возможности выбора существующих программных средств – библиотек компьютерного зрения;
- глубокое изучение выбранного метода;
- определение языка разработки системы.

Разработанная системы должна обеспечивать:

- детектирование лица и его распознавание;
- работу в режиме реального времени;
- возможность обучения существующей базы данных лиц;
- простоту использования.

На основе вышеизложенного сформирован следующий список этапов разработки:

- установка выбранной среды разработки;
- установка выбранной библиотеки компьютерного зрения;
- разработка системы детектирование лиц;
- разработка системы распознавания лиц;
- тестирования системы.

2. Библиотеки компьютерного зрения

OpenCV – Библиотека компьютерного зрения с открытым исходным кодом (Open Source Computer Vision Library), содержащая более 500 функций, предназначенных для выполнения в реальном времени.

Библиотека содержит алгоритмы для обработки, реконструкции и очистки изображений, распознавания образов, захвата видео, слежения за объектами, калибровки камер и др.

Изначально перед OpenCV ставились следующие цели:

- исследование машинного зрения, разработка и оптимизация кода;
- распространение информации по предметной области, разработка общей инфраструктуры на которой могли бы основываться разработчики, код должен быть читаем и свободно передаваем другим программистам;
- приложения должны быть переносимыми, оптимизированными, код которых не обязательно должен быть открытым. Приложения также могут создаваться в коммерческих целях.

Открытая лицензия для OpenCV была составлена таким образом, чтобы было возможно создавать коммерческие приложения, используя

любые возможности OpenCV. Разработчики не обязаны делать свой проект открытым или публиковать изменения, внесённые ими в библиотеку. Отчасти из-за таких либеральных условий существует большое сообщество пользователей, включающее в себя такие крупные компании как IBM, Microsoft, Intel, Sony, Siemens, Google и это далеко не полный список, а также научно-исследовательские центры, такие как Стэнфорд, Массачусетский технологический институт, CMU, Кембридж и INRIA. OpenCV популярна во всём мире, причём большие сообщества пользователей можно найти в Китае, Японии, России, Европе и Израиле.

С момента своего альфа-релиза в январе 1999 года, OpenCV была использована во многих приложениях и научно-исследовательских работах, в том числе: наложение обычных карт и фотографий со спутника, выравнивание документов при сканировании, удаление шума из медицинских изображений, анализ объектов, системы безопасности, автоматическое наблюдение, системы контроля качества на производстве, калибровка камер, а также беспилотные летательные, наземные и подводные аппараты. Она даже была использована для распознавания звука и музыки, где методы распознавания образов были применены к изображениям спектрограмм звука. Библиотека стала ключевой частью системы зрения робота «Stanley» из Стэнфорда, выигравшего 2.000.000\$ на Больших Пустынных Гонках Роботов DARPA [2].

3. Предварительная обработка изображения

Во всех задачах компьютерного зрения перед началом выполнения основных задач изображения проходят предварительную обработку.

Все методы предварительной обработки изображений можно разделить на пространственные и частотные.

Пространственная область. Термин относится к плоскости изображения как таковой, и данная категория объединяет подходы, основанные на прямом манипулировании пикселями изображения.

Частотная область. Термин относится к модификации спектра, формируемого путем применения к изображению преобразования Фурье.

Смешанные подходы. Рассматриваются алгоритмы, базирующиеся на различных комбинациях методов, использующих частотную и временную область.

На первом этапе предварительной обработки происходит формирование изображения. Формированием изображения называется процедура непосредственного получения изображения в виде расположенного в памяти видеопроцессора массива дискретных элементов – пикселей, образующих матрицу или контур.

Часто применяемые методы предварительной обработки изображения:

- градиентное преобразование;
- выравнивание гистограммы;
- линейная фильтрация: арифметический фильтр, фильтр гаусса и др.;
- нелинейная фильтрация, например, медианный фильтр, который прекрасно борется с импульсными помехами

4. Изучение существующих подходов к распознаванию лиц

Существует большое разнообразие алгоритмов распознавания лиц, однако, можно выделить общую структуру процесса распознавания (рис. 1).

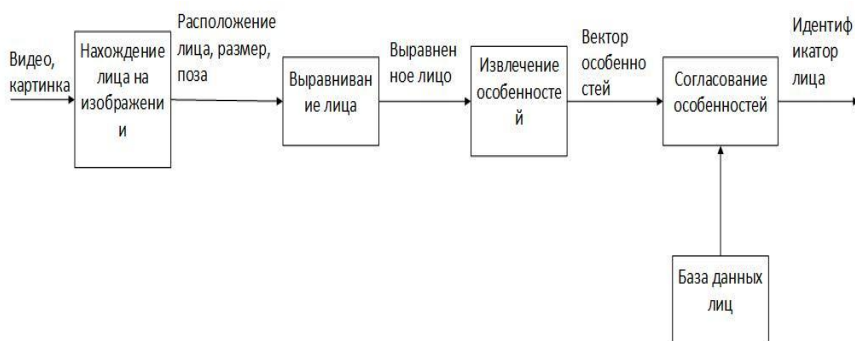


Рис. 1. Общая структура процесса распознавания

На первом этапе производится детектирование и локализация лица на изображении. На этапе распознавания производится выравнивание изображения лица (геометрическое и яркостное), вычисление признаков и непосредственно распознавание – сравнение вычисленных признаков с заложенными в базу данных эталонами. Основным отличием всех представленных алгоритмов будет вычисление признаков и сравнение их совокупностей между собой [3-4].

Среди множества существующих подходов к распознаванию наиболее известными являются:

- метод гибкого сравнения на графах;
- метод главных компонент;
- скрытые Марковские модели;
- нейронные сети;
- активные модели внешнего вида;
- активные модели формы.

Основные проблемы, связанные с разработкой систем распознавания лиц:

- проблема освещенности;
- проблема положения головы (лицо – это 3D объект).

5. PCA Eigenface

Проанализировав существующие подходы к распознаванию лиц, было принято решение о выборе алгоритма PCA Eigenface. Этот алгоритм является хорошо изученным, несмотря на свою сравнительную простоту, он показывает достойные результаты.

Первоначально метод главных компонент начал применяться в статистике для снижения пространства признаков без существенной потери информации. В задаче распознавания лиц его применяют главным образом для представления изображения лица вектором малой размерности (главных компонент), который сравнивается затем с эталонными векторами, заложенными в базу данных.

Главной целью метода главных компонент является значительное уменьшение размерности пространства признаков таким образом, чтобы оно как можно лучше описывало «типичные» образы, принадлежащие множеству лиц. Используя этот метод можно выявить различные изменчивости в обучающей выборке изображений лиц и описать эту изменчивость в базисе нескольких ортогональных векторов, которые называются собственными (eigenface).

Полученный один раз на обучающей выборке изображений лиц набор собственных векторов используется для кодирования всех остальных изображений лиц, которые представляются взвешенной комбинацией этих собственных векторов. Используя ограниченное количество собственных векторов можно получить сжатую аппроксимацию входному изображению лица, которую затем можно хранить в базе данных в виде вектора коэффициентов, служащего одновременно ключом поиска в базе данных лиц [5-6].

Заключение

В ходе работы была реализована система распознавания лиц, способная в режиме реального времени детектировать и идентифицировать лица. Так же в разработанной системе имеется возможность внесения лиц в базу данных – обучение системы

Список литературы

1. Форсайт, Д. Компьютерное зрение. Современный подход / Д. Форсайт, Ж. Понс; пер. с англ. – Москва : Издательский дом «Вильямс», 2004. – 928 с.

2. Гайко, П. OpenCV – описание библиотеки компьютерного зрения [Электронный ресурс] / П. Гайко. – Режим доступа: <http://prog-master.com/opencv-opisanie>
3. Шапиро, Л. Компьютерное зрение / Л. Шапиро, Дж. Стокман; пер. с англ. – Москва : БИНОМ; Лаборатория знаний, 2006. – 752 с.
4. Гонсалес, Р. Цифровая обработка изображений / Р. Гонсалес, Р. Вудс – Москва : Техносфера, 2005. – 1072 с.
5. Прэтт, У. Цифровая обработка изображений / У. Прэтт; пер. с англ. – Москва : Мир, 1982. – 782 с.
6. Гонсалес, Р. Цифровая обработка изображений / Р. Гонсалес, Р. Вудс. – 3-е изд., перераб. и доп. – Москва : Техносфера, 2012. – 1104 с.

МЕТОДЫ АНАЛИЗА ОТКРЫТЫХ ДАННЫХ ПОЛЬЗОВАТЕЛЕЙ СОЦИАЛЬНЫХ СЕТЕЙ

Ю. Ю. Иванников

Студент

Е. Ю. Митрофанова

Ассистент

Введение

Анализ социальных данных стремительно набирает популярность во всём мире благодаря появлению в 1990-х годах онлайн-сервисов социальных сетей (SixDegrees, LiveJournal, Facebook, Twitter, YouTube и другие). С этим связан феномен социализации персональных данных: стали публично доступными факты биографии, переписка, дневники, фото-, видео-, аудиоматериалы, заметки о путешествиях и т. д. Таким образом, социальные сети являются уникальным источником данных о личной жизни и интересах реальных людей. Это открывает беспрецедентные возможности для решения исследовательских и бизнес-задач (многие из которых до этого невозможно было решать эффективно из-за недостатка данных), а также создания вспомогательных сервисов и приложений для пользователей социальных сетей. Кроме того, этим обуславливается повышенный интерес к сбору и анализу социальных данных со стороны компаний и исследовательских центров [1].

1. Данные из социальных сетей

По данным ВЦИОМ на 2014 год, 82 процента пользователей Сети зарегистрированы в тех или иных социальных сетях.

На сегодняшний день, практически все социальные сети (включая упомянутые выше) предоставляют API для взаимодействия с внешними приложениями. Кроме того, регистрируясь в социальной сети,

пользователь принимает условия пользовательского соглашения, по которому разрешает открытый доступ к информации о себе в рамках законодательства [2]. Это означает, что, имея инструменты для работы с социальными сетями, можно получать определенные поведенческие биометрические образцы личности непосредственно из упомянутых источников.

Таким образом, оценивая поведение пользователя и его социальную активность, можно с определенной точностью сделать предположение о факте соответствия пользователя той или иной личности на основе поведенческих признаков, доступ к которым возможен в социальных сетях, таких, как:

- геолокация, геотеги;
- связи между пользователями;
- место учебы, факультет, курс, год окончания;
- место жительства, место рождения;
- место работы;
- ссылки на веб-ресурсы и другие социальные сети;
- активность по отношению к другим пользователям (количество записей, комментариев и т. н. «лайков»);
- общая активность, среднее время, проведенное на веб-ресурсе, наиболее частое время посещения и т. д.

Помимо всего вышеупомянутого, отдельный интерес представляет изменение поведенческих показателей пользователя с течением времени: изменение места жительства, смена интересов, статистика посещаемости и прочее.

2. Обзор методов анализа данных

Цель данной работы – исследовать возможность идентификации личности, основываясь на поведенческих данных, полученных из социальных сетей. Кроме того, необходимо дать оценку существующим методам анализа данных из социальных сетей.

К основным практическим задачам, решаемым в рамках анализа данных из социальных сетей можно отнести:

- идентификация пользователя;
- определение демографической принадлежности пользователя;
- классификация пользователей и поиск сообществ;
- измерение информационного влияния события;
- выявление трендов в обществе.

Задача построения системы анализа данных из социальных сетей подразумевает под собой обработку больших объемов данных о пользователях, их нормализацию и последующий анализ.

В общем случае методы анализа данных имеют следующую классификацию [3]:

1. корреляционный анализ – совокупность методов, позволяющих обнаружить зависимость между несколькими случайными величинами;

2. регрессионный анализ заключается в определении аналитической формы связи, в которой изменение результативного признака обусловлено влиянием одного или нескольких факторных признаков, а множество всех прочих факторов, также оказывающих влияние на результативный признак, принимается за постоянные и средние значения;

3. факторный анализ – многомерный метод, применяемый для изучения взаимосвязей между значениями переменных;

4. дисперсионный анализ – метод в математической статистике, направленный на поиск зависимостей в экспериментальных данных путём исследования значимости различий в средних значениях.

3. Пример решения задачи поведенческой идентификации

В рамках данной работы был спланирован и проведен эксперимент с использованием специально разработанного программного комплекса. В качестве источника данных выступала социальная сеть VK.com. В ходе эксперимента использовалась выборка, содержащая данные около 1000 реальных пользователей социальной сети VK.com за 10 дней. При этом такие характеристики пользователя, как ФИО и год рождения не использовались, а для наглядности каждому пользователю был присвоен целочисленный идентификатор. Элемент выборки представлял собой следующий вектор параметров:

- пол;
- место жительства;
- среднее количество добавленных записей к себе на стену в день;
- среднее время, проведенное онлайн в день;
- среднее время онлайн, сгруппированное по часам (представлено в виде 24 характеристик, каждая из которых соответствует одному часу из суток).

Оценка точности распознавания пользователя из данной выборки производилась следующим образом:

1. из множества пользователей в выборке один пользователь выбирался случайным образом;

2. генерировались тестовые векторы данных на основе данных о выбранном пользователе за 5, 2 и 1 день;

3. тестовые векторы подавались на вход системы в виде JSON-файла;

4. для каждого вектора сохранялись результаты работы системы в виде JSON-файла.

В качестве метода идентификации использовался метод наименьших квадратов.

В ходе эксперимента были получены результаты, позволяющие дать оценку точности работы системы в целом и метода наименьших квадратов в частности. Данные представлены в графическом виде на рис. 1-рис. 4 (здесь каждая диаграмма представляет собой результаты тестирования системы для входных векторов признаков, взятых у случайных пользователей, усредненных для каждого из них на определенное время; на оси X находятся деления с количеством пользователей в используемой выборке, а ось Y представляет собой процент правильного распознавания).



Рис. 1. Процент правильного распознавания вектора признаков, представляющего собой усредненные данные о пользователе за 1 сутки



Рис. 2. Процент правильного распознавания вектора признаков, представляющего собой усредненные данные о пользователе за 3 суток



Рис. 3. Процент правильного распознавания вектора признаков, представляющего собой усредненные данные о пользователе за 5 суток



Рис. 4. Процент правильного распознавания вектора признаков, представляющего собой усредненные данные о пользователе за 10 суток

Заключение

Данная статья представляет собой обзор методов анализа пользовательских данных из социальных сетей, а также эксперимент по поведенческой идентификации личности. Целью данного исследования было проверка гипотезы о возможности использования данных из социальных сетей для практических нужд. Результаты исследования являются подтверждением гипотезы, и точность, полученная в результате эксперимента достаточно высока (94% для наилучших условий и 43% для наихудших).

Список литературы

1. Анализ социальных сетей : методы и приложения [Электронный ресурс] / Коршунов А., Белобродов И., Бунун Н., [и др.] // Труды ИСП

РАН. – 2014. – №1. – [Режим доступа] : <http://cyberleninka.ru/article/n/analiz-sotsialnyh-setey-metody-i-prilozheniya>.

2. Шаров, В. Биометрические методы компьютерной безопасности / В. Шаров // ВУТЕ Россия. – 2005. – №4. – С. 19-23.

3. Freeman, L. Research methods in social network analysis / L. Freeman, D. White, A. Romney. – New York : Wiley, 1992. – 478 p.

РЕАЛИЗАЦИЯ ФОНОВОГО СЕРВИСА С ВОЗМОЖНОСТЬЮ ОБРАБОТКИ СИСТЕМНЫХ СОБЫТИЙ ДЛЯ МОБИЛЬНЫХ УСТРОЙСТВ НА ПЛАТФОРМЕ ANDROID

М. С. Ильин

Студент

П. С. Лысачев

Старший преподаватель

Введение

В современном мире достаточно широко распространено использование коротких текстовых сообщений всевозможными онлайн-сервисами. Текстовые сообщения могут использоваться для различных целей, в том числе и для рекламы, но в первую очередь служат для идентификации личности пользователя. Например, текстовые сообщения применяются для подтверждения платежных транзакций или заказа товаров и услуг.

Данный способ является достаточно удобным и универсальным средством для осуществления подобных операций, однако преподносит ряд неудобств. Поскольку большинство из перечисленных действий, чаще всего, совершаются через персональный компьютер, пользователь может не иметь доступа к мобильному телефону, который, например, находится в другой комнате, в сумке или разряжен. Данную проблему поможет решить сервис, позволяющий переадресовывать некоторые уведомления со смартфона на персональный компьютер, такие как входящие текстовые сообщения или информацию о скорой разрядке аккумуляторной батареи. Кроме того, данный сервис поможет не пропустить важный звонок, если на смартфоне активирован беззвучный режим, сообщив пользователю, работающему за персональным компьютером о данном событии.

1. Цель работы

Одной из основных задач при разработке подобного сервиса уведомлений является реализация подсистемы обработки событий

операционной системы для их дальнейшей передачи на персональный компьютер. В данной статье рассматривается способ реализации подсистемы обработки системных событий и фонового сервиса, позволяющего координировать работу данной подсистемы для смартфонов на базе операционной системы Android.

1. Способы обработки системных событий в Android

Системное API операционной системы Android предоставляет некоторый набор инструментов для обработки различных системных событий. Самым простым способом, позволяющим осуществить обработку, является использование абстрактного класса `BroadcastReceiver`. При определении наследников данного класса необходимо реализовать абстрактный метод `onReceive`, который принимает два параметра. Первый из них, имеющий тип `Context`, является объектом, представляющим контекст в рамках которого был запущен обработчик широковещательных событий. Второй параметр имеет тип `Intent` [1]. Одно из предназначений объектов данного типа в Android – осуществление единообразной передачи дополнительной информации между различными объектами. В зависимости от обрабатываемого события, данный объект будет содержать необходимую информацию, например, входящее текстовое сообщение и дополнительные данные об отправителе. Пользовательская обработка события происходит в данном методе (`onReceive`). Например, здесь можно обработать и отправить необходимую информацию на персональный компьютер, используя сетевое или Bluetooth соединение.

Для обеспечения базовой функциональности сервиса уведомлений необходим перехват следующих событий:

- входящее текстовое сообщение;
- изменение уровня заряда аккумулятора.

Соответственно, необходима реализация двух классов-наследников класса `BroadcastReceiver` с соответствующей обработкой полученных данных.

Кроме того, для добавления мобильному приложению дополнительной функциональности (запуск приложения при старте ОС, активация сервиса при подключении к домашнему Wi-Fi) нужно реализовать обработчики загрузки ОС и изменения состояния сетевого подключения.

Для получения соответствующих данных в обработчиках используются полученные объекты типа `Intent`. Информацию по работе с данными объектами при обработке конкретных событий можно найти в официальной документации [2]. В данной статье не рассматривается реализация конкретных обработчиков, поскольку, это не является ее основной целью.

2. Обработка входящих звонков

Для обработки входящих звонков, существует более гибкий способ, чем использование наследников класса `BroadcastReceiver`. В библиотеке операционной системы Android поставляется класс `PhoneStateListener`, который предназначен для отслеживания различных событий жизненного цикла сотового телефона, одним из которых и является голосовой вызов. Один из методов, предоставляемых данным классом является метод `onCallStateChanged`, который в качестве параметров принимает идентификатор текущего состояния и номер абонента с которым установлена связь. Получаемых данных достаточно для того, чтобы предоставить пользователю ПК необходимую информацию о входящем голосовом вызове, способами, аналогичными с рассмотренными ранее наследниками класса `BroadcastHandler`.

3. Конфигурация приложения

Система безопасности операционной системы Android требует указания разрешений, необходимых приложению для работы. Разрешения задаются строковыми константами и указываются в файле основной конфигурации Android-приложения – `AndroidManifest.xml`. Пример указания разрешения на доступ к просмотру состояния телефона приведен в листинге 1.

Листинг 1

Разрешение на чтение состояния телефона

```
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
```

Помимо указанного, для работы реализованных обработчиков необходимо указать разрешения для обработки событий загрузки устройства, приема и чтения текстовых сообщений и доступа к состоянию сетевого подключения смартфона.

Кроме указания разрешений, в файле конфигурации необходимо указать применяемые широковещательные обработчики. В момент их объявления можно указать фильтры событий, которые необходимо обрабатывать. Однако, данное действие может быть совершено и в момент регистрации обработчика из программного кода. Приведем пример объявления обработчика загрузки устройства (листинг 2). Поскольку он должен работать до запуска основного приложения, фильтр событий необходимо настроить именно в файле конфигурации.

После выполнения данной конфигурации для запуска необходимых обработчиков достаточно зарегистрировать их в системе с указанием необходимых фильтров, если это не было сделано при конфигурации.

Регистрация обработчика загрузки смартфона

```

<receiver android:name=".BootReceiver"
android:enabled="true">
  <intent-filter>
    <action
android:name="android.intent.action.BOOT_COMPLETED" />
  </intent-filter>
</receiver>

```

4. Фоновые сервисы в Android

Операционная система Android предоставляет готовый набор классов для реализации системных сервисов, работающих в фоновом режиме. В рамках рассматриваемого приложения подобный системный сервис используется для централизованного управления работой обработчиков системных событий. Рассмотрим способ реализации подобного сервиса.

Для определения собственного системного сервиса необходимо реализовать наследника абстрактного класса Service, предоставляемого платформой Android. При реализации данного класса нужно переопределить методы onCreate, onStartCommand и onDestroy. Данные методы будут вызываться при создании, старте и уничтожении системного сервиса соответственно.

Поскольку операционная система Android управляет распределением памяти динамически, при ее нехватке запущенный системный сервис может быть остановлен и уничтожен. Однако имеется возможность обойти данную проблему. При старте сервиса (метод onStartCommand) необходимо вызвать метод startForeground расширяемого класса Service. Поскольку средства безопасности Android не позволяют запускать подобные сервисы без уведомления пользователя, в данный метод передаются два параметра: уникальный идентификатор сервиса и экземпляр класса Notification, который содержит информацию об активности сервиса для отображения в области системных уведомлений.

5. Итоговая подсистема обработки событий

На рисунке изображена итоговая диаграмма классов разработанной подсистемы обработки системных событий. Как видно из данной диаграммы, обработчики системных событий создаются в экземпляре класса NotifierService и сосуществуют с ним. Экземпляры классов-обработчиков должны создаваться вместе с сервисом и регистрироваться в системе при старте фонового сервиса. Кроме того, обязательно нужно отменять регистрацию используемых обработчиков при остановке сервиса. Используемый сервис может обладать несколько большей

функциональностью, необходимой приложению, например, управлять подсистемой работы с сетью.

Данный способ позволяет гибким образом управлять всей подсистемой обработчиков. Для управления ее жизненным циклом необходимо только лишь запускать или останавливать основной фоновый сервис.

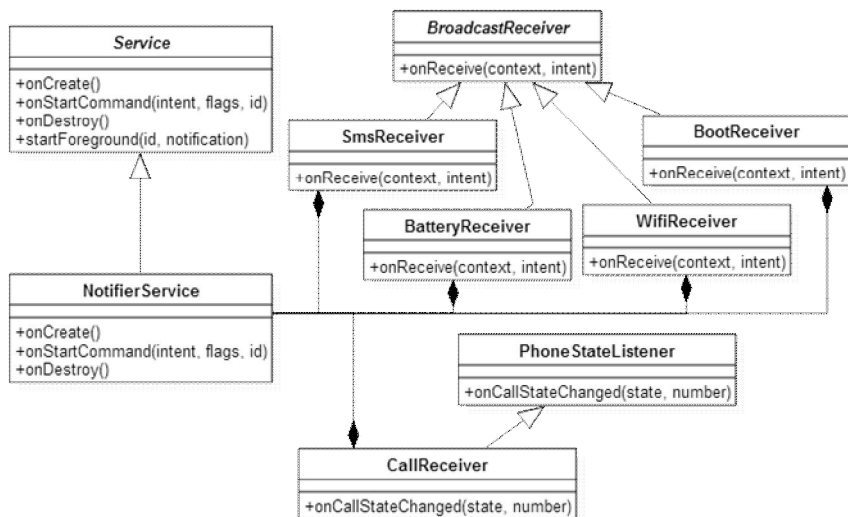


Рисунок. Диаграмма классов разработанной подсистемы

Для управления работой фоновго сервиса используются методы `startService` и `stopService`. Данные методы доступны в любом объекте, имеющем доступ к контексту выполнения Android. Как для регистрации широковещательных обработчиков, так и для ее отмены используются методы `registerReceiver` и `unregisterReceiver`, первый из которых в качестве дополнительного параметра может принимать фильтры системных событий, о которых говорилось ранее.

Для регистрации обработчика состояния телефона используется несколько другой подход с использованием системного сервиса `TelephonyManager`.

Более подробную информацию о работе с данными методами и инструментами можно узнать в приведенных источниках [2] и [3].

Заключение

В рамках данной статьи была рассмотрена одна из возможных реализаций подсистемы обработки различных системных событий для

смартфонов на базе операционной системы Android с централизованным управлением и работой в фоновом режиме.

Рассмотренный подход является достаточно универсальным и может применяться как для описанного сервиса удаленных уведомлений для персонального компьютера, так и для реализации других приложений, работающих с системными уведомлениями.

Список литературы

1. Eckel, B. Thinking in Java / B. Eckel. – New Jersey: Prentice Hall, 2006. – 1057 с.
2. Introduction to Android [Электронный ресурс] : Android Developers. – Режим доступа : <http://developer.android.com/guide/index.html>
3. Meier, R. Professional Android Application Development / R. Meier. – Indianapolis : Wiley Publishing Inc., 2009. – 409 p.

РАСПОЗНАВАНИЕ ОБРАЗОВ МЕТОДОМ ВИОЛЫ-ДЖОНСА С ИСПОЛЬЗОВАНИЕМ ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛЕНИЙ НА ОСНОВЕ ТЕХНОЛОГИИ CUDA

Г. И. Казун

Студент

А. А. Сирота

Профессор

Введение

Современные технологии распознавания образов решают сложные задачи, предоставляя удобный инструмент для обнаружения необходимых объектов. При обработке изображения достаточно большого размера его анализ может занять значительное время.

Многие задачи, выполняемые на персональных компьютерах, связаны с быстродействием обработки информации. Существуют технологии, позволяющие с помощью GPU ускорить процесс вычислений.

1. Технология CUDA

CUDA — архитектура параллельных вычислений, которая позволяет значительно увеличить вычислительную производительность систем благодаря использованию графических процессоров фирмы NVidia.

CUDA Toolkit позволяет пользователям разработать на специальном языке программирования, похожем на C, алгоритмы, осуществляющие обработку данных на GPU NVIDIA, а также предоставляет инструменты

для организации доступа к графическому процессору и управления его памятью по своему усмотрению.

GPU — отдельное устройство в персональном компьютере, ноутбуке, мобильном устройстве или игровой приставке, выполняющее вычисления, предназначенные для обработки и вывода графической информации. С помощью специально разработанной параллельной архитектуры они в разы превосходят стандартные центральные процессоры в обработке графической информации.

2. Метод Виолы-Джонса

Метод был разработан и представлен в 2001 году Полом Виолой и Майклом Джонсом.

Основные принципы, на которых основан метод:

- изображение в интегральном представлении;
- признаки Хаара;
- каскады признаков;
- сканирующее окно.

В методе Виолы-Джонса сканирующее окно перемещается следующим образом (рис. 1): окно принимает минимальный размер, далее оно с определённым шагом передвигается по горизонтальной линии изображения.

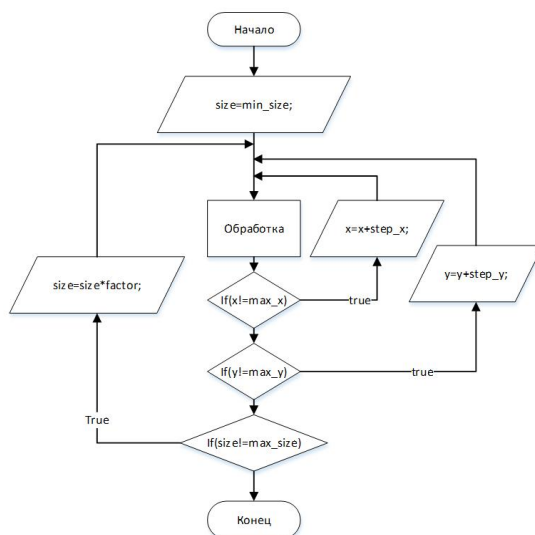


Рис. 1. Структурная схема алгоритма сканирующего окна стандартного метода

Достигнув границы изображения, окно передвигается на шаг по вертикали и проходит следующую горизонтальную линию. Этот цикл будет повторяться до тех пор, пока не будет достигнута граница изображения по вертикали, после чего сканирующее окно увеличится с помощью заданного коэффициента и будет заново проходить по изображению. Его увеличение будет происходить до тех пор, пока размер не станет равен размеру изображения.

3. Цель работы

В данной статье рассматривается задача анализа и реализации метода распознавания образов Виолы-Джонса с помощью параллельных вычислений на графическом процессоре и проводится анализ ее эффективности.

Для решения поставленных задач на графическом процессоре использовалась машина со следующими техническими характеристиками:

- Материнская плата: ASUS Z87M-PLUS
- Процессор: Intel Core i7 4770K 3.50GHz
- Видеокарта: NVIDIA GeForce GTX 750 2Gb
- ОЗУ: 1 x DDR3 4Gb

4. Анализ и реализация известных результатов в области обнаружения объектов методом Виолы-Джонса

Был проанализирован и реализован алгоритм, основанный на версии метода Виолы-Джонса с использованием библиотеки OpenCV.

Далее, используя технологию CUDA [1, 2], можно привести получившийся алгоритм к следующему виду (рис. 2). Алгоритм распознавания разбивается на блоки, каждому из которых отводится уникальный размер распознающего окна. Каждый блок итеративно проходит изображение по вертикали и с помощью параллельных нитей анализирует каждое окно, расположенное на горизонтальной линии.

Так же на основе статьи [3] можно реализовать алгоритм (рис. 3) в котором размер сканирующего окна увеличивается итеративно, а на обработку каждой линии по вертикали отведён параллельный блок, в котором нити одновременно сканируют элементы по горизонтали.

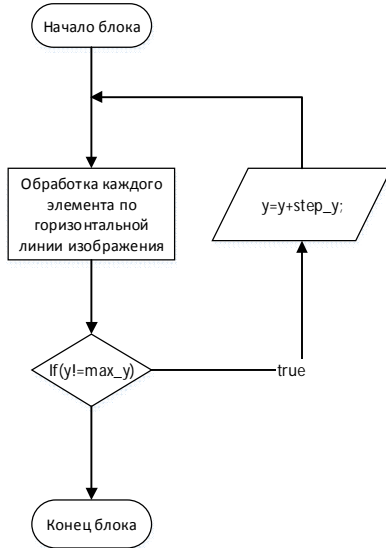


Рис. 2. Структурная схема алгоритма сканирующего окна, основанного на параллельной обработке размеров окна и строк

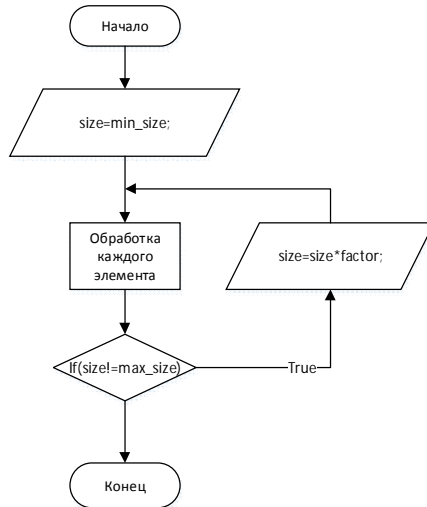


Рис. 3. Структурная схема алгоритма сканирующего окна, основанного на параллельной обработке каждого элемента

5. Анализ полученных методов с различными входными данными

GPU без использования параллельных вычислений не является эффективным средством для ускорения алгоритма, так как результаты однопоточного алгоритма распознавания методом Виолы-Джонса в разы хуже, чем тот же алгоритм, выполняемый на CPU. Для анализа изображения размером 1024 x 1024 графическому процессору потребовалось 57 секунд, в то время как на центральных процессорах Intel различных семейств было затрачено 3.9 секунды.

При обработке изображения размером 1024 x 1024 пикселей было обнаружено 89 различных областей с искомыми объектами. Сканирование на CPU Intel различных семейств заняло в среднем 3.9 секунды, в то время как методу №1, основанному на параллельной обработке размеров окна и строк, потребовалось 3.2 секунды, а методу №2, основанному на параллельной обработке каждого элемента, 1.8 секунды. Преимущество метода №2 в данном тесте заключается в меньшем количестве итеративных циклов прохождения изображения сверху вниз, чем в методе №1.

Таблица 1

Показатели затраченного времени

| Тип процессора | Время, с |
|-----------------------|-----------------|
| 2Duo E7200 | 4,9 |
| I3 3240 | 4,1 |
| I5 2450m | 3,7 |
| I7 4770 | 2,8 |
| GTX 750 №1 | 3,2 |
| GTX 750 №2 | 1,8 |

При обработке одинаковых изображений, размерами 1024 x 1024 пикселей и 2048 x 2048, пикселей был обнаружен 351 объект.

Таблица 2

Показатели затраченного времени

| Тип процессора | Время, затраченное на обработку изображения 1024 x 1024, с | Время, затраченное на обработку изображения 2048 x 2048, с |
|-----------------------|---|---|
| 2Duo E7200 | 22,4 | 22,1 |
| I3 3240 | 16,5 | 16,9 |
| I5 2450m | 13,8 | 14,2 |
| I7 4770 | 9,6 | 9,5 |
| GTX 750 №1 | 6,8 | 6,7 |
| GTX 750 №2 | 2,6 | 3,8 |

Из представленных результатов можно сделать вывод, что размер изображения почти не влияет на затраченное время в алгоритме, реализованном на CPU и в методе №1. В методе №2 есть разница во времени, так как в нём присутствует цикличное прохождение при увеличении окна, а в методе №1 в данном случае создаётся параллельный блок.

При обработке изображения размерам 4096 x 4096 пикселей было обнаружено 1372 объекта.

Таблица 3

Показатели затраченного времени

| Тип процессора | Время, с |
|-----------------------|-----------------|
| 2Duo E7200 | 4,9 |
| I3 3240 | 4,1 |
| I5 2450m | 3,7 |
| I7 4770 | 2,8 |
| GTX 750 №1 | 3,2 |
| GTX 750 №2 | 1,8 |

Из результатов следует, что при значительном увеличении объектов на изображении метод №2, выполняемый за 24.5 секунды уступает методу №1 в быстродействии, который выполнялся почти в 2 раза быстрее чем №1 и в 4 раза быстрее по сравнению со средним значением затраченного времени на CPU.

Заключение

В рамках данной работы были реализованы алгоритмы метода распознавания образов Виолы-Джонса с помощью параллельных вычислений и проанализированы результаты с использованием различных входных данных.

С помощью параллельных вычислений, основанных на технологии CUDA, можно добиться значительного прироста в быстродействии задач, нестандартного характера для видеокарт. Так, например, скорость распознавания образов методом Виолы-Джонса можно сократить в 3-5 раз в зависимости от входных данных. Что касается метода, основанном на параллельной обработке сканирующих окон и элементов по горизонтали, по сравнению с методом, обрабатывающим каждый элемент параллельно, можно отметить, что эффективность первого появляется только при обработке крупных изображений, так как он не обрабатывает каждое окно независимо, а сконцентрирован на увеличении сканирующего окна с использованием параллельных вычислений, которое зависит от размеров входного изображения.

Список литературы

1. Боресков, А. В. Параллельные вычисления на GPU. Архитектура и программная модель CUDA : Учебное пособие / А. В. Боресков. – Москва : Издательство Московского университета, 2012. – 336 с.
2. Боресков, А. В. Основы работы с технологией CUDA / А. В. Боресков, А. А. Харламов. – Москва : ДМК-Пресс, 2010. – 232 с.
3. Акимов, А. В. Разработка и исследование алгоритмов распознавания изображений на основе метода Виолы-Джонса с использованием технологии вычислений на графических процессорах CUDA / А. В. Акимов, А. А. Сирота // Вестник Воронеж. гос. ун-та. Сер. Системный анализ и информационные технологии. – 2014. – № 3. – С. 100-109.

РАЗРАБОТКА МОБИЛЬНОГО ПРИЛОЖЕНИЯ НА ПЛАТФОРМЕ 1С:ПРЕДПРИЯТИЕ 8.3

Е. А. Копытина
Студент
А. И. Другалев
Ассистент

Введение

В современном обществе с каждым годом ускоряется течение времени. Особенно в мегаполисах, люди постоянно в движении, постоянно в работе и нам так необходимо быть всегда на связи, знать обо всем, что происходит в мире, чтобы принимать правильные решения. Принятие же несвоевременных и ошибочных решений приводит к дополнительным расходам и убыткам.

В наших руках всегда находится мобильный телефон, благодаря которому люди больше не привязаны к рабочему месту и даже к определенной точке земного шара. Мобильные приложения значительно облегчают жизнь людей и позволяют работать с необходимой предметной областью.

Платформа «1С: Предприятие 8» предоставляет огромные возможности для разработки бизнес-приложений и фактически является индустриальным стандартом автоматизации управления и учета в России [1].

1. Постановка задачи

Для повышения эффективности работы предприятия нужно улучшить взаимодействие между сотрудниками и руководителем, поэтому возникает необходимость в создании инструмента, выполняющего данную задачу.

Цель работы: разработка приложения «Обмен сообщениями в 1С: Предприятие 8.3».

К разрабатываемому приложению выдвинуты следующие требования:

- возможность работы приложения на мобильной платформе Android;
- возможность просмотра истории сообщений;
- отправка сообщения сразу нескольким пользователям;
- отсутствие необходимости перенастройки конфигурации перед началом работы с приложением.

2. Используемые объекты конфигурации

В качестве основы взята типовая конфигурация «Управление небольшой фирмой».

Для реализации задачи потребуется объект для хранения списка пользователей. В качестве такого объекта будет использован объект конфигурации Справочник, предназначенный для работы со списками данных.

Также будет использован Регистр сведений, предназначенный для описания структуры хранения данных в разрезе нескольких измерений. На основе объекта Регистр сведений платформа создает в базе данных таблицу, в которой может храниться произвольная информация, «привязанная» к набору измерений. Важной особенностью этого объекта конфигурации является его способность хранить данные с привязкой ко времени – периодический Регистр сведений.

Для переопределения стандартного поведения системы во время записи объектов служит объект конфигурации Обработка.

И наконец, будет использован объект конфигурации Общая форма. Формы в 1С: Предприятие 8 предназначены для отображения и редактирования информации, содержащейся в базе данных. Формы, не принадлежащие конкретным объектам прикладного решения, называются общими формами [2].

3. Логика приложения

Реализуемое приложение должно работать следующим образом: пользователь на форме отправки сообщения имеет возможность выбора одного либо нескольких получателей сообщения. При нажатии на кнопку выбора нескольких получателей, открывается форма выбора получателей, где пользователь может просмотреть список получателей, список активных

в данный момент времени получателей, осуществить выбор всех получателей либо отменить свой выбор. Сделав свой выбор, пользователь нажимает на кнопку ок – открывается форма отправки сообщения, затем пользователь набирает сообщение и нажимает на кнопку отправить.

4. Реализация интерфейса

Справочник «Пользователи» имеет следующий вид (рис. 1).

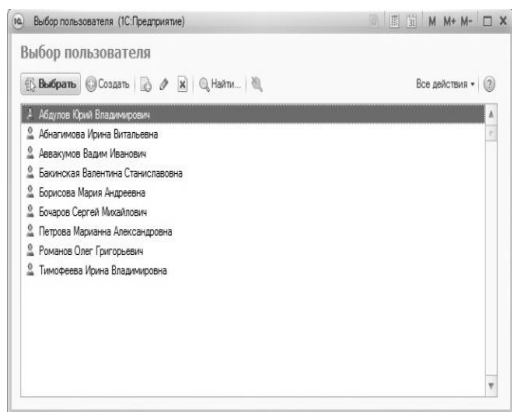


Рис. 1. Справочник «Пользователи»

На форме Обработки «НаписатьСообщение» (рис. 2) происходит выбор одного пользователя из Справочника «Пользователи» либо по кнопке осуществляется переход на Общую форму «ФормаВыбораПолучателей», где выбирается несколько получателей.

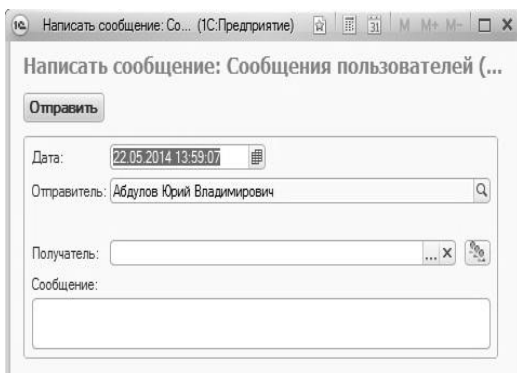


Рис. 2. Форма Обработки «НаписатьСообщение»

Общая форма «ФормаВыбораПолучателей» представлена на рис. 3.

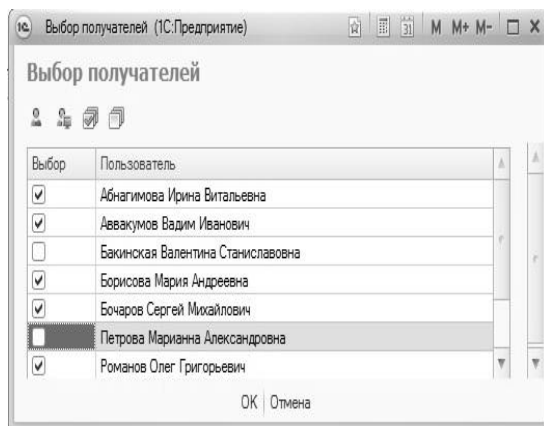


Рис. 3. «ФормаВыбораПолучателей». Все пользователи

Возможность просмотра и выбора активных пользователей показана на рис. 4.

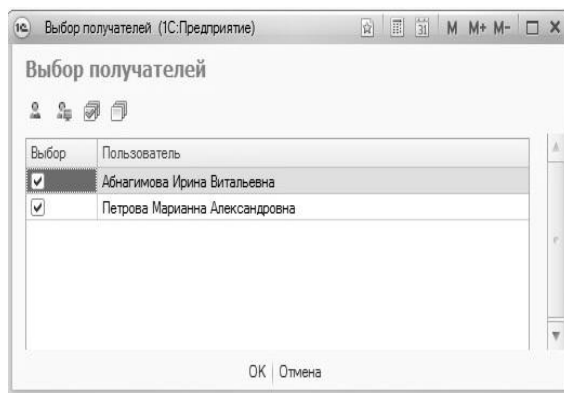


Рис. 4. «ФормаВыбораПолучателей». Активные пользователи

При нажатии на кнопку ОК происходит переход на форму Обработки «НаписатьСообщение» и в реквизит Получатели заносятся выбранные получатели из формы «ФормаВыбораПолучателей» (рис. 5).

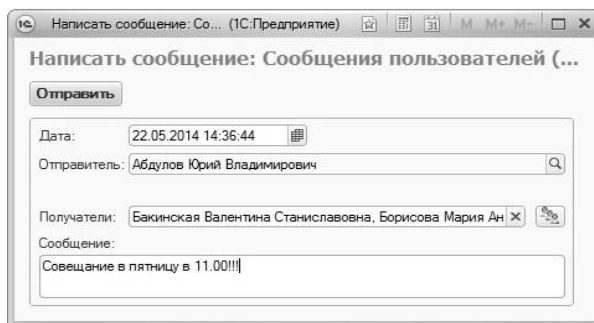


Рис. 5. «ФормаВыбораПолучателей». С заполненными реквизитами

При нажатии на кнопку отправить все данные из обработчика попадают в Регистр сведений, который занимается рассылкой сообщений.

Общая форма «ФормаСообщения» представлена на рис. 6. Таким образом, пользователь просматривает новое сообщение и отвечает на него.

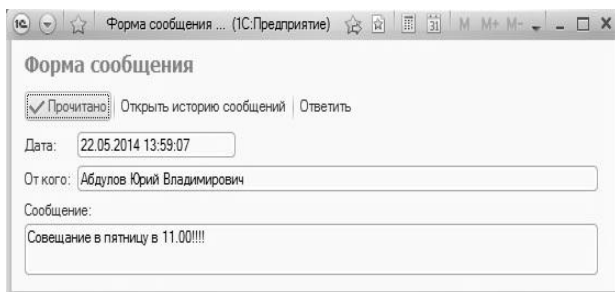


Рис. 6. Общая форма «ФормаСообщения»

Имеется возможность просмотра истории сообщений (рис.7).

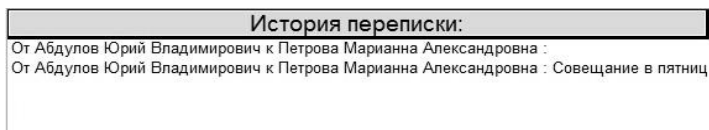


Рис. 7. История сообщений

Заключение

Данная статья посвящена разработке приложения «Обмен сообщениями в 1С:Предприятие 8.3». Приложение улучшает взаимодействие между сотрудниками и руководителем, что способствует повышению эффективности работы предприятия.

Модуль позволяет:

- отправлять сообщения не только одному, но и нескольким пользователям;
- просматривать историю сообщений;
- работать с приложением на мобильной платформе Android.

Перед началом работы с приложением нет необходимости в перенастройке конфигурации.

Список литературы

1. Ажеронок, В. А. Профессиональная разработка в системе 1С:Предприятие 8 / В. А. Ажеронок, А. П. Габеев, Д. И. Гончаров. – Москва : 1С-Паблишинг, 2014. – 1400 с.
2. Радченко, М. Г. 1С : Предприятие 8.2. Практическое пособие разработчика. Примеры и типовые приемы / М. Г. Радченко, Е. Ю. Хрусталева. – Москва : 1С-Паблишинг, 2009. – 874 с.

РАЗРАБОТКА КОМПЬЮТЕРНОГО НЕЙРОИНТЕРФЕЙСА НА ОСНОВЕ РЕГИСТРАЦИИ УСТОЙЧИВЫХ ЗРИТЕЛЬНЫХ ВЫЗВАННЫХ ПОТЕНЦИАЛОВ

В. В. Коротков

Студент

А. А. Вахтин

Доцент

Введение

Человеческое существование можно описать как процесс взаимодействия с объектами окружающего мира. Под взаимодействием обычно подразумевается непосредственный физический контакт, требующий приложения мышечных усилий. Однако существует ряд заболеваний, которые приводят к ограничению физических способностей человека и невозможности взаимодействия с устройствами, призванными облегчить его жизнь, традиционными методами.

Именно поэтому в настоящее время набирают всё большую популярность исследования, направленные на устранение промежуточного этапа взаимодействия путём создания интерфейсов управления компьютером на основе сигнала, снимаемого непосредственно с человеческого мозга. Такие интерфейсы называются компьютерными нейроинтерфейсами. В качестве сигнала наиболее часто используется

сигнал электроэнцефалографа (ЭЭГ), который измеряет разность потенциалов в различных точках скальпа.

1. Устойчивые зрительные вызванные потенциалы

Устойчивые зрительные вызванные потенциалы (Steady State Visually Evoked Potentials – SSVEP) представляют собой сигналы, являющиеся естественными откликами на визуальную стимуляцию на специфических частотах. Когда сетчатка возбуждается визуальным стимулом, мерцающим на постоянной частоте в промежутке между 3.5 и 75 герц, мозг генерирует электрическую активность на той же самой (или кратной ей) частоте стимуляции [1].

Таким образом, возможно построение интерфейса, содержащего несколько стимулов, мерцающих на разных частотах и обозначающих различные команды, которые человек может отдать компьютеру. Тогда задача сводится к определению стимула, на который человек смотрит в данный момент, путём детектирования в сигнале ЭЭГ вынужденного потенциала на одной из частот стимуляции.

SSVEP наиболее хорошо идентифицируемы в промежутке 5-25 Гц. Одним из главных достоинств данной технологии является неплохое соотношение сигнал/шум.

2. Особенности стимуляции с экрана монитора

Целью работы являлась разработка компьютерного нейроинтерфейса, ориентированного на обычные персональные компьютеры, а также практическая реализация и применение одного из новейших алгоритмов детектирования SSVEP, основанного на индексе синхронизации. Данный метод превосходит основные существующие на данный момент алгоритмы по точности распознавания [2]. Кроме того, требовалось связать разработанное приложение с имевшимся серверным программным обеспечением, считывающим сигнал ЭЭГ.

При стимулировании с экрана монитора необходимо решить две проблемы: определить необходимые технические параметры монитора и программный способ отображения индикаторов. Параметры монитора не должны быть слишком ограниченными, чтобы нейроинтерфейс был пригоден для использования наибольшим числом владельцев обычных компьютеров. Монитор компьютера отображает изображение покадрово. Каждый кадр отрисовывается с определённой фиксированной частотой. Самая распространённая на данный момент частота обновления монитора – 60 Гц, хотя существуют модели с частотами 75 Гц, 85 Гц и выше.

Стимулирование может производиться мерцающими двухцветными индикаторами. Для корректного детектирования частоты SSVEP, стимулирующий его сигнал должен быть максимально стабилен, поэтому

количество частот, которые можно использовать для стимуляции с экрана монитора, ограничено (таблица). Полный период индикатора должен укладываться в целое число кадров монитора. Так как SSVEP наиболее корректно идентифицируются в интервале 5-20 Гц, то были получены 7 частот, пригодных для отображения с экрана монитора [3].

Таблица

Частоты, пригодные для отображения на типовом мониторе (60 Гц)

| Кол-во кадров | Период, мс | Частота, Гц | Симулирующий сигнал |
|---------------|------------|-------------|---------------------|
| 3 | 50.00 | 20.00 | 011 |
| 4 | 66.67 | 15.00 | 0011 |
| 5 | 83.33 | 12.00 | 00111 |
| 6 | 100.00 | 10.00 | 000111 |
| 7 | 116.67 | 8.57 | 0001111 |
| 8 | 133.33 | 7.50 | 00001111 |
| 9 | 150.00 | 6.66 | 000011111 |

При этом совместное использование 10 Гц и 20 Гц, а также 7.5 Гц и 15 Гц может привести к ошибкам детектирования из-за совпадения фундаментальных частот колебания одних стимулов со вторыми гармониками других.

3. Индекс многомерной синхронизации

Для реализации детектирования SSVEP был выбран алгоритм MSI, основанный на нахождении индекса синхронизации между сигналом ЭЭГ и эталоном. Колебательные процессы называются синхронизированными, если соответствие их фаз не меняется с течением времени [4].

Сигнал электроэнцефалографа представляется в виде матрицы размером $N \times M$:

$$X = \begin{pmatrix} X_1^1 & \dots & X_M^1 \\ \vdots & \ddots & \vdots \\ X_1^N & \dots & X_M^N \end{pmatrix}, \quad (1)$$

где X_i^k – отсчёт k -того канала в i -тый момент времени, N – количество каналов, M – количество отсчётов.

Для каждой из распознаваемых частот f_i строится матрица эталонного сигнала размером $2N_h \times M$:

$$Y_{f_i} = \begin{pmatrix} \sin(2\pi \cdot f_i \cdot t) \\ \sin(2\pi \cdot f_i \cdot t) \\ \vdots \\ \sin(2\pi \cdot N_h \cdot f_i \cdot t) \\ \cos(2\pi \cdot N_h \cdot f_i \cdot t) \end{pmatrix}, t = \frac{1}{F_s}, \frac{2}{F_s}, \dots, \frac{M}{F_s}, \quad (2)$$

где F_s – частота дискретизации, N_h – количество гармоник синусоидальных и косинусоидальных компонент.

Задача сводится к нахождению метрики, характеризующей схожесть регистрируемого электроэнцефалографом сигнала и эталонных сигналов. Такой метрикой в рамках алгоритма выступает индекс синхронизации.

Предполагается, что сигналы (строки матриц) нормализованы и имеют нулевое математическое ожидание и единичную дисперсию. Для нахождения индекса синхронизации необходимо объединить строки матриц и найти матрицу ковариации (также корреляции по Пирсону, т. к. сигналы нормализованы), которая может быть представлена в виде [5]:

$$C = \begin{bmatrix} C11 & C12 \\ C21 & C22 \end{bmatrix}, \quad (3)$$

$$C11 = \frac{1}{M} XX^T, \quad C12 = \frac{1}{M} XY^T, \quad C21 = \frac{1}{M} YX^T, \quad C22 = \frac{1}{M} YY^T. \quad (4)$$

Чтобы уменьшить вклад в итоговую меру корреляций между каналами ЭЭГ и между компонентами эталонного сигнала, применяется следующее линейное преобразование [5]:

$$U = \begin{bmatrix} C11^{-1/2} & 0 \\ 0 & C22^{-1/2} \end{bmatrix}, \quad (5)$$

$$R = UCU^T = \begin{bmatrix} I_{N \times N} & C11^{-1/2} C12 C22^{-1/2} \\ C22^{-1/2} C21 C11^{-1/2} & I_{2N_h \times 2N_h} \end{bmatrix}, \quad (6)$$

где $I_{N \times N}$ – единичная матрица размером $N \times N$, $I_{2N_h \times 2N_h}$ – единичная матрица размером $2N_h \times 2N_h$.

Находятся нормализованные собственные значения матрицы R :

$$\lambda'_i = \frac{\lambda_i}{\sum_{i=1}^P \lambda_i} = \frac{\lambda_i}{tr(R)} = \frac{\lambda_i}{P}, \quad (7)$$

где $\lambda_1, \lambda_2, \dots, \lambda_P$ – собственные значения, $P = N + 2N_h$ – размерность матрицы R .

Индекс синхронизации рассчитывается по формуле [5]:

$$S = 1 + \frac{\sum_{i=1}^P \lambda_i' \log(\lambda_i')}{\log(P)} \quad (8)$$

Найдя индексы S_1, S_2, \dots, S_K для каждого из K эталонных сигналов, стимулируемая частота может быть определена по правилу максимума:

$$T = \max_i S_i, i = 1, 2, \dots, K. \quad (9)$$

4. Детали реализации

Для реализации приложения был выбран язык C++ и библиотека QT. Кроме того, использовалась библиотека линейной алгебры Eigen и средства OpenGL для отрисовки стимулов и вертикальной синхронизации кадров.

Приложение было интегрировано с уже имевшимся программным обеспечением по считыванию сигнала ЭЭГ и его пересылке по сети. Схема полученной системы представлена на рис. 1. Информация об электрической активности мозга считывается электроэнцефалографом, разбивается на кадры и пересылается приложению.

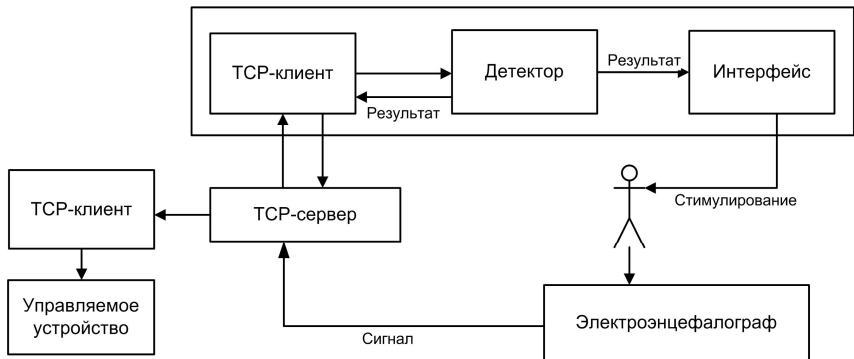


Рис. 1. Схема системы

В приложении выделены три отдельных модуля, отвечающих за различную функциональность:

- интерфейс, отображающий мерцающие с определёнными частотами стимулы;
- TCP-клиент, который принимает данные электроэнцефалографа, обёрнутые в кадры, и пересылает их детектору;
- детектор, который накапливает достаточное количество данных, обрабатывает их и принимает решение о детектировании стимулируемой частоты.

Результат детектирования визуализируется интерфейсом, а также отправляется с помощью TCP-клиента через сервер на прочие устройства, установившие связь с этим сервером. Таким образом, при помощи нейроинтерфейса возможно организовать управление любым устройством, имеющим доступ в сеть.

Диаграмма классов приложения изображена на рис. 2. Набор детектируемых частот и соответствующих им команд управляется классом-контейнером FrequencySet.

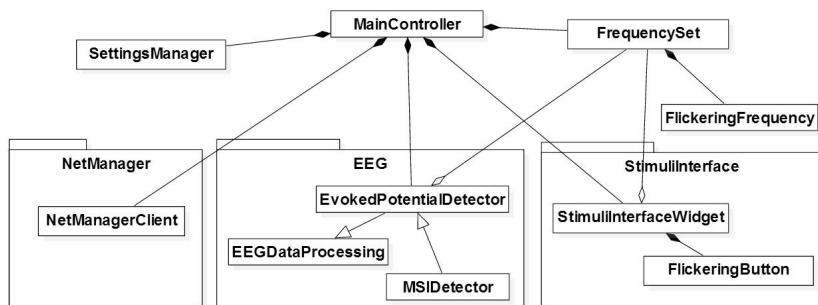


Рис. 2. Диаграмма классов приложения

Сетевое взаимодействие с сервером осуществляется классом NetManagerClient, отображение интерфейса мерцающих стимулов – классом StimuliInterfaceWidget. Непосредственно обработка сигнала реализуется методами классов модуля EEG. Класс EEGDataProcessing представляет собой базовый класс обработчика, который ответственен за приём и накопление данных. Класс EvokedPotentialDetector реализует стадии предварительной обработки и детектирования, оставляя реализацию детектирования на потомков. MSIDetector – класс, использующий алгоритм MSI для детектирования вызванных потенциалов. При этом вычисление индексов синхронизации производится параллельно с помощью средств QtConcurrent. Сохранение и загрузка настроек детектора осуществляется методами класса SettingsManager. Кроме того, на диаграмме опущены классы-виджеты, представляющие пользовательский интерфейс для настройки и управления перечисленными классами, включающими бизнес-логику приложения.

Заключение

Данная статья посвящена разработке компьютерного нейроинтерфейса на основе регистрации устойчивых зрительных вызванных потенциалов мозга. Был применён алгоритм детектирования частоты стимуляции, использующий индекс синхронизации для

определения зависимости между сигналом ЭЭГ и эталонными сигналами. В итоге была получена система, которая позволяет отдавать команды любому устройству, имеющему доступ к серверу, посредством зрительной концентрации на мерцающих стимулах.

Список литературы

1. User adaptive BCIs: SSVEP and P300 based interfaces / F. Beverina [et al.] // *PsychNology Journal*. – 2003. – № 1. – P. 331-354.
2. A comparison of techniques and technologies for SSVEP classification / R.M.G. Tello [et al.] // *Biosignals and Biorobotics Conference (2014) : Biosignals and Robotics for Better and Safer Living, Salvador, 26-28 May 2014*. – P. 1-6.
3. Reliable visual stimuli on LCD screens for SSVEP based BCI / H. Cecotti [etc.] // *18th European Signal Processing Conference, Aalborg, 23-27 Aug. 2010*. – P. 919-923.
4. Carmeli, C. Assessing cooperative behavior in dynamical networks with applications to brain data / C. Carmeli. – Lausanne : Ecole Polytechnique Federal de Lausanne, 2006. – 139 p.
5. Multivariate synchronization index for frequency recognition of SSVEP-based brain-computer interface / Y. Zhang [et al.] // *Journal of Neuroscience Methods*. – 2014. – № 221. – P. 32-40.

АЛГОРИТМИЧЕСКИЙ ПОДХОД К ИЗУЧЕНИЮ ПОЛИНОМИАЛЬНЫХ ОТОБРАЖЕНИЙ

С. Г. Косачев

Студент

А. В. Атанов

Ассистент

Введение

Данная работа посвящена обсуждению открытой до настоящего времени гипотезы о якобиане (Jacobian Conjecture, JC), впервые сформулированной Келлером в [1]. В простейшем двумерном случае она формулируется следующим образом.

Теорема (гипотеза о якобиане). Пусть $P(x, y)$ и $Q(x, y)$ — полиномы с комплексными коэффициентами, причем якобиан

$$J_f = \frac{\partial P}{\partial x} \cdot \frac{\partial Q}{\partial y} - \frac{\partial P}{\partial y} \cdot \frac{\partial Q}{\partial x} \equiv \delta = \text{const} \neq 0. \quad (1)$$

Тогда отображение $C^2 \rightarrow C^2$, $(x, y) \mapsto (u, v) = (P(x, y), Q(x, y))$ является взаимно-однозначным.

Существует множество подходов к изучению представленной проблемы, однако до настоящего времени получены лишь некоторые частные результаты. Отметим, что для рассмотрения гипотезы о якобиане в случае полиномов P и Q малых степеней активно привлекаются компьютерные алгоритмы (см., например, [2]).

Далее мы будем использовать один из вариантов гипотезы о якобиане, которая может быть сформулирована как утверждение о представимости любого келлерова отображения (т. е. полиномиального отображения $f = (P(x, y), Q(x, y))$), удовлетворяющего условию (1) двумерного комплексного пространства в виде суперпозиции аффинных невырожденных и треугольных преобразований этого пространства.

В работе изучаются келлеровы отображения, у которых степень полинома $P(x, y) = 2n$, а степень полинома $Q(x, y) = n$. Для случая $1 < n \leq 6$ изучаются условия существования таких отображений, а также представимость их в виде суперпозиций аффинных и треугольных преобразований. Решение поставленных задач проводится в рамках «коэффициентного» подхода (примененного, например, в [3]), заключающегося в рассмотрении ограничений на коэффициенты изучаемых отображений и сведении задачи к решению системы нелинейных уравнений, квадратично зависящих от коэффициентов полиномов $P(x, y)$ и $Q(x, y)$.

1. Сведение гипотезы о якобиане к решению системы нелинейных уравнений

Разложим многочлены $P(x, y)$ и $Q(x, y)$ на однородные составляющие:

$$\begin{cases} P(x, y) = P_n(x, y) + P_{n-1}(x, y) + \dots + P_1(x, y), \\ Q(x, y) = Q_m(x, y) + Q_{m-1}(x, y) + \dots + Q_1(x, y), \end{cases} \quad (2)$$

Отметим, что от констант P_0 и Q_0 , которые формально должны присутствовать в разложениях (2), можно легко избавиться за счет сдвига координат. Степени многочлена $P(x, y)$ и $Q(x, y)$ считаем равными, соответственно, степеням n и m их старших однородных компонент. Пару полиномов $(P(x, y), Q(x, y))$ будем называть (n, m) -отображением.

Якобиан J_f допускает такое же, как для P и Q , разложение на однородные составляющие:

$$J_f = R(x, y) = R_{n+m-2}(x, y) + \dots + R_1(x, y) + R_0(x, y) = \sum_{i=0}^{n+m-2} R_i. \quad (3)$$

В формулах (2) каждый из многочленов P_i и Q_j ($i = 0, \dots, n; j = 0, \dots, m$) является однородной формой степеней i и j соответственно:

$$P_i = A_{(i,0)}x^i + A_{(i-1,1)}x^{i-1}y + \dots + A_{(1,i-1)}xy^{i-1} + A_{(0,i)}y^i,$$

$$Q_j = B_{(j,0)}x^j + B_{(j-1,1)}x^{j-1}y + \dots + B_{(1,j-1)}xy^{j-1} + B_{(0,j)}y^j.$$

Полиномы R_i из равенства (3) определяются следующим образом:

$$R_i = \sum_{k+l=i+2} \left(\frac{\partial P_k}{\partial x} \cdot \frac{\partial Q_l}{\partial y} - \frac{\partial P_k}{\partial y} \cdot \frac{\partial Q_l}{\partial x} \right) = \sum_{k+l=i+2} [P_k, Q_l]. \quad (4)$$

Таким образом, учитывая записи (3) и (4), условие (1) можно свести к системе уравнений, каждое из которых описывает свою компоненту однородности многочлена $R(x, y)$:

$$\left\{ \begin{array}{l} R_{n+m-2}(x, y) = [P_n, Q_m] = 0, \\ R_{(n+m-2)-1}(x, y) = [P_n, Q_{m-1}] + [P_{n-1}, Q_m] = 0, \\ \dots \\ R_{(n+m-2)-k}(x, y) = [P_n, Q_{m-k}] + \dots + [P_{n-k}, Q_m] = 0, \\ \dots \\ R_1(x, y) = [P_2, Q_1] + [P_1, Q_2] = 0, \\ R_0(x, y) = [P_1, Q_1] = \delta \neq 0. \end{array} \right. \quad (5)$$

Отметим, что система (5) является нелинейной (квадратичной) относительно коэффициентов пары $(P(x, y), Q(x, y))$. Решение таких систем представляет собой значительные трудности, так как общего алгоритма, позволяющего получить точное решение нелинейной системы, как известно, не существует.

Таким образом, задачу изучения келлеровых отображений можно переформулировать как задачу нахождения решений системы (5) — точнее, поиска коэффициентов $A_{(i_1, j_1)}, B_{(i_2, j_2)}$ пары $(P(x, y), Q(x, y))$, удовлетворяющих этой системе.

2. Основные результаты

Были рассмотрены $(2n, n)$ -отображения $f = (P(x, y), Q(x, y))$, где $1 < n \leq 6$ (случай $n = 1$ тривиален). Кроме того, предполагалось, что

компонента $Q_m(x, y)$ в разложении (2) имеет вид $Q_m(x, y) = x^m$. Таким образом, требовалось изучить пять различных вариантов отображений: (4,2), (6,3), (8,4), (10,5) и (12,6).

Рассмотрение всех указанных случаев проводилось с использованием системы символьной математики Maple, в которой был реализован алгоритм, включающий следующую последовательность шагов:

1. Формирование полиномов $P(x, y)$ и $Q(x, y)$ заданных порядков.
2. Вычисление соответствующего якобиана.
3. Выделение коэффициентов при каждом мономе в выражении для якобиана (3) и формирование системы уравнений вида (5).
4. Поиск в системе уравнений, линейных по какому-либо коэффициенту $A_{(i1, j1)}, B_{(i2, j2)}$, и их решение.
5. Поиск в системе отдельных уравнений вида $T_1 \cdot T_2 \cdot \dots \cdot T_s = 0$, т. е. допускающих разложение левой части на множители, и рассмотрение подслучаев для каждого из вариантов: $T_1 = 0, T_2 = 0, \dots, T_s = 0$ (здесь T_i — полином, зависящий от коэффициентов $A_{(i1, j1)}, B_{(i2, j2)}$).

Следует отметить, что на начальном этапе число (нелинейных) уравнений в системе вида (5) очень велико и равно $(n+m)(n+m-1)/2$. Например, в случае (8,4) необходимо рассмотреть 66 уравнений, а в случае (12,6) — уже 153. При этом встроенные в системы символьной математики средства для автоматического решения систем уравнений не справляются с системами вида (5) даже при рассматриваемых небольших значениях n и m . Это обстоятельство и приводит к необходимости разработки специальных алгоритмов для решения подобных систем, максимально полно использующих их структуру.

Применение указанного алгоритма не позволяет автоматически решать всю систему до конца, однако существенно сокращает число уравнений. При этом, однако, система из квадратичной превращается в систему более высокого порядка. Тем не менее, во всех рассмотренных случаях получившиеся сокращенные системы были полностью решены, что позволило сделать следующие выводы о представимости всех изучаемых келлеровых отображений в виде треугольных преобразований.

Теорема 1. Келлеровы (4,2)-, (6,3)- и (10,5)-отображения представимы в виде следующих композиций треугольных преобразований:

- а) (1,2)- и (2,1)-отображений для случая (4,2);
- б) (1,3)- и (2,1)-отображений для случая (6,3);
- в) (1,5)- и (2,1)-отображений для случая (10,5);

Теорема 2. Келлерово (8,4)-отображение может быть представлено в виде следующих композиций треугольных преобразований:

- а) двойной композиции (1,4)- и (2,1)-отображений;
- б) тройной композиции (1,2)-, (2,1)- и (1,2)-отображений.

Теорема 3. Келлерово (12,6)-отображение может быть представлено в виде следующих композиций треугольных преобразований:

- а) двойной композиции (1,6)- и (2,1)-отображений;
- б) тройной композиции (1,2)-, (3,1)- и (1,2)-отображений;
- в) тройной композиции (1,3)-, (2,1) и (1,2)-отображений.

Получаемые в результате отображения имеют достаточно громоздкий вид. Поэтому приведем для примера лишь одну из композиций треугольных преобразований, отвечающую случаю б) теоремы 3:

$$\begin{cases} x_1 = x, \\ y_1 = y + \frac{3}{B_{41}}x^2 + \frac{B_{50}}{B_{41}}x, \end{cases} \quad \begin{cases} x_2 = \lambda_1 x_1 + \frac{B_{41}^3}{27}y_1^3 + \lambda_2 y_1^2 + \lambda_3 y_1, \\ y_2 = -\frac{\delta}{\lambda_1}y_1, \end{cases}$$

$$\begin{cases} x_3 = x_2, \\ y_3 = y_2 + A_{12,0}x_2^2 + \left(A_{60} - \frac{A_{12,0}}{729}\lambda_4 \right)x_2, \end{cases}$$

где

$$\begin{aligned} \lambda_1 &= (3B_{50}^3 B_{40} - B_{50}^2 - 27B_{50}B_{20} + 81B_{10})/81, \\ \lambda_2 &= B_{41}^2(3B_{40} - B_{50}^2)/27, \quad \lambda_3 = B_{41}(B_{50}^4 + 27B_{20} - 3B_{50}^2 B_{40})/81, \\ \lambda_4 &= 324B_{50}^2 B_{40}^2 + 1458B_{40}B_{20} - 180B_{50}^4 B_{40} + 1458B_{10}B_{50} + 25B_{50}^6. \end{aligned}$$

Заключение

Путем сведения гипотезы о якобиане к решению системы нелинейных уравнений (5), удалось показать, что $(2n, n)$ -отображения, где $1 < n \leq 6$, сводятся к суперпозициям треугольных преобразований, а, следовательно, удовлетворяют гипотезе. Рассмотрение даже таких максимально упрощенных примеров показывает значительные трудности, возникающие уже в случае небольших степеней полиномов $P(x, y)$ и $Q(x, y)$. Решение изучаемой системы (5) и, тем более, поиск суперпозиций треугольных преобразований для случаев более высоких степеней представляется очень сложной задачей. Отдельного внимания заслуживает не упоминающийся в литературе вопрос о количестве различных композиций треугольных преобразований, которыми может быть представлено келлерово отображение. Разработанная схема изучения структуры келлеровых

отображений является удобной при исследовании гипотезы о якобиане, однако требует дальнейшего усовершенствования и автоматизации.

Список литературы

1. Keller, O. Ganze Cremona-Transformationen / O. Keller // *Monatsh. Mat. Phys.* – 1939. – № 47. – P. 299-306.
2. Moh, T. On the global Jacobian Conjecture and the configuration of roots / T. Moh, J. Reine // *Angew. Math.* – 1983. – № 340. – P. 140-212.
3. Атанов, А. В. Об ограничениях и запретах на коэффициенты келлеровых отображений / А. В. Атанов, А. В. Лобода // *Вестник Воронежского государственного университета. Серия : Физика. Математика.* – 2008. – № 2. – С. 108-114.

МЕТОДЫ ЧИСЛЕННОЙ ОЦЕНКИ КАЧЕСТВА ВЫДЕЛЕНИЯ ГРАНИЦ НА ИЗОБРАЖЕНИЯХ

Д. С. Лебедько
Студент
Н. А. Тюкачёв
Доцент

Введение

За последнее десятилетие машинное зрение проникло во множество областей деятельности человека. Системы безопасности, основанные на анализе изображений, системы автоматизированного производства с контролем качества при помощи методов машинного зрения, системы анализа медицинских изображений — это лишь немногие примеры применения машинного зрения в прикладных отраслях деятельности человека. Важное место в процессе машинного зрения занимает этап сегментации изображений. Этап сегментации находится в самом начале цепочки преобразований системы машинного зрения. Очень часто сегментация сводится к выделению границ или контуров объектов на изображении. Границы объектов, выделяемые человеком, и границы объектов, выделяемые компьютером, очевидно, будут отличаться. Для достижения оптимального результата сегментации необходимо использовать численные методы оценки качества выделения границ объектов. В данной статье сначала даётся краткий обзор методов выделения границ объектов на изображении, после чего рассматриваются методы численной оценки качества выделения границ. Далее даётся краткое описание применения данных метрик на практике.

1. Методы выделения границ на изображениях

На практике наиболее широко используются **методы на основе операторов дифференцирования** первого и второго рода [1].

Методы на основе операторов дифференцирования первого рода выделяют широкие границы, однако устойчивы к шумам, в то время как методы на основе операторов дифференцирования второго рода выделяют тонкие границы и чувствительны к шумам. Далее приведены примеры наиболее часто используемых операторов дифференцирования первого порядка.

Оператор Робертса

$$\nabla I(i, j) = |I(i+1, j+1) - I(i, j)| + |I(i+1, j) - I(i, j+1)| \quad (1)$$

Оператор Собеля

$$\begin{aligned} \nabla I(i, j) = & |(I(i-1, j-1) + 2 * I(i-1, j) + I(i-1, j+1)) - \\ & -(I(i+1, j-1) + 2 * I(i+1, j) + I(i+1, j+1))| + \\ & + |(I(i-1, j-1) + 2 * I(i, j-1) + I(i+1, j-1)) - \\ & -(I(i-1, j+1) + 2 * I(i, j+1) + I(i+1, j+1))|. \end{aligned} \quad (2)$$

Оператор Превитта

$$\begin{aligned} \nabla I(i, j) = & |(I(i-1, j-1) + I(i-1, j) + I(i-1, j+1)) - \\ & -(I(i+1, j-1) + I(i+1, j) + I(i+1, j+1))| + \\ & + |(I(i-1, j-1) + I(i, j-1) + I(i+1, j-1)) - \\ & -(I(i-1, j+1) + I(i, j+1) + I(i+1, j+1))|. \end{aligned} \quad (3)$$

Операторы Собеля и Превитта имеют наибольшее значение отклика на горизонтальных и вертикальных перепадах яркости, в то время как оператор Робертса имеет наибольшее значения отклика на диагональных перепадах яркости.

Наиболее часто используемым оператором дифференцирования второго порядка является **оператор Лапласа**. Существуют различные варианты дискретного приближения оператора Лапласа, но в данной статье для краткости будет приведён лишь один вариант его вычисления.

$$\begin{aligned} \Delta I(i, j) = & I(i-1, j-1) + I(i-1, j) + I(i-1, j+1) + I(i, j-1) + \\ & + I(i, j+1) + I(i+1, j-1) + I(i+1, j) + I(i+1, j-1) - 8 * I(i, j). \end{aligned} \quad (4)$$

Все вышеперечисленные методы выделения границ работают с изображением размера $m \times n$ и дают на выходе изображение той же размерности со значением градиента или лапласиана для каждой точки исходного изображения. Для получения границ изображения необходимо провести пороговую обработку полученного изображения. Пороговая обработка может быть как глобальной, так и локальной, в зависимости от контекста решаемой задачи.

Кроме перечисленных дифференциальных методов выделения границ существуют методы выделения границ, основанные на предварительной пороговой обработке, в результате которой изображение делится на фон и объект. Такие методы также известны как **методы выделения контуров по регионам**. Предположим, что точки объекта имеют белый цвет, а фон – чёрный. В таком случае задача определения границ объекта сводится к прохождению по контуру объекта начиная с определённой стартовой точки. При повторном достижении стартовой точки считается, что контур объекта был выделен.

Примерами методов, работающих по такому принципу являются **метод выделения контуров по квадратам** (Square Tracing) и **метод выделения контуров по квадратам с учётом окрестности Мура** (Moore Neighbour Tracing) [5].

Алгоритм метода выделения контуров по квадратам в формальном виде может быть описан следующим образом:

1. Поиск из верхнего угла точки $s \in P$.
2. Добавить s к B .
3. Положить текущую стартовую точку s точкой p .
4. Повернуть из текущей точки p налево и запомнить новую точку в p .
5. Если точка p белая – повернуть налево и добавить p в B . Если же точка p чёрная – повернуть направо.
6. Если p не равно s – вернуться к шагу 5.

Следует отметить, что для поворота необходимо хранить текущее направление движения.

Алгоритм метода выделения контуров по квадратам с учётом окрестности Мура:

1. Поиск из верхнего угла точки $s \in P$.
2. Добавить s к B .
3. Положить текущую стартовую точку s точкой p .
4. Вернуться назад в точку, предшествующую точке s т. е. $c = previous(s)$.
5. Переместиться в следующую точку окрестности Мура точки p по часовой стрелке т. е. $c = clockwise(M(p), state)$.
6. Если цвет точки c белый, то необходимо добавить c к B , запомнить найденную точку в p т.е. $p = c$ и вернуть в c ту точку, из которой был осуществлён вход в p т.е. $c = previous(p)$.
7. Если $c = s$, необходимо прекратить построение контура. В противном случае вернуться к шагу 5.

где I – входное одноканальное бинарное изображение, содержащее связную компоненту $P \subset I$. $B(b_1, b_2, \dots, b_n) \subset P$ – контур связной компоненты P .

2. Методы численной оценки качества выделения границ

Границы объектов, получаемые в результате сегментации, могут иметь разрывы, точки шума, а также могут быть смещены относительно действительного с точки зрения человека положения. Методы численной оценки качества выделения границ позволяют определить качество полученных результатов, учитывая наличие перечисленных ошибок. Далее все рассматриваемые метрики качества предполагают наличие идеального изображения, содержащего границы объектов.

Самыми простыми численными метриками качества являются **вероятность верного определения граничной точки и вероятности ошибок первого и второго рода.**

$$P_{co} = \frac{TP}{\max(N_I, N_B)}, \quad (5)$$

$$P_{nd} = \frac{FN}{\max(N_I, N_B)}, \quad (6)$$

$$P_{fa} = \frac{FP}{\max(N_I, N_B)}. \quad (7)$$

Данные метрики позволяют оценить качество выделения границ в предположении, что объект на изображении статичен. Это значит, что применение данных метрик невозможно в случае априорного знания о том, что объект может быть смещён относительно идеального положения.

Проблема оценки качества выделения границ подвижного объекта может быть решена при помощи использования **метрики Прэтта** [2]:

$$IMP = \frac{1}{\max(N_I, N_B)} \sum_{i=1}^{N_B} \frac{1}{1 + \alpha * d_i}. \quad (8)$$

Данная метрика вводит штраф за смещение границы объекта, также за наличие лишних граничных точек.

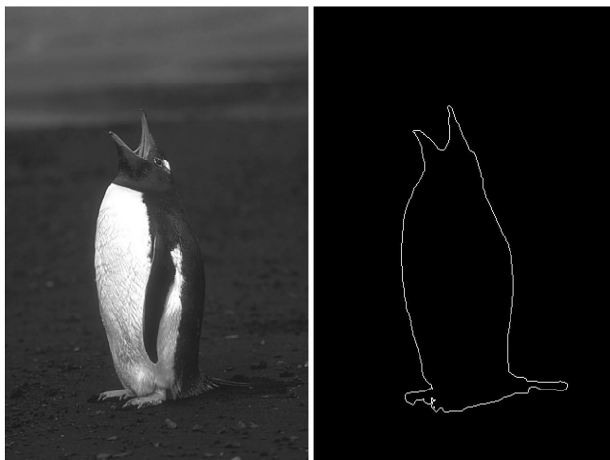
Величины (5)-(8) могут быть объединены в составную метрику, определённую следующим образом:

$$D4 = \sqrt{(P_{co} - 1)^2 + (IMP - 1)^2 + P_{nd} + P_{fa}}. \quad (9)$$

Данная метрика имеет значения в диапазоне $[0, 2]$: 0 для идеального детектора границ и 2 – для наихудшего.

3. Пример применения метрик на практике

На рис. 1 представлены исходное изображение животного и изображение его контура подготовленного человеком.



а

б

а – оригинальное изображение, б – контур

Рис. 1. Тестовое изображение

Из рис. 2 можно увидеть, что в некоторых точках метрика Прэтта принимает максимальные значения для каждого из методов выделения контуров. Выбрав те пороги бинаризации, где достигаются максимальные значения метрики, можно получить результаты сегментации, представленные на рис. 3.

Выбор оптимальных порогов сегментации также может быть проведён при помощи минимизации метрики $D4$.

Данные метрики позволяют не только оценивать качество выделения границ объектов на изображении, но также могут быть использованы для автоматического выбора оптимальных порогов сегментации. Такой подход может сочетаться с методом выбора порогов сегментации при помощи построения частотной гистограммы уровней яркости изображения. Сначала при помощи гистограммы может быть выбран диапазон возможных порогов сегментации, после чего оптимальный порог сегментации будет выбран описанным ранее методом максимизации значения IMP метрики или минимизации значения метрики $D4$.

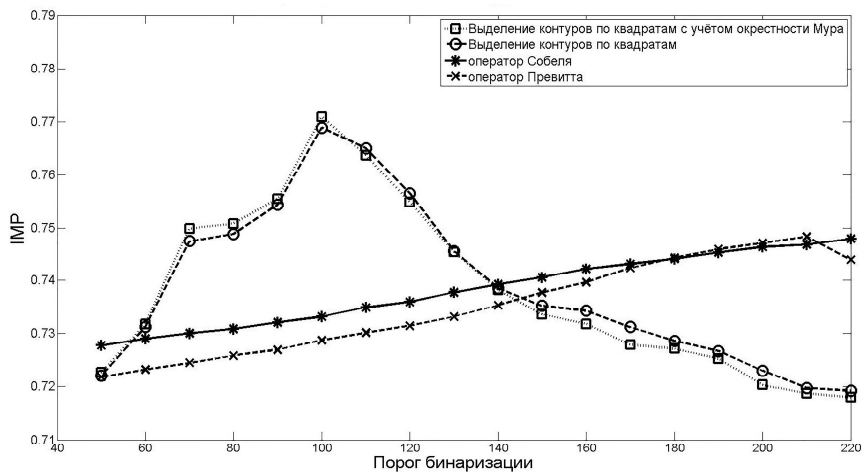


Рис. 2. Зависимость метрики Прэтта от порога бинаризации при использовании различных методов выделения границ

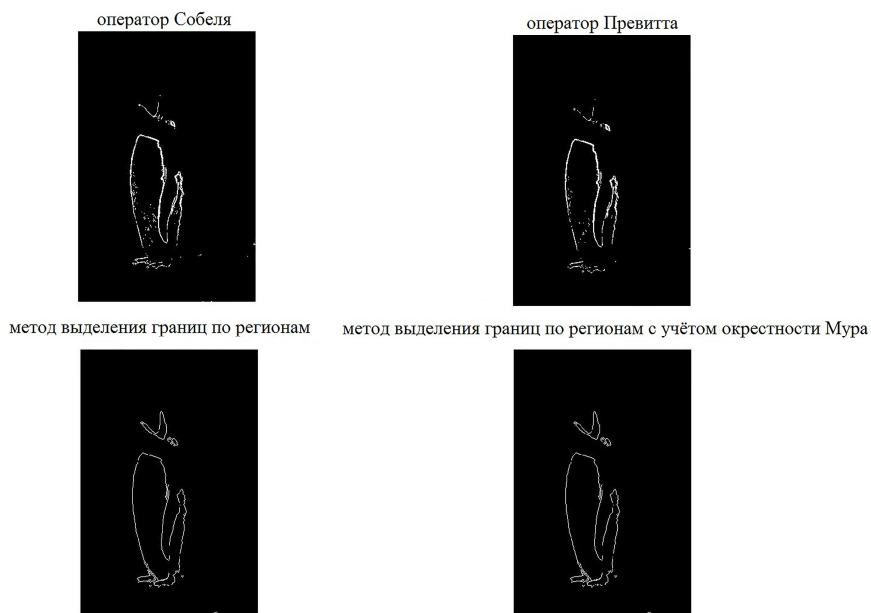


Рис. 3. Результат сегментации с оптимальными по критерию IMP порогами

Заключение

В данной статье кратко рассмотрены некоторые методы выделения границ на изображениях, а также метрики оценки качества работы данных методов. Кроме того, в статье предложено использование рассмотренных метрик оценки качества как средства выбора оптимальных порогов сегментации для достижения наилучшего качества сегментации. Рассмотренные методы оценки качества выделения границ объектов на изображении были реализованы в web-ориентированной информационной системе, которая позволяет исследовать методы выделения границ на изображении.

Список литературы

1. Гонсалес, Р. Цифровая обработка изображений / Р. Гонсалес, Р. Вудс. – Москва : Технофера, 2005. – 1070 с.
2. Прэтт, У. Цифровая обработка изображений / У. Прэтт. – Москва : Мир, 1982. – 790 с.
3. Теоретические основы цифровой обработки изображений / В. А. Сойфер, В. В. Сергеев, С. Б. Попов [и др.]. – Самара : Самарский государственный аэрокосмический университет им. академика С. П. Королёва, 2000. – 255 с.
4. Toussaint, G. Introduction to pattern recognition [Электронный ресурс]: электронная книга / G. Toussaint. – Режим доступа : <http://cgm.cs.mcgill.ca/~godfried/teaching/mir-reading-assignments/Chapter-1-Introduction-Pattern-Recognition.pdf>
5. Toussaint, G. Grids, connectivity and contour tracing [Электронный ресурс] : электронная книга / G. Toussaint. – Режим доступа : <http://cgm.cs.mcgill.ca/~godfried/teaching/mir-reading-assignments/Chapter-2-Grids-Connectivity-Contour-Tracing.pdf>

РАЗРАБОТКА ПРОГРАММНО-АППАРАТНОГО КОМПЛЕКСА ДЛЯ КОНТРОЛЯ И АНАЛИЗА РАДИОЧАСТОТНОГО СПЕКТРА С ИСПОЛЬЗОВАНИЕМ SDR ТЕХНОЛОГИЙ

В. С. Левин

Студент

Ю. Б. Нечаев

Профессор

Введение

Повышение за последнее десятилетие пропускной способности сетей передачи данных с 100 Мб/с до 1000 Мб/с и выше, прогресс в цифровой обработке радиосигналов, широкое проникновение во все области деятельности беспроводных технологий обусловили появление новых видов радио-систем. Основная тенденция – локализация аналоговых элементов радиосистем (фильтров, усилителей, генераторов несущей частоты, цифро-аналоговых и аналого-цифровых преобразователей) на входах/выходах радиосистем, широкое применение цифровых методов обработки радиосигналов, децентрализация цифровой обработки. Под децентрализацией понимается применение различных типов вычислительных средств на разных этапах обработки сигналов и разнесение этих вычислительных средств в пространстве и времени. Сочетание оцифрованного радиосигнала с метками, показывающими, когда и где была произведена оцифровка, позволяет хранить, передавать, обрабатывать радиосигнал огромным количеством способов, извлекать из радиосигнала не только передаваемые данные, но и временную, пространственную и другую информацию. В данной статье рассмотрены SDR технологии. Основная особенность SDR технологий в том, что аппаратная часть минимизирована, а алгоритмы обработки сигнала реализуются программно, что позволяет легко модернизировать их, графически демонстрировать процессы, происходящие в различных точках системы.

1. Программно-определяемая радиосистема

В классическом представлении программно-определяемая радиосистема (Software Defined Radio, SDR) это радиоустройство с программируемыми параметрами (SDR): радиопередатчик и/или радиоприемник, использующий технологию, позволяющую с помощью программного обеспечения устанавливать или изменять рабочие

радиочастотные параметры, включая, в частности, диапазон частот, тип модуляции или выходную мощность, за исключением изменения рабочих параметров, используемых в ходе обычной предварительно определенной работы с предварительными установками радиоустройства, согласно той или иной спецификации или стандарта системы [1, 2].

Целью работы является разработка программно-аппаратного комплекса для контроля и анализа радиочастотного спектра с использованием SDR технологий.

Область применения комплекса:

- мониторинг работы различных передатчиков в указанном диапазоне;
- для обучения специалистов в области радио-технологий;
- в военных целях;
- в научных целях.

С помощью специальных драйверов многие современные компьютерные ТВ-тюнеры, например, RTL2832 (рис. 1) с USB входом для приёма цифрового телевидения, могут управляться с помощью соответствующего программного обеспечения и передавать на компьютер принятый и оцифрованный сигнал.

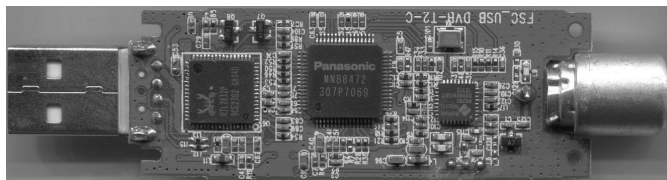


Рис.1. ТВ-тюнер RTL2832

Технические характеристики RTL2832:

- Диапазон частот: 52 – 2200 мГц;
- модуляция: AM, FM, NFM, LSB, USB, CW (ADS-B, D-STAR, AIS и другие виды);
- полоса обзора: меняется от 250кГц до 3МГц;
- чувствительность: 0.22мКв (на 438МГц, в режиме NFM);
- входное сопротивление приёмника: 50 Ом;
- диапазонные фильтры: только внешние
- разрядность АЦП: 8бит;
- динамический диапазон: 50 дБ (в режиме CW);
- задержка принимаемого сигнала: 340мсек;
- интерфейс: USB 2.0;
- операционная система: Windows, Linux, Android.

Антенна, подключенная к тюнеру, принимает сигналы с эфира. Чип E4000 выделяет участок радиодиапазона, начинающийся там, где мы ему указали и шириной 3 МГц, усиливает его. Чип RTL2832 оцифровывает этот участок и передает по USB на компьютер. Программа (GnuRadio, HSDR или Matlab RTL-SDR) настраивается на выбранную частоту в пределах выбранных ранее 3 МГц, выполняет демодуляцию указанным способом и отправляет получившийся звук на звуковую карту. Также она может отправлять команды чипу E4000 на перестройку на другой участок диапазона [3].

2. Программно-аппаратный комплекс

В настоящее время ведутся работы в среде Matlab RTL-SDR над программно-аппаратным комплексом для контроля и анализа радиочастотного спектра с использованием SDR технологий. Блок схема комплекса в среде моделирования Matlab Simulink представлена на рис. 2.

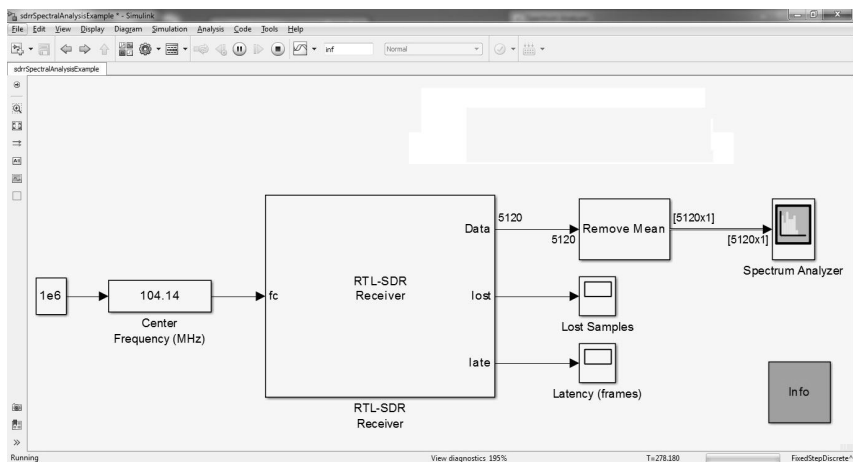


Рис. 2. Элемент реализации комплекса в среде моделирования Simulink

Блок Center Frequency(MHz) отвечает за выбор частоты на частотном диапазоне. RTL-SDR Receiver отвечает за данные полученные с SDR-приемника. Lost Samples отвечает за информацию о пропущенных кадрах. Latency отвечает за информацию о задержке кадров. Spectrum Analyzer показывает анализ радиочастотного спектра (рис. 3).

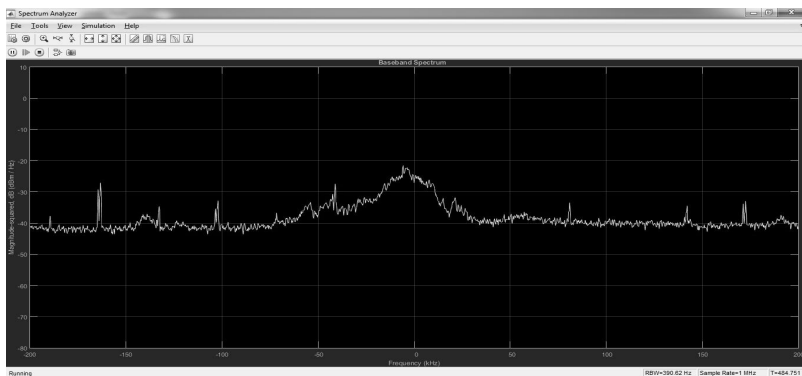


Рис. 3. Спектр аналогового телеканала

В рабочем состоянии комплекс может измерять характеристики сигнала:

- пики занимаемой полосы частот;
- пики коэффициентов мощности соседних каналов;
- пики гармонических и интермодуляционных уровней;
- сможете стимулировать свободный динамический диапазон.

Заключение

USB-приемники RTL-SDR могут служить бюджетным вариантом SDR-платформы для решения целого ряда практических задач. При выполнении лабораторных работ и курсовых проектов в процессе изучения дисциплин, связанных с обработкой сигналов и физического уровня телекоммуникационных систем, в бизнесе, в военных целях. Основная особенность в том, что аппаратная часть минимизирована, а алгоритмы обработки сигнала реализуются программно, что позволяет легко модернизировать их, графически демонстрировать процессы, происходящие в различных точках системы.

Список литературы

1. Международный союз электросвязи. Определения системы радиосвязи с программируемыми параметрами (SDR) и системы когнитивного радио (CRS) / Международный союз электросвязи. Отчет МСЭ-R SM.2152. Электронная публикация. – Женева, 2010. – С. 2-5.
2. Сорохтин, Е. М. Распределенные программно-определяемые радиосистемы / Е. М. Сорохтин, С. А. Минеев // Вестник Нижегородского университета им. Н. И. Лобачевского. – 2010. – № 5. – С. 383-388.
3. Борисов, В. Г. Юный радиолюбитель / В. Г. Борисов, Ф. Н. Горбунов. – Москва : Радио и связь, 1992. – 409 с.

РЕАЛИЗАЦИЯ GPS-ТРЕКЕРА НА ОСНОВЕ МИКРОКОНТРОЛЛЕРА

Д. А. Мезин
Студент
Д. Н. Борисов
Доцент

Введение

Особое место в развитии систем спутниковой связи занимает возможность отслеживания местоположения объекта. Процесс отслеживания – это комплекс методов измерений, позволяющих измерить точное местоположение объекта в реальном времени.

Данный комплекс методов основан на измерении расстояния от антенны на объекте, координаты которого необходимо получить, до спутников, положение которых известно с большой точностью. В настоящее время процесс отслеживания объектов используется практически в любом направлении деятельности, в частности можно выделить наиболее значимые направления:

- отслеживание места положения человека;
- отслеживания перевозки грузов;
- отслеживание положения военных объектов.

Передача полученных координат устройства на основе микроконтроллера может осуществляться по разным каналам, однако наиболее целесообразно передавать их по беспроводному каналу с использованием GPRS [1].

1. Выбор компонентов

Целью работы является реализация GPS-трекера на основе микроконтроллера. Для достижения поставленной цели необходимо решить следующие задачи:

- анализ микроконтроллеров и модулей;
- анализ среды разработки;
- разработка приложения;
- тестирование GPS-трекера.

Для начала было необходимо проанализировать существующие аналоги на рынке, выбрать подходящие компоненты для разрабатываемого устройства. Они должны удовлетворять ряду условий: быть доступными, легкими в освоении и иметь подходящую стоимость. Анализ показал, что для реализации трекера достаточно 3-х компонентов: GPS приемника,

GPRS передатчика и микроконтроллера в качестве управляющего устройства.

GPS приемник. Задача приёмника – принять сигналы со спутников, определить текущее положение спутников, определить расстояние до спутников, определить своё местоположение, скорость, направление движения и отправить эти данные на управляющее устройство. В качестве приёмника в GPS-трекере выступает GPS модуль Skylab GPS SKM53, представленный на рис. 1.



Рис. 1. Приемный модуль Skylab GPS SKM53

GPRS передатчик. Задача передатчика – передать полученные данные с приёмника на удалённое устройство. В качестве передатчика в GPS-трекере выступает GPRS модуль SIM 900, представленный на рис. 2.

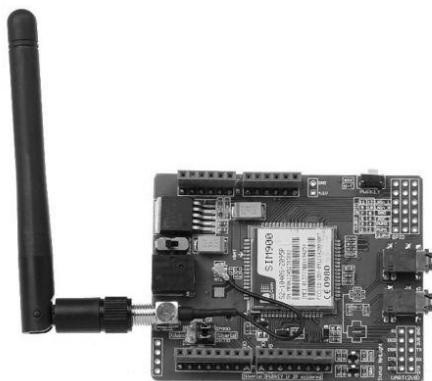


Рис. 2. GPRS модуль SIM 900

Управляющее устройство. Задача управляющего устройства – связать в единую систему (трекер) приёмник и передатчик, т. е. принять данные с приёмника и отправить с помощью передатчик на удалённый сервер. В качестве управляющего устройства в GPS-трекере выступает микроконтроллер Arduino Mega 2560, представленный на рис. 3.

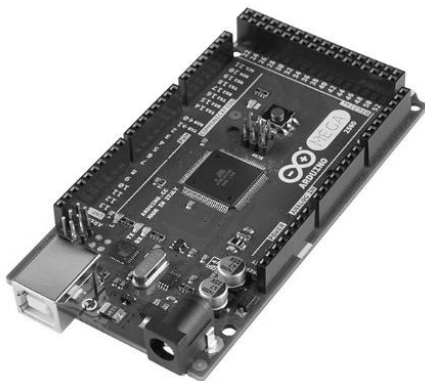


Рис. 3. Arduino Mega 2560

2. Разработка приложения

При написании приложения для GPS-трекера использовалась среда Atmel Studio [2]. Основное преимущество платформы Atmel Studio 6 состоит в том, что она абсолютно бесплатна. В нее интегрирована обширная библиотека бесплатного исходного кода Atmel Software Framework (ASF) а так же, она содержит свыше 1600 примеров ARM- и AVR-проектов [3]. В качестве программатора использовалась плата AVR Dragon Atmel (рис. 4).

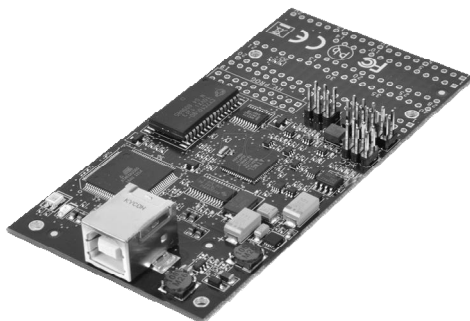


Рис. 4. AVR Dragon Atmel

Программатор AVR DRAGON фирмы Atmel. Используется для написания, отладки и «заливки» (передачи) приложения на микроконтроллер [4].

На рис. 5 показан упрощенный алгоритм работы приложения.



Рис. 5. Основной алгоритм работы приложения

Основной принцип программирования микроконтроллеров заключается в прерывании [5]. На рис. 6. показан пример отправки команды в модем и ожидание ответа от него.

```

void modem_send_cmd (const char* string)
{
    while (1) //ожидать до появления флага принятого "OK" от модема
    {
        _delay_ms(200);
        modem_flag_ok = 0;
        modem_send_string(string);
        for (unsigned char j=0; j< MODEM_WAIT_OK; j++) //ожидать ответа заданное кол-во времени
        {
            if (modem_flag_ok) break;
            _delay_ms(10);
        }
        if (modem_flag_ok) //ответ получен - выход
        {
            modem_flag_ok = 0;
            return;
        }
    }
}
}

```

Рис. 6. Пример кода отправки команды в модем и ожидание ответа

3. Тестирование GPS-трекера

Перед включением необходимо установить SIM-карту со снятым PIN-кодом и положительным балансом в держатель платы GPRS. Подключить антенну в плате GPRS. Установить переключатель питания платы GPRS в положение «EXT». Резистор, подключенный к плате Arduino, нужен для предотвращения ложных сбросов микроконтроллера при повышенном уровне электромагнитных помех. Включить адаптер питания в сеть 220В. Нажать и удерживать 2 секунды кнопку включения (PWRKEY) на плате GPRS.

На плате Arduino Mega должны загореться зеленый и красный светодиоды, на плате GPRS – зеленый светодиод рядом с кнопкой «PWRKEY» и красный светодиод «STATUS». Частые вспышки светодиода «NETLIGHT» платы GPRS говорят о процессе регистрации в сети GSM, редкие вспышки – об успешной регистрации.

На рис. 7 представлен GPS-трекер в собранном виде. В результате работы трекера, после установления его местоположения, он отправляет SMS в соответствующем виде (рис. 8) на прописанный в программе номер.

Заключение

Данная статья посвящена реализации GPS-трекера на основе микроконтроллера Arduino Mega 2560 с использованием GPS и GPRS модулей. Проведен выбор компонентов микроконтроллера и разработано приложение для него. В итоге получилась полностью рабочая система для определения координат местоположения и отправки его с использованием технологии Short Message Service.

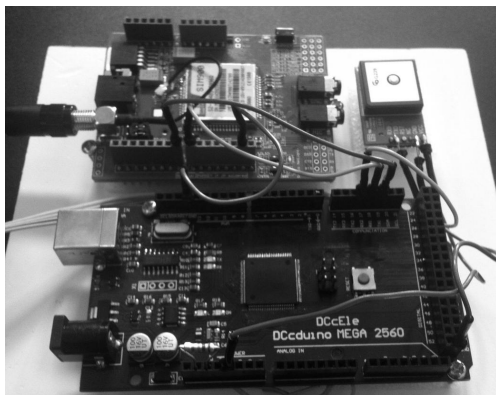


Рис. 7. GPS-трекер

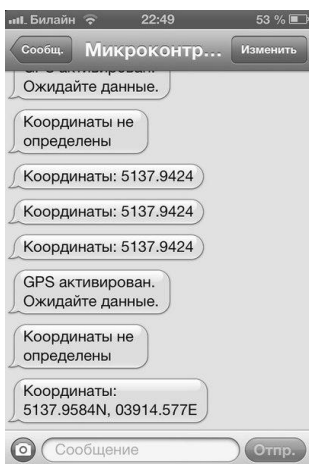


Рис. 8. Сообщение присланное с GPS-трекера

Список литературы

1. Щеголевых, П. А. Реализация охранной системы на контроллере Netduino Plus с использованием GSM и LAN каналов для передачи данных / П. А. Щеголевых, Д. Н. Борисов // Сборник студенческих научных работ факультета компьютерных наук ВГУ. – Под ред. Д.Н. Борисова. – Вып. 8. – Воронеж : Издательский дом ВГУ, 2014. – С. 195-200.
2. Белов, А. Б. Конструирование устройств на микроконтроллерах / А. Б. Белов. – Москва : Наука и Техника, 2005. – 255 с.

3. Предко, М. В. Руководство по микроконтроллерам. Том 1 / М. В. Предко. – Москва : Постмаркет, 2001. – 416 с.
4. Предко, М. В. Руководство по микроконтроллерам. Том 2 / М. В. Предко. – Москва : Постмаркет, 2001. – 488 с.
5. Вуд, А. Микропроцессоры в вопросах и ответах / А. Вуд. – Москва : Энергоатомиздат, 1985. – 184 с.

ОПРЕДЕЛЕНИЕ ПОЛОЖЕНИЯ УСТРОЙСТВА В ПРОСТРАНСТВЕ ПО ЕГО ДАТЧИКАМ

А. А. Назарова
Студент
В. С. Тарасов
Ассистент

Введение

Современные технологии развиваются быстрыми темпами. У устройств появляются все новые возможности. Одной из них является ориентация устройства в пространстве. Она используется во многих программах, примерами которых могут служить игры, в которых управление осуществляется за счет поворота устройства, или же программы построения трехмерных карт местности по снимкам видеокамеры, управления как самим устройством, так и другими за счет его перемещения и т.д.

Ориентация мобильного устройства в пространстве определяется за счет инерциальных навигационных систем (ИНС) – систем, основанных на свойствах инерции тел. Их преимуществом является то, что они автономны, то есть не требуют информации о внешних ориентирах или поступающих извне сигналах. Работают такие системы с помощью получения и обработки данных с датчиков параметров углового движения и ускорения. Мобильное устройство является бесплатформенной ИНС, так как оба датчика располагаются на самом устройстве.

1. Проецирование данных

Цель исследования состоит в разработке алгоритма, позволяющего обрабатывать информацию с датчиков положения мобильного устройства и строить траекторию его движения.

Датчиками, предоставляющими информацию об ускорениях и углах поворота являются акселерометр и гироскоп соответственно. Данные считываются по каждой из трех координатных осей, ориентированных

относительно мобильного устройства. В процессе движения оси меняют свое направление вместе с ним, что приводит к задаче проецирования данных на неподвижную систему координат. Так же из-за высокой чувствительности датчиков необходимо решить задачу фильтрации данных.

Проецирование данных на неподвижную систему координат осуществляется с помощью расчета углов поворота устройства (крена, тангажа и рысканья). Их можно вычислить тремя способами:

- по данным гироскопа (используя интеграл);
- по данным акселерометра (с помощью тригонометрических функций);
- по данным с обоих датчиков (на основе комплементарного фильтра).

Гироскоп позволяет вычислить точные углы поворота в краткосрочной перспективе, но со временем из-за интегрирования ошибка показаний накапливается. Акселерометр, наоборот дает зашумленное измерение углов из-за его чувствительности к внешним воздействиям, но его показания не зависят от предыдущих. Поэтому для вычисления углов поворота используется комплементарный фильтр, который представляет результирующий угол A , как сумму углов вычисленных по показаниям гироскопа A_g и акселерометра A_a , взятых с определенным весом, который регулируется коэффициентом K :

$$A = (1 - K)A_g + KA_a$$

Значение коэффициента K выбирается опытным путем в зависимости от характеристик датчиков устройства, обычно значение колеблется в пределах от 0.01 до 0.15. Так для мобильного устройства было выбрано значение 0.08, отклонение для которого составило $\pm 0.06^\circ$.

Поворот осуществляется с помощью кватернионов, так как по сравнению с матрицами поворота, они требуют вычисления меньшего количества математических операций и обладают легкостью интерполяции [1]. Кватернион в каждый момент времени $q = [q_r, q_i, q_j, q_k]$ вычисляется из углов крена (ϕ), тангажа (θ) и рысканья (ψ) следующим образом:

$$q_r = \cos(\psi/2)\cos(\theta/2)\cos(\phi/2) - \sin(\psi/2)\sin(\theta/2)\sin(\phi/2),$$

$$q_i = \cos(\psi/2)\cos(\theta/2)\sin(\phi/2) - \sin(\psi/2)\sin(\theta/2)\cos(\phi/2),$$

$$q_j = \cos(\psi/2)\sin(\theta/2)\cos(\phi/2) - \sin(\psi/2)\cos(\theta/2)\sin(\phi/2),$$

$$q_k = \sin(\psi/2)\cos(\theta/2)\cos(\phi/2) - \cos(\psi/2)\sin(\theta/2)\sin(\phi/2).$$

А проецирование осуществляется с помощью матрицы кватернионов:

$$Q = \begin{bmatrix} 1 - 2q_j^2 - 2q_k^2 & 2(q_i q_j - q_k q_r) & 2(q_i q_k + q_j q_r) \\ 2(q_i q_j + q_k q_r) & 1 - 2q_i^2 - 2q_k^2 & 2(q_j q_k - q_i q_r) \\ 2(q_i q_k - q_j q_r) & 2(q_j q_k + q_i q_r) & 1 - 2q_i^2 - 2q_j^2 \end{bmatrix}.$$

Тогда спроецированный вектор ускорений (\bar{a}') вычисляется по формуле:

$$\bar{a}' = Q\bar{a}.$$

2. Фильтрация данных и расчет траектории

Существуют различные методы обработки информации и одним из них является фильтр Калмана [2], который является рекурсивным фильтром и позволяет убрать шумы и лишние данные.

Отличительной чертой фильтра Калмана является то что он оперирует не только оценками состояния, но и плотностями его распределения, на основе формулы Байеса условной вероятности [3]:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)},$$

где $P(A)$ – априорная вероятность гипотезы A ; $P(A|B)$ – вероятность гипотезы A при наступлении события B (апостериорная вероятность); $P(B|A)$ – вероятность наступления события B при истинности гипотезы A ; $P(B)$ – полная вероятность наступления события B .

Большинство явлений подчиняется Гауссовскому распределению вероятностей, которое в одномерном случае задается функцией плотности вероятностей:

$$p = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}},$$

где μ – математическое ожидание (среднее значение) распределения; σ – среднеквадратичное отклонение распределения.

Тогда работу фильтра можно описать следующим образом: пусть фильтр помнит значение x_1 и его дисперсию σ_1^2 , то при поступлении нового значения x_2 с отклонением σ_2^2 , фильтр перемножая их предсказывает значение x_3 с отклонением меньшим, чем у двух предыдущих. Результат представлен на рис.1:

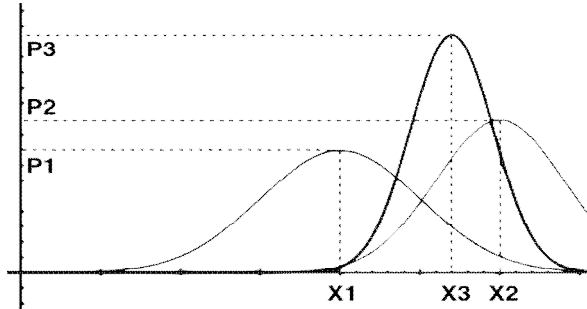


Рис. 1. Плотность распределений вероятности

Работа фильтра основана на двух повторяющихся циклических стадиях: предсказание и корректировка.

Предсказание состояния системы строится на основе начальных данных x_{k-1} и P_{k-1} . Следующая формула дает предсказание состояния системы:

$$\hat{x}_{k-1} = F_k \hat{x}_{k-1|k-1} + B_k u_k,$$

а также выполняется предсказание ошибки ковариации:

$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_k,$$

где $\hat{x}_{k|k-1}$ – предсказание состояния системы в момент времени t_k при учете наблюдений с начала работы фильтра и по момент t_{k-1} включительно; F_k – матрица перехода между состояниями; B_k – матрица применения управляющего воздействия; u_k – известная управляющая функция; $P_{k|k-1}$ – предсказание ошибки; Q_k – ковариация шума процесса.

После предсказания, как уже говорилось происходит корректировка. В каждый момент времени значение координаты зашумлено, поэтому предположим, что y_k тоже является нормально распределенным с математическим ожиданием равным 0 и нормальным отклонением равным σ_z . Вычисляется отклонение наблюдения на k шаге от ожидаемого при предсказании:

$$y_k = z_k - H_k \hat{x}_{k|k-1}$$

и ковариационная матрица для вектора ошибки:

$$S_k = H_k P_{k|k-1} H_k^T + R_k,$$

Рассчитывается коэффициент Калмана:

$$K_k = P_{k|k-1} H_k^T S_k^{-1},$$

Вычисляется оценка вектора состояния системы:

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k y_k.$$

и ковариационная матрица для оценки вектора состояния системы:

$$P_{k|k} = (I - K_k H_k) P_{k|k-1},$$

где z_k – измерение в текущий момент времени; K_k – матрица коэффициентов усиления Калмана; $R_k = M \begin{bmatrix} y_k & y_k^T \end{bmatrix}$ – ковариация шума измерения; $H_k = \begin{bmatrix} 1 & 0 \end{bmatrix}$ – матрица, определяющая отношение между измерениями и состоянием системы.

Применим фильтр Калмана к данным акселерометра по координатной оси Ox [4]:

$$x_{k+1} = x_k + v_k^x dt + \frac{a_k^x dt^2}{2},$$

$$v_{k+1}^x = v_k^x + a_k^x dt.$$

Состояние системы имеет вид:

$$X_k = \begin{pmatrix} x_k \\ v_k^x \end{pmatrix}.$$

Матрица перехода, для применения фильтра к уравнениям кинематики, принимает следующий вид:

$$F_k = \begin{bmatrix} 1 & dt \\ 0 & 1 \end{bmatrix},$$

вводится матрица $G_k = \begin{bmatrix} dt^2/2 \\ dt \end{bmatrix}$, а $B_k = 0$, так как управляющее воздействие отсутствует.

Тогда

$$Q = G_k G_k^T \sigma_a^2 = \begin{bmatrix} \frac{dt^4}{4} & \frac{dt^3}{2} \\ \frac{dt^3}{2} & dt^2 \end{bmatrix} \sigma_a^2.$$

Тогда для $k+1$ шага:

$$x_{k+1} = \begin{bmatrix} 1 & dt \\ 0 & 1 \end{bmatrix} x_k + \begin{bmatrix} dt^2/2 \\ dt \end{bmatrix} a_k^x.$$

В начальный момент времени $x_{0|0} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, если начальная скорость известна, то матрица ковариации $P_{0|0} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$ – нулевая, а если начальная скорость не известна, то $P_{0|0} = \begin{bmatrix} L & 0 \\ 0 & L \end{bmatrix}$, где L – достаточно большое число, так как в этом случае далее фильтр предпочтет данные, полученные из предсказаний, чем уже существующие в системе.

Применяя эти формулы к данным был получен результат, представленный на рис. 2:

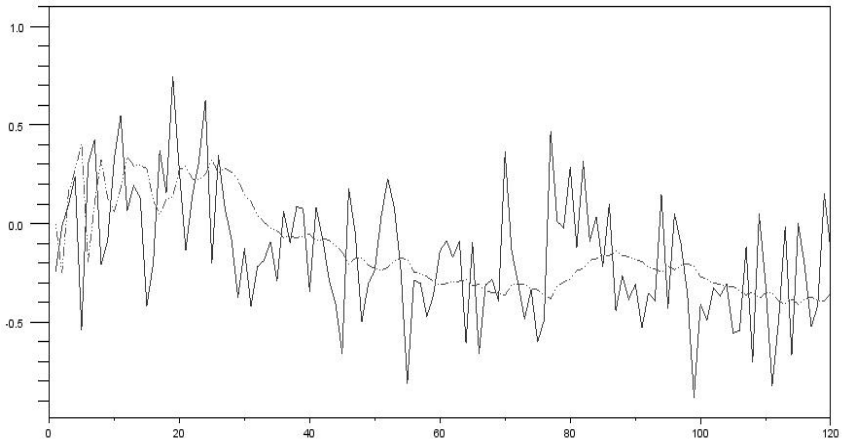


Рис. 2. Применение фильтра Калмана

На нем сплошной линией отображена нефильтрованная информация о координатах устройства, а штриховая линия – информация обработанная с помощью фильтра Калмана.

Изменение ошибки результатов вычисления представлено на рис. 3, (для значений скорости и для координат).

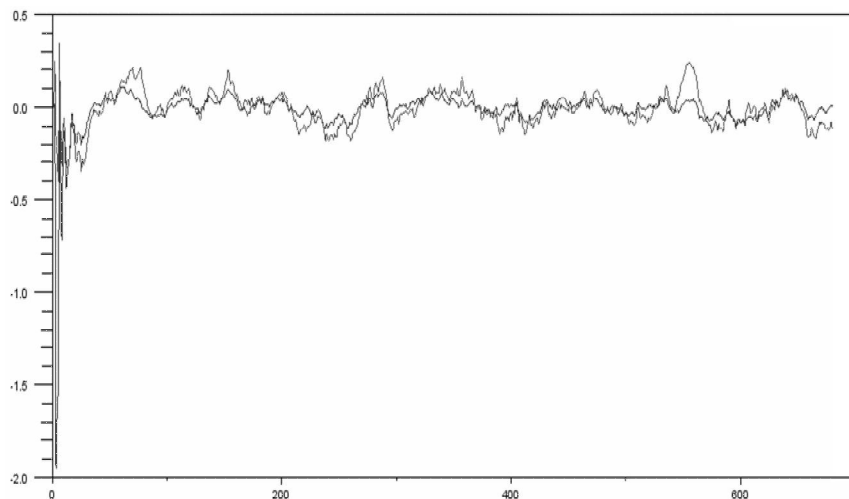


Рис 3. Ошибка вычисления значений скорости и координаты

Заключение

Из полученных результатов можно сделать вывод о том, что представленный алгоритм обработки данных показал пригодный результат для его использования при решении задачи определения положения устройства в пространстве по его датчикам. Для повышения точности работы фильтра можно задать ненулевые, близкие к реальным параметры.

Список литературы

1. Hamilton, W. On quaternions, or on a new system of imaginaries in algebra / W. Hamilton; edited by. D. Wilkins. – Dublin : School of Mathematics Trinity College, 2000. – 92 p.
2. Kalman, R. A New Approach to Linear Filtering and Prediction Problems / R. Kalman // Journal of Basic Engineering. – 2011. – Vol. 82. – P. 3545-3552.
3. Welch, G. An Introduction to the Kalman Filter / G. Welch, G. Bishop. – North Carolina : Chapel Hill, 2006. – 82 p.
4. Shoemake, K. Animating Rotation with Quaternion Curves / K. Shoemake. – San Francisco : SIGGRAPH, 1985. – 254 p.

МЕТОДИКА КОМПЬЮТЕРНОГО АНАЛИЗА ПРОИЗВОДСТВЕННЫХ БИЗНЕС-ПРОЦЕССОВ НА ПРИМЕРЕ СКЛАДА ПРОИЗВОДСТВЕННЫХ РЕСУРСОВ

В. А. Назарьева

Студент

М. Г. Матвеев

Профессор

Введение

Современные представления об управлении предприятием базируются на том, что бизнес необходимо четко определять, измерять, анализировать и улучшать. Неправильное планирование бизнеса может привести к краху всей компании и потере конкурентоспособности. Понять, что именно нужно улучшить – сложно, а проводить эксперименты на реальной компании – дорого.

Для создания эффективной системы управления предприятием в настоящее время достаточно широко используется процессный подход. Процессный подход – это подход к организации и анализу деятельности компании, основанный на выделении и рассмотрении ее бизнес-процессов [1].

Создание модели бизнес-процессов предприятия позволяет осуществить первоначальный анализ его деятельности. Более глубокий анализ бизнес-процессов возможно осуществить, используя методы «имитационного моделирования» и «функционально-стоимостного анализа». Имитационное моделирование позволяет определить время, затрачиваемое на выполнение процесса, функционально-стоимостной анализ (ФСА) направлен на определение стоимости процесса [2].

1. Методика компьютерного анализа

Имитационное моделирование (ИМ) – это метод исследования, основанный на том, что изучаемая система заменяется имитирующей. С имитирующей системой проводят эксперименты (не прибегая к экспериментам на реальном объекте) и в результате получают информацию об изучаемой системе. Метод позволяет имитировать выполнение модели бизнес-процессов так, как оно происходило бы в действительности, с учетом графиков рабочего времени и занятости временных ресурсов и наличия необходимого количества материальных ресурсов [3].

Для определения стоимости процесса совместно с ИМ целесообразно проводить функционально-стоимостной анализ.

Функционально-стоимостной анализ используется для операционно-ориентированного расчета себестоимости продукта (услуги). В основе ФСА лежит положение о том, что для производства продукта (услуги) необходимо выполнить ряд действий, каждое из которых требует определенных ресурсов. Расходы на выполнение каждого действия рассчитываются путем переноса стоимости ресурсов на стоимость действия. Сумма расходов на выполнение каждого действия, с определенными поправками, и будет составлять себестоимость продукта (услуги) [4].

В результате проведения ИМ и ФСА можно:

- оценить средние значения и разброс ключевых параметров процесса;
- найти самые затратные или самые длительные процессы;
- найти "бутылочные горлышки" – перегруженные ресурсы, к которым постоянно выстраивается очередь операций, в результате чего они задерживают выполнение всех процессов;
- найти ресурсы с низкой загрузкой;
- проанализировать производство и потребление материальных ресурсов [2].

Таким образом, ИМ и ФСА позволяют оптимизировать бизнес-процессы предприятия, не проводя эксперименты над компанией и сотрудниками.

Показатели, которые подвергаются оптимизации в первую очередь, это:

- затраты процесса;
- продолжительность процесса.

Неудовлетворительные значения этих показателей снижают эффективность процесса, что приводит к потере денежных средств компании и недовольству руководителя. Высокая стоимость процесса напрямую увеличивает затраты компании. Длительное выполнение процесса увеличивает вероятность того, что его результат будет получен не вовремя и к этому времени уже может быть никому не нужен.

2. Цель эксперимента

Цель эксперимента в данной статье – выполнить компьютерный анализ производственных бизнес-процессов, используя методы имитационного моделирования и функционально-стоимостного анализа, и провести оптимизацию бизнес-процессов с точки зрения улучшения их эффективности.

В качестве примера был рассмотрен небольшой склад. Для обслуживания транспорта, прибывающего на склад, предусмотрено два разгрузочных поста, два грузчика, два кладовщика и три дополнительных парковочных места для парковки машин на территории склада.

Машины с грузом на склад прибывают не в определенное время, а в случайные моменты времени, то есть интервал между прибытием машин является случайной величиной. Количество груза, которое находится в машине – тоже величина случайная, следовательно, время разгрузки товара для каждой прибывшей на склад машины является случайным. Из-за хаотичного прибытия машин на склад, возникает проблема увеличения времени простоя транспорта и неравномерной загрузки персонала склада.

3. Определение бизнес-процессов

Прежде чем провести ИМ и ФСА, необходимо было построить диаграмму процесса работы склада в нотации EPC (рис. 1).

Нотация EPC (Event-Driven Process Chain – событийная цепочка процессов) используется для описания процессов нижнего уровня. Данная нотация используется для представления алгоритма выполнения процесса.

Диаграмма процесса в нотации EPC представляет собой упорядоченную комбинацию событий и функций. Для каждой функции могут быть определены начальные и конечные события, участники, исполнители, материальные и документальные потоки, сопровождающие её [5].

Возникновение стартового события "Приехала машина с товаром" запускает последовательное выполнение шагов процесса "Разгрузка и обслуживание машин на складе": завершение одного шага приводит к началу выполнения следующего. Однако на диаграмме присутствуют точки ветвления, в которых системе необходимо принять решение, по какому пути продолжить выполнение процесса.

Для всех процессов, расположенных на диаграмме, необходимо было заполнить время их выполнения, так как количество разгруженных машин на складе напрямую зависит не только от времени разгрузки одной машины, а также от времени выполнения каждого шага бизнес-процесса. Время выполнения шага процесса может быть фиксированной или случайной величиной.

Длительность шага процесса зависит не только от нормативного времени его выполнения, но и от доступности ресурсов (временных и материальных), необходимых для его выполнения. Если необходимых ресурсов в нужный момент времени нет, то выполнение процесса останавливается. Поэтому для каждого шага процесса необходимо было заполнить перечень и количество необходимых ресурсов для его выполнения.

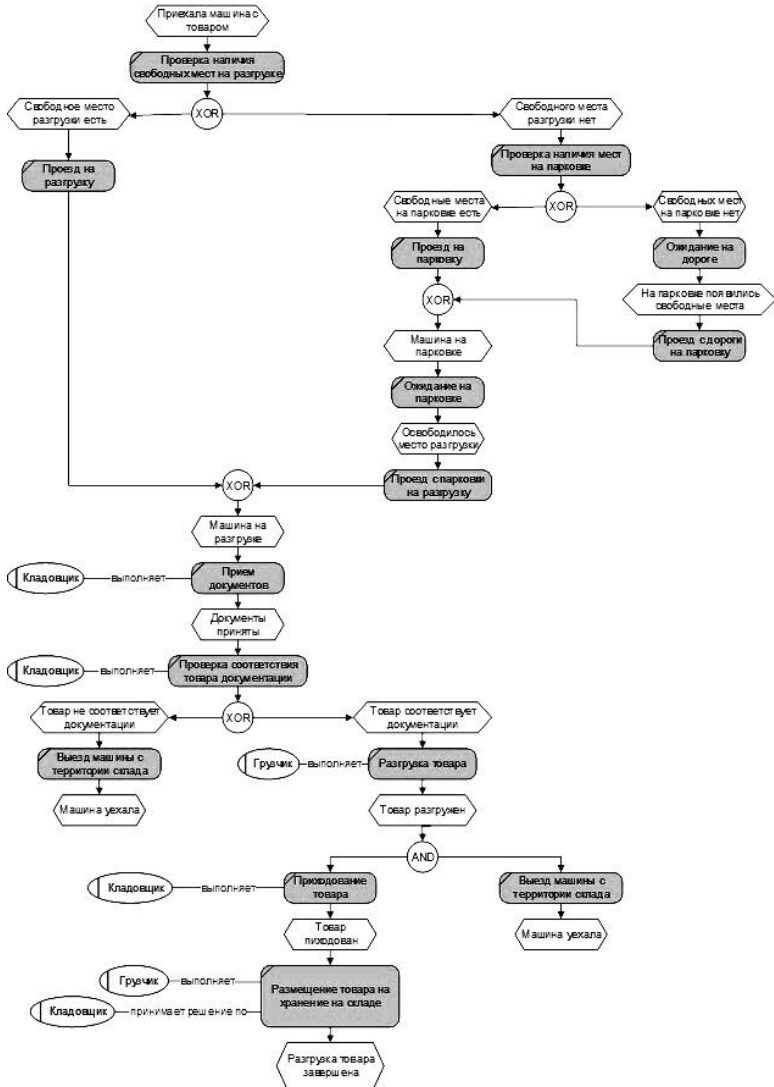


Рис. 1. Диаграмма процесса «Разгрузка и обслуживание машин на складе»

4. Результаты имитации

Проведя имитацию работы склада с заданными параметрами в течение одного рабочего дня, мы получили следующие показатели процесса (рис. 2).

| Процесс | Количество выполнений | Средняя длительность |
|--|-----------------------|----------------------|
| ТП2.1 Проверка наличия свободных мест на разгрузке | 12 | 0:05:00 |
| ТП2.2 Проверка наличия мест на парковке | 8 | 0:05:00 |
| ТП2.3 Проезд на разгрузку | 4 | 0:05:00 |
| ТП2.4 Проезд на парковку | 7 | 0:05:00 |
| ТП2.5 Ожидание на дороге | 1 | 0:46:00 |
| ТП2.6 Проезд с дороги на парковку | 1 | 0:35:00 |
| ТП2.7 Ожидание на парковке | 8 | 0:42:00 |
| ТП2.8 Проезд с парковки на разгрузку | 5 | 0:46:00 |
| ТП2.9 Прием документов | 8 | 0:42:30 |
| ТП2.10 Проверка соответствия товара документации | 7 | 0:10:00 |
| ТП2.11 Выезд машины с территории склада | 1 | 0:05:00 |
| ТП2.12 Разгрузка товара | 6 | 0:46:40 |
| ТП2.13 Приходование товара | 6 | 0:25:00 |
| ТП2.14 Выезд машины с территории склада | 6 | 0:05:00 |
| ТП2.15 Размещение товара на хранение на складе | 5 | 1:13:00 |

Рис 2. Параметры шагов процесса, полученные в результате имитации

Анализируя результаты имитации можно сделать вывод о том, что склад работает неэффективно:

- из 12 машин было обслужено всего 6;
- товар, привезенный только 5-ю машинами, был размещен на хранение на складе;
- из 12 машин 8 ожидали на парковке (среднее время ожидания 42 минуты);
- в среднем на обслуживание одной машины было потрачено 230 руб.;
- машина, которая прибыла на склад в первые часы его работы, была обслужена за 1 час, а машина, которая простояла в очереди на разгрузку, была обслужена только через 4 часа 50 минут.

Таким образом, в конце рабочего дня 5 машин остались не обслуженными, из них 3 машины ожидают на парковке, 2 машины ожидают обслуживания на разгрузке. По предварительной договоренности с поставщиками товара владелец склада должен заплатить штраф в размере 500 руб. водителю машины, в том случае, если машина приехала на склад, но не была обслужена. С учетом этого, стоимость процесса будет составлять 4 271 руб. в день, а содержание склада за месяц владельцу обойдется примерно в 100 000 руб., за год – в 1 200 000 руб.

Очевидно, что такие показатели работы склада недопустимы.

5. Оптимизация процесса

Выходом из сложившейся ситуации может быть решение: переоборудовать одно парковочное место под место разгрузки и добавить для его обслуживания одного грузчика. Для того чтобы сократить время

размещения товара на хранение на складе, купить один погрузчик. Очевидно, что оставлять недоделанную работу на следующий рабочий день нецелесообразно, учитывая интенсивность ежедневного потока поступления машин на склад, увеличить рабочее время сотрудников склада.

После проведения оптимизации показатели работы склада значительно улучшились:

- все прибывшие на склад машины были разгружены;
- весь товар, привезенный машинами, был размещен на складе;
- снизилось время ожидания машины на парковке до 25 минут;

После оптимизации стоимость процесса стала составлять 2 688 руб. в день, а, следовательно, примерно 60 тыс. руб. в месяц и 720 тыс. руб. в год.

Заключение

Данная статья посвящена рассмотрению методики компьютерного анализа производственных бизнес-процессов. В ходе эксперимента была рассмотрена работа данной методики на примере склада производственных ресурсов. В итоге, имитационное моделирование и функционально-стоимостной анализ позволили проанализировать существующие бизнес-процессы, увидеть их слабые стороны и провести оптимизацию бизнес-процессов с точки зрения улучшения их эффективности.

Таким образом, использование имитационного моделирования и функционально-стоимостного анализа позволили оптимизировать бизнес-процессы предприятия, не проводя эксперименты над компанией и сотрудниками.

Список литературы

1. Вилкова, Л. Менеджмент процессов / Л. Вилкова, В. Таратухина; под ред. Й. Беккер. – Москва : Эксмо, 2007. – 384 с.
2. Имитационное моделирование и функционально-стоимостной анализ. Методика / Группа компаний «Современные технологии управления». – Самара : Альфа-Пресс, 2010. – 81 с.
3. Лукасевич, И. Я. Имитационное моделирование инвестиционных рисков / И. Я. Лукасевич. – Москва : ЮНИТИ, 2006. – 105 с.
4. Афанасьев, А. В. Определение себестоимости методом Activity-Based Costing / А. В. Афанасьев // Финансовый директор. – 2003. – №7. – С. 35-37.
5. Описание нотации EPC / Группа компаний «Современные технологии управления». – Самара : Альфа-Пресс, 2010. – 5 с.

ЗАЩИТА ЦИФРОВЫХ ИЗОБРАЖЕНИЙ ОТ НЕСАНКЦИОНИРОВАННОГО ИСПОЛЬЗОВАНИЯ С ИСПОЛЬЗОВАНИЕМ «САМОРЕКОНСТРУИРУЮЩЕЙ» СТЕГАНОГРАФИИ

М. Е. Ольшанская

Студент

М. А. Дрюченко

Доцент

Введение

Для противодействия возможному злоупотреблению, связанным с умышленным разрушением, кражей, несанкционированным доступом, чтением, копированием и тиражированием информации, необходимо грамотно выбирать меры и средства обеспечения защиты информации, а также хорошо знать административные и законодательные меры в этой области. На сегодняшний день для защиты информации в компьютерных системах чаще всего применяются программно-аппаратные средства, использующие современные криптографические методы, позволяющие эффективно решать вопросы безопасности, связанные с обеспечением конфиденциальности и целостности данных. Вместе с тем, для отдельных приложений информационной безопасности, таких как скрытая коммуникация, защищенное скрытое хранение информации, защита авторских прав и контроль незаконного тиражирования цифровых объектов, традиционных криптографических методов бывает недостаточно. Для этих целей целесообразно использовать стеганографические методы [1]. Если рассматривать классические схемы создания невидимых цифровых водяных знаков (ЦВЗ) [2], то они позволяют решить указанные выше проблемы частично, оставляя злоумышленникам возможность беспрепятственного использования цифрового контента без его размещения и тиражирования в сети. К примерам подобных незаконных вариантов использования цифровых изображений можно отнести просмотр и использование незаконно полученных авторских изображений или изображений конфиденциального характера, продажу объектов интеллектуальной собственности, промышленный шпионаж.

Одним из возможных перспективных подходов к развитию технологий контроля использования и защиты прав авторства на цифровые изображения (и любые объекты графического контента, обладающие

визуальной избыточностью), является подход, основанный на внесении значительных видимых искажений во фрагменты растра с одновременным стеганографическим кодированием исходного представления искаженных фрагментов в нетронутые области растра. Искажение изображения приводит к невозможности нормального его использования неавторизованными лицами. Для восстановления исходного представления растра применяется обратное стеганографическое преобразование. Важно также отметить, что создаваемые искажения не должны поддаваться компенсации с использованием известных алгоритмов восстановления испорченных областей изображения, наподобие алгоритмов Inpaint [3].

1. Алгоритм «самореконструирующей» стеганографии

На рис. 1 представлена обобщенная блок-схема реализации алгоритма внесения искажений параллельно с использованием процедуры стеганографического кодирования данных.

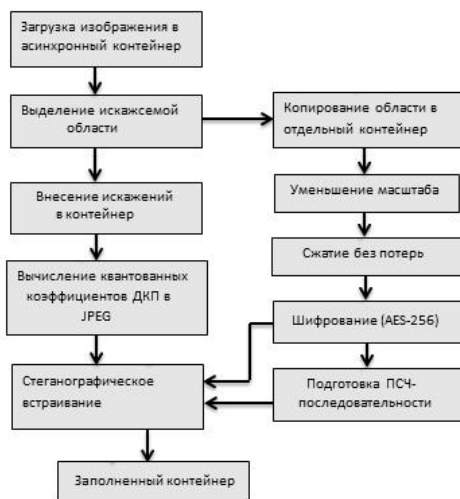


Рис. 1. Обобщенная схема предлагаемого процесса защиты изображений

Термин «самореконструкция» можно определить следующим образом. Это процесс восстановления исходного или приближенного к исходному представлению цифрового объекта на основе его текущего (отличного от исходного) представления.

Сформулируем в общем виде постановку задачи стеганографического кодирования/декодирования искаженных фрагментов изображения. Пусть I – исходное изображение размером $W \times H$ пикселей, $I_{xy}^{(b)} \subset I$ –

прямоугольный искажаемый фрагмент размером $w \times h$, $w \ll W$, $h \ll H$, имеющий смещение на исходном растре $x = \overline{1, W - w}$, $y = \overline{1, H - h}$. Сначала к I применяется процедура внесения визуальных искажений

$$\tilde{I} = G(I, x, y, w, h). \quad (1)$$

Вносимые в I искажения в простейшем случае могут быть реализованы однородной заливкой заданным цветом выделенной области $I_{xy}^{(b)}$. Значения пикселей вне искажаемой области не модифицируются

$$\tilde{I}_{i,j} = I_{i,j}, \quad i \notin [x, x + w] \vee j \notin [y, y + h].$$

Далее в искаженное изображение \tilde{I} с использованием ключа k реализуется процедура стеганографического встраивания исходного представления испорченного фрагмента $I_{xy}^{(b)}$

$$J = F_1(\tilde{I}, I_{xy}^{(b)}, k), \quad (2)$$

в результате чего формируется «защищенное» изображение J , содержащее искаженную область, не поддающуюся исправлению без знания ключа k .

Для восстановления первоначального представления изображения I' применяется процедура стеганографического декодирования вида

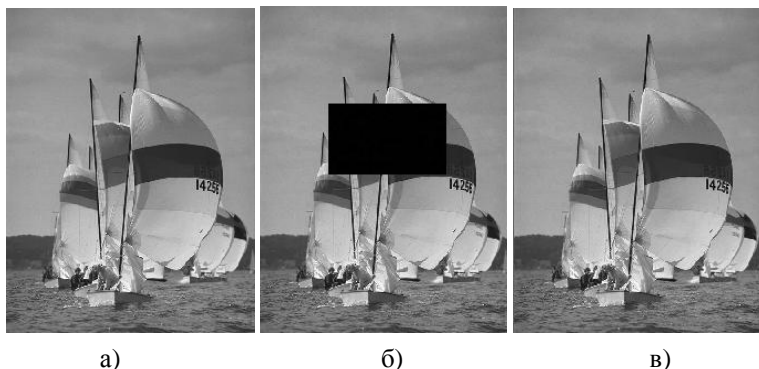
$$I' = F_2(J, k). \quad (3)$$

Для минимизации визуальных искажений требуется обеспечить выполнение условия $\|I - I'\| \rightarrow \min$, то есть «реконструированное» изображение должно минимальным образом отличаться от исходного. Также, для повышения надежности схемы защиты в части противодействия попыткам возможной компенсации искажений Inpaint-подобными алгоритмами, необходимо максимизировать отношение площадей искажаемой области к общей площади исходного раstra

$\frac{S(I_{xy}^{(b)})}{S(I)} = \frac{wh}{WH} \rightarrow \max$. Пример искаженного заполненного и реконструированного контейнеров приведен на рис. 2.

В блоке «Уменьшение масштаба» на схеме (рис. 1) с использованием алгоритма билинейной интерполяции реализуется уменьшение выделенного фрагмента $I_{xy}^{(b)}$ (в исходном его представлении). При небольших значениях коэффициента масштабирования данный процесс эквивалентен сжатию фрагмента с незначительными потерями качества. Сжатие данных потенциально позволяет максимизировать коэффициент эффективного использования контейнера, выражаемый отношением

размера наибольшего сообщения, которое возможно скрыть в контейнер к объему последнего.



а) – исходное, б) – искаженное заливкой, в) – «реконструированное»

Рис. 2. Тестовое изображение «sailboats under spinnakers.jpg»

В качестве базового стеганографического алгоритма, удовлетворяющего требованиям строгого (не вероятностного) характера работы процедуры извлечения информации F_2 и высокой пропускной способности алгоритма, позволяющей «реконструировать» большие области контейнера, рассматривался алгоритм модификации наименее значащих бит (НЗБ) квантованных коэффициентов дискретного косинусного преобразования (ДКП) JPEG [4]. На фотографических изображениях данный алгоритм позволяет эффективно скрывать массивы данных, соответствующие искажаемым фрагментам площадью до 10-15% от площади незаполненного графического контейнера.

Для повышения стойкости предлагаемой схемы защиты изображений к возможным атакам злоумышленника, направленным на восстановление искаженных областей графического контейнера, помимо стеганографических преобразований целесообразно применять криптографические методы. В реализованном тестовом программном модуле самореконструирующей стеганографии перед стеганографическим встраиванием исходное (в уменьшенном масштабе) представление искажаемого фрагмента раstra $I_{xy}^{(b)}$ шифруется с использованием блочного симметричного алгоритма AES (размер ключа 128 бит). Также в процессе стеганографического кодирования информации элементы носителя выбираются в порядке, определяемом псевдослучайной числовой последовательностью (ПСЧП), зависящей от введенных пользователем секретных данных.

Начальное состояние генератора ПСЧП, а также ключ шифрования формируются на основе текстовой строки пароля и «соли» – параметров, задаваемых пользователем. Для выработки ключа на основе строкового парольного значения используется алгоритм PBKDF2 [5].

2. Исследование эффективности алгоритма

Для оценки искажений использовался объективный критерий, основанный на сравнении исходного и преобразованного изображения – среднеквадратическая ошибка (MSE):

$$MSE_{I,I'} = (wh)^{-1} \sum_{i=1, j=1}^{w,h} (I_{i,j} - I'_{i,j})^2,$$

где I – оригинальное изображение; I' – реконструированное; w и h – ширина и высота I, I' ; Max_I – максимальное значение цвета изображения I . На рис. 3 приведены зависимости MSE от размера искажаемой области $I_{xy}^{(b)}$.

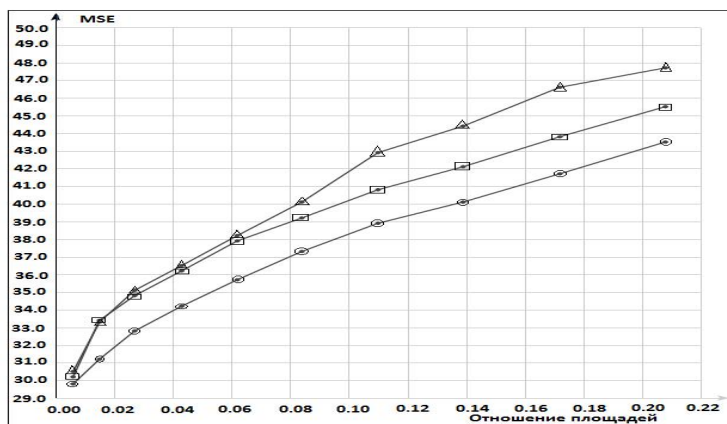


Рис. 3. Зависимость MSE от отношения площадей искажаемого фрагмента и контейнера

На графике представлены зависимости, полученные для тестовых изображений трех типов, определяемых характером их содержимого. Первый тип (треугольные маркеры) соответствует контрастным изображениям с большим числом мелких деталей, второй (круглые маркеры) – гладким изображениям, содержащим большие монотонные области плавного перехода цветов, третий (квадратные маркеры) – изображениям смешанного типа.

Как показали результаты экспериментов с увеличением площади искажаемой области MSE искажения будет возрастать приблизительно линейно. Для реализованного стегаалгоритма приемлемое значение MSE при работе с jpeg контейнерами достигается, когда значение отношения площадей составляет порядка 10-15%. Так как именно при этом показателе достигается максимально эффективное скрытие, т. е. выделяется довольно большая область искажения, но это не вызывает значительного падения качества восстановленного изображения.

Заключение

Данная статья посвящена описанию нового подхода к защите цифровых изображений, основанного на ограничении их функциональности за счет внесения не поддающихся компенсации видимых искажений, с последующим использованием принципов самореконструирующей стеганографии для восстановления приближенного к исходному представлению испорченных областей. Среди преимуществ предложенного подхода отметим следующие:

- отсутствие у неавторизованных лиц возможности полноценного использования контента без знания секретных ключей (соответствующих разрешений со стороны правообладателя);
- отсутствие необходимости хранения исходных копий неискаженных изображений или видео файлов;
- возможность предварительного ознакомления потенциальных пользователей с «частично испорченным» контентом с перспективой дальнейшего его приобретения;
- относительная простота реализации.

Список литературы

1. Грибунин, В. Г. Цифровая стеганография / В. Г. Грибунин, И. Н. Оков. – Москва : Солон-пресс, 2002. – 272 с.
2. Конахович, Г. Ф. Компьютерная стеганография. Теория и практика / Г. Ф. Конахович, А. Ю. Пузыренко. – Киев : МК-Пресс, 2006. – 288 с.
3. Oliveira, M. Fast Digital Image Inpainting / M. Oliveira, B. Bowen, R. McKenna, Yu-Sung Chang // International Conference on Visualization, Imaging and Image Processing (VIP 2001), Marbella, Spain. 2001.
4. Provos, N. Hide and seek: An introduction to steganography / N. Provos, P. Honeyman // IEEE Security Privacy. – 2003. – P. 32-44.
5. PKCS #5: Password-Based Cryptography Specification. Version 2.0 [Электронный ресурс]. – Режим доступа : <http://www.rfc-base.org/txt/rfc-2898.txt>

ПРОГРАММНАЯ ОБОЛОЧКА ДЛЯ РАСПОЗНАВАНИЯ ЛИЦ ПО 2D-СНИМКУ

Н. Н. Осолодков

Студент

С. А. Запрягаев

Профессор

Введение

Идентификация человека по изображению лица – одна из самых актуальных технологий компьютерного зрения благодаря востребованности в области систем безопасности и поиска фотографий в сети Интернет.

Список применений данной технологии включает в себя:

- Определение личности по фотографии;
- Подтверждение личности по биометрическому паспорту;
- Использование в системах контроля и управления доступом;
- Поиск человека в видеоархиве системы видеонаблюдения.

Множество исследований, проводящихся в ведущих научных центрах всего мира в течение нескольких десятилетий, так и не привело к созданию реально работающих систем компьютерного зрения, способных обнаруживать и распознавать человека в любых условиях. В данной статье рассматривается система распознавания лиц, состоящая из камеры и программной оболочки, позволяющая по 2D-фотоснимку идентифицировать человека, если он заведомо присутствует в базе известных лиц. В последующем, разработанную систему планируется поддерживать усовершенствовать, а так же иметь возможность внедрять ее в различные области применения.

1. Элементы теории распознавания образов

Классическая постановка задачи распознавания образов: дано множество объектов. Относительно них необходимо провести классификацию. Множество представлено подмножествами, которые называются классами. Заданы: информация о классах, описание всего множества и описание информации об объекте, принадлежность которого к определенному классу неизвестна. Требуется по имеющейся информации о классах и описании объекта установить, к какому классу относится этот объект [1].

Наиболее часто в задачах распознавания образов рассматриваются монохромные изображения, что дает возможность

рассматривать изображение как функцию на плоскости. Если рассмотреть точечное множество на плоскости T , где функция $f(x, y)$ выражает в каждой точке изображения его характеристику — яркость, прозрачность, оптическую плотность, то такая функция есть формальная запись изображения.

Множество же всех возможных функций $f(x, y)$ на плоскости T — есть модель множества всех изображений X . Вводя понятие *сходства* между образами можно поставить задачу распознавания. Конкретный вид такой постановки сильно зависит от последующих этапов при распознавании в соответствии с тем или иным подходом [1].

2. Описание программной оболочки

Описываемая в статье программная оболочка позволяет захватывать видеопоток с Web-камеры, получать снимок лица, обрабатывать его и подавать на вход классификатора. По результатам работы определять принадлежность входного лица к базе заведомо известных людей. Интерфейс программной оболочки изображен на рисунке.

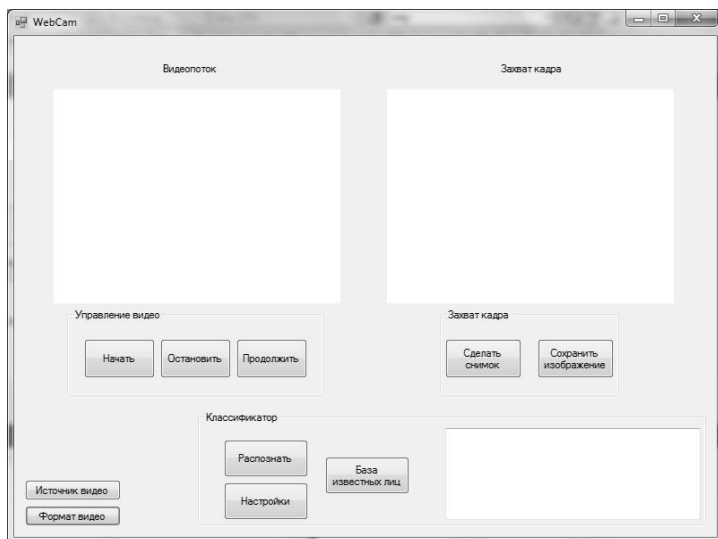


Рисунок. Интерфейс реализованной программной оболочки

Приложение написано на языке программирования C# с использованием внешней библиотеки *WebCam_Capture.dll*, позволяющей работать с Web-камерой. В левой части окна в области видеопотока после выбора источника видео отображается видео-картинка в режиме

непрерывного времени. При нажатии кнопки «Остановить» показ видео останавливается. Продолжить показ можно нажатием кнопки «Продолжить». В правой части окна в области захвата кадра расположены элементы управления, отвечающие за фиксацию кадра и возможности сохранения его в формате .jpg с разрешением 320 x 240 пикселей на компьютер. Так же имеется окно просмотра полученного снимка. Программный модуль позволяет выбирать источник видеосигнала, а так же настраивать основные параметры видео, такие как разрешение, глубина цвета, сжатие и др. Есть возможность управлять настройками камеры. Данные опции доступны при нажатии кнопок «Формат видео» и «Источник видео».

3. Подготовка обучающих данных

Для подготовки обучающей базы изображений лиц были привлечены 20 человек. С помощью описанного выше программного модуля и Web-камеры Defender Multicam WF-10HD была получена база цветных фотоснимков из 20 лиц разных людей по 10 снимков лица каждого человека. При создании 10 вариантов каждого лица были зафиксированы различные варианты освещения, выражения лица, небольшой наклон и поворот головы, а так же несущественные изменения в причёске. Существенных изменений в образы человека не вносилось, так как методы распознавания по 2D-снимку лица нестойки к сильному видоизменению лица.

В дальнейшем были подготовлены обучающие выборки из полученной базы фотоснимков. Для существенного уменьшения объема обрабатываемых данных было принято решение преобразовать цветные изображения в монохромные, так как каждый пиксель цветного изображения кодируется тремя компонентами R, G, B, а пиксель монохромного – одной компонентой.

Для обесцвечивания изображений был реализован алгоритм на языке C# в данной программной оболочке. Перебор массива байт изображения осуществляется с шагом 3, поскольку каждый пиксель кодируется 3-мя байтами R G B. В цикле составляющие R G и B суммируются весами 0.21, 0.72 и 0.07 соответственно. Вычисляется среднее значение из трёх составляющих, после чего вместо значений R G и B помещается это среднее значение, таким образом, задаётся необходимый тон серого.

Результатом работы алгоритма является обесцвеченное фото. Данный переход необходим для уменьшения размерности векторов-признаков образов, так как алгоритм распознавания основан на применении многослойной нейронной сети [2]. При этом потеря качества распознавания несущественна, а затраты на обучение нейронной сети и время работы конечной системы значительно сокращаются.

4. Создание и обучение нейронной сети

После перевода всех цветных снимков лиц базы в монохромные необходимо подготовить векторы признаков для подачи их на входные нейроны сети [3]. Так как полученный снимок представляет собой матрицу размерности 320 x 240 пикселей, элементы которой есть десятичные числа от 0 до 255, то необходимо развернуть ее в вектор. Данная функция реализована в программной оболочке. С ее помощью для каждого черно-белого снимка лица создается файл с расширением .txt, в котором в виде столбца записаны десятичные значения всех $320 \cdot 240 = 76800$ пикселей. Данные текстовые файлы в дальнейшем необходимы для загрузки обучающей выборки в программный продукт STATISTICA 10 (64 bit) с пакетом *STATISTICA Automated Neural Networks (SANN)*, в котором, собственно, будут созданы и обучены нейронные сети для текущей задачи [2].

При выборе данных для анализа, в качестве категориальной целевой переменной указывается переменная с метками классов образов, т. е. с именами людей. В качестве непрерывной входной переменной выбираются все вектор-столбцы признаков. В работе использовалась стратегия построения автоматизированной нейронной сети (НС). Для обучения и тестирования модели выбираются коды для подвыборок: обучающей и контрольной. 70% подготовленных для обучения данных будет использоваться для обучения сети, а оставшиеся 30% – для контроля адекватности работы построенной модели. Во вкладке «Подвыборки (ПНС и АНС)» выбираются соответствующие данные для кодов обучающей и контрольной подвыборки. Далее создаются 5 сетей на различных подвыборках для последующего их анализа и выбора сети с наилучшими показателями [3]. Задаются минимальные настройки: архитектура сети – многослойный персептрон, функции активации нейронов скрытого и выходного слоя – логистические и гиперболические (так как они используются в задачах классификации), количество нейронов скрытого слоя [2]. Использовалась функция затухающей регуляризации весов для скрытых слоев с параметрами по умолчанию [3].

По завершении обучения имеется 5 обученных моделей сетей на архитектуре многослойного персептрона. Проанализируем полученные модели с помощью встроенных средств «Итоги моделей», «Матрица ошибок». Была выбрана сеть с максимальным значением производительности обучения и контрольной производительности, которые составили 98,75% и 97,14% соответственно. В качестве алгоритма обучения автоматически был выбран алгоритм Бройдена-Флетчера-Гольдфарба-Шанно (BFGS) [2]. Функция активации скрытых и выходных слоев – гиперболическая [3].

После выбора наилучшей сети выполнено ее сохранение. В пакете STATISTICA 10 присутствует опциональная возможность генерации исходного кода на языках C, C++, C#, Java, PMML. Так как нам необходимо встроить обученную модель в программную оболочку, реализованную на языке C#, было выполнено сохранение сети в соответствующем формате. Далее созданные классы с обученной нейронной сетью были включены в приложение.

5. Тестирование

Разработанный программный модуль, дополненный классами с обученной нейронной сетью, готов к тестированию и дальнейшей эксплуатации. Для тестирования классификатора достаточно захватить кадр с лицом распознаваемого человека и нажать кнопку «Распознать» (см. рисунок). Снимок лица автоматически преобразуется в монохромный вариант, матрица пикселей изображения разворачивается в вектор и сохраняется в одномерном массиве длины $N \times M$ (текущее разрешение снимка). Данный массив подается в качестве аргумента методу, который в свою очередь работает с классами нейронной сети. Таким способом, вычисляется вероятность того, к какому классу принадлежит входной образ. Окончательный выбор производится по правилу «Победитель получает все!» [3], т. е. тот класс, вероятность принадлежности к которому наибольшая, и является выходным классом.

В случае же, если лицо не опознано, будет выведено соответствующее сообщение. Принятие решения о том, что образ не принадлежит ни одному из зарегистрированных классов, осуществляется на основе сравнения максимальной вероятности совпадения с каким-либо классом с пороговым значением: если максимальная вероятность совпадения с каким-либо классом меньше пороговой, то принимается решение о том, что данного образа нет в базе.

В качестве порога экспериментальным путем было выбрано значение 0.75 или 75% совпадения входного образа с эталонами. Данный подход позволяет практически безошибочно распознавать входные образы, либо делать вывод о том, что такой образ не зарегистрирован в базе.

При успешной идентификации человека в поле правого нижнего угла будет выведена вся имеющаяся информация по данной персоне (см. рисунок). Если распознавание прошло неуспешно или же входной образ не найден в базе, то на экран будет выведено соответствующее сообщение.

Для добавления новых людей в базу образов необходимо обучать нейронную сеть с учетом новых признаков и классов.

6. Оценка системы

Для оценки адекватности работы полученного алгоритма воспользуемся некоторыми устоявшимися понятиями из биометрии: FAR, FRR.

Вероятность ложного недопуска (ошибка первого рода) – **False Rejection Rate (FRR)** – для зарегистрированных в системе людей определяет процент случаев отказа в доступе.

Сначала нужно посчитать FRR для каждого пользователя в отдельности:

$$FRR(n) = \frac{n}{N}, \quad (1)$$

где n – число запрещенных попыток доступа зарегистрированному лицу; N – общее число попыток.

После этого FRR нужно усреднить по всей базе для расчета общего значения:

$$FRR = \frac{1}{N} \sum_{n=1}^N FRR(n). \quad (2)$$

Вероятность ложного допуска (ошибка второго рода) – **False Acceptance Rate (FAR)** – вероятность допуска, не зарегистрированного в системе человека.

Вероятность ложного распознавания как человека, данные которого имеются в базе [FAR(n)], может быть рассчитана по формуле:

$$FAR(n) = \frac{n}{N}, \quad (3)$$

где n – число успешных независимых попыток, когда система предоставляет доступ незарегистрированному пользователю; N – общее число независимых попыток.

Чем больше попыток будет сделано, тем статистически надежнее результаты будут получены. Общий FAR рассчитывается по (4) как арифметическое среднее всех персональных характеристик:

$$FAR = \frac{1}{N} \sum_{n=1}^N FAR(n). \quad (4)$$

Система признается тем стабильнее и надежнее, чем меньше значение FRR при одинаковых значениях FAR.

Экспериментальным путем были вычислены следующие оценки разработанной системы: FAR = 0.05%; FRR = 1.1%. Для более наглядной интерпретации полученных результатов можно оценить, как часто будут возникать ложные совпадения, если установить систему идентификации на

проходной организации с численностью персонала N человек. Вероятность ложного совпадения полученного камерой и программным модулем для базы данных из N человек равна $FAR \cdot N$.

Каждый день через пункт контроля доступа проходит тоже порядка N человек. Тогда вероятность ошибки за рабочий день $FAR \cdot (N \cdot N)$. Конечно, в зависимости от целей системы идентификации вероятность ошибки за единицу времени может сильно варьироваться, но если принять допустимым одну ошибку в течение рабочего дня, то:

$$FAR \cdot N^2 \approx 1 \Rightarrow N \approx \sqrt{\frac{1}{FAR}}. \quad (5)$$

Таким образом, подставив значение $FAR = 0.05\%$ в (5), получим, что стабильная работа системы идентификации при данном значении FAR возможна при численности персонала $N \approx 45$. При этом средняя точность распознавания составила 85-90%

Заключение

В рамках данной работы была реализована система для распознавания и идентификации человека по снимку лица.

Разработанный программный модуль обладает следующими показателями:

- Точность распознавания составляет 85-90%.
- Имеется возможность добавления (удаления) образов в базу лиц.
- Работа с любыми WEB- IP-камерами.
- Возможность интеграции в различные проекты, такие как системы контроля и управления доступом (СКУД), электронные карточки.

Список литературы

1. Патрик, Э. А. Основы теории распознавания образов : учебник / Э. А. Патрик; пер. с англ. В. М. Баронкина, Б. А. Смиренина, Ю. С. Шинакова; под ред. Б. Р. Левина. – Москва : Советское радио, 1980. – 407 с.
2. Хайкин, С. Нейронные сети. Полный курс : учебник / С. Хайкин. – 2-е изд. – Москва : Вильямс, 2006. – 1104 с.
3. Боровиков, В. П. Популярное введение в современный анализ данных в системе STATISTICA : учебник / В. П. Боровиков. – Москва, 2013. – 288 с.

КОМПЬЮТЕРНЫЕ АЛГОРИТМЫ ОПИСАНИЯ АФФИННО-ОДНОРОДНЫХ ПОВЕРХНОСТЕЙ $(1/2, \varepsilon)$ -ТИПОВ В ПРОСТРАНСТВЕ \mathbb{C}^3

Е. А. Павлова

Студент

А. В. Лобода

Профессор

Введение

В данной работе обсуждаются вещественно-аналитические гиперповерхности в трехмерном комплексном пространстве \mathbb{C}^3 , однородные относительно аффинных преобразований этого пространства.

Будем считать, что уравнения обсуждаемых поверхностей имеют следующий вид (см. [1]):

$$v = (|z_1|^2 + |z_2|^2) + (\varepsilon_1 z_1^2 + \varepsilon_2 z_2^2) + \overline{(\varepsilon_1 z_1^2 + \varepsilon_2 z_2^2)} + i\alpha(z_1 - \bar{z}_1)u + F_3^{(0)} + \sum_{k \geq 4} F_k(z, \bar{z}, u), \quad (1)$$

где z_1, z_2, w - комплексные переменные, $u = \operatorname{Re} w, v = \operatorname{Im} w, \alpha \in \{0, 1\}$.

В работе рассматривается случай, для которого $\varepsilon_1 = \frac{1}{2}, \varepsilon_2 = \varepsilon$, где $0 < \varepsilon \neq \frac{1}{2}$. Поверхности (1), удовлетворяющие этим условиям, будем называть поверхностями $(\frac{1}{2}, \varepsilon)$ -типов.

Понятие однородности обычно связывают с группами Ли. При этом группе Ли аффинных преобразований однородной поверхности M соответствует алгебра Ли $\mathfrak{g}(M)$ аффинных векторных полей в пространстве \mathbb{C}^3 . Все поля из $\mathfrak{g}(M)$ являются касательными к обсуждаемой аффинно-однородной вещественной гиперповерхности M ; алгебру Ли $\mathfrak{g}(M)$ фактически можно считать матричной алгеброй (см. [1]).

Предметом данной работы является изучение матричных алгебр Ли, отвечающих однородным поверхностям $(\frac{1}{2}, \varepsilon)$ -типов, на основе использования пакета символьных вычислений Maple и компьютерных алгоритмов.

1. Постановка конкретной задачи

Определение. Матричная алгебра Ли $g(M)$ – это линейное пространство матриц, замкнутое относительно дополнительной операции антикоммутативного (скобочного) умножения:

$$[E_k, E_l] = E_k E_l - E_l E_k, \quad (2)$$

В задаче об однородных поверхностях важным является вопрос о возможных размерностях обсуждаемых алгебр Ли.

Теорема 1 (Павлова Е.А., 2014 г.). Вещественная размерность алгебры Ли векторных полей, касательных к аффинно-однородной поверхности типа $\left(\frac{1}{2}, \varepsilon\right)$, удовлетворяет ограничениям

$$5 \leq \dim_R g(M) \leq 6.$$

Случай 6-мерных алгебр Ли полностью описан в работе [2].

В данной работе рассматривается пятимерный случай, являющийся более сложным, а потому основным для поверхностей типов $\left(\frac{1}{2}, \varepsilon\right)$.

Базис 5-мерной алгебры Ли в этом случае имеет следующий специальный вид (см. [1]):

$$E_1 = \begin{pmatrix} A1_1 & A2_1 & A3_1 & 1 \\ B1_1 & B2_1 & B3_1 & 0 \\ 4i & 0 & 2m_1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, E_2 = \begin{pmatrix} A1_2 & A2_2 & A3_2 & i \\ B1_2 & B2_2 & B3_2 & 0 \\ 0 & 0 & 2(m_2 - ia) & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix},$$

$$E_3 = \begin{pmatrix} A1_3 & A2_3 & A3_3 & 0 \\ B1_3 & B2_3 & B3_3 & 1 \\ 0 & 2i(1 + 2\varepsilon) & 2m_3 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, E_4 = \begin{pmatrix} A1_4 & A2_4 & A3_4 & 0 \\ B1_4 & B2_4 & B3_4 & i \\ 0 & 2(1 - 2\varepsilon) & 2m_4 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix},$$

$$E_5 = \begin{pmatrix} A1_5 & A2_5 & A3_5 & 0 \\ B1_5 & B2_5 & B3_5 & 0 \\ -a & 0 & 2(m_5 + i\lambda) & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad (3)$$

Замечание 1. Элементы $A1, A2, A3, B1, B2, B3$ данных матриц являются комплексными числами, а элементы $m_1 - m_5, \lambda$ – вещественные.

Замечание 2. Далее мы полагаем $\alpha = 0$. Случай ненулевого α , как показано в [3], для обсуждаемых поверхностей невозможен. В то же время пример однородной поверхности типа $\left(\frac{1}{2}, \frac{1}{2}\right)$ с $\alpha = 1$ имеется в [4].

Требуется описать все матричные алгебры Ли с базисами вида (3) с учетом двух данных замечаний.

2. «Арифметика» задачи

Так как структура алгебры Ли означает требование замкнутости относительно операции антикоммутиративного (скобочного) умножения, то для любой пары базисных матриц $E_k \neq E_l$ должно выполняться равенство

$$[E_k, E_l] = r_1 E_1 + r_2 E_2 + r_3 E_3 + r_4 E_4 + r_5 E_5 \quad (4)$$

для некоторых вещественных r_1, \dots, r_5 .

Стандартная схема исследования системы таких равенств и, как следствие, однородности подробно описана в [1], [5]-[6]. Для случая

однородной поверхности типа $\left(\frac{1}{2}, \varepsilon\right)$, как и для многих других типов,

основной интерес в этой схеме представляет система, состоящая из 25 квадратичных уравнений относительно 16 неизвестных параметров. Этими параметрами являются вещественные и мнимые части некоторых матричных элементов из базиса (3). Отметим также, что имеется замена переменных

$$z_1 \rightarrow z_1^* + Nw^*,$$

которая сохраняет канонический вид (1) уравнения изучаемой поверхности.

Детальный анализ коэффициентов получаемого при этом нового канонического уравнения, а также изучение связей таких коэффициентов с элементами базисных матриц (3) позволяет получать дополнительные

ограничения на такие элементы. Например, в изучаемых типах $\left(\frac{1}{2}, \varepsilon\right)$

всегда можно считать выполненным только один из двух вариантов:

$$1) t_{27} = \operatorname{Re}(A_{24}) = 0, t_{28} = \operatorname{Im}(A_{24}) = 0 \quad \text{или} \quad 2) t_{27} = 1, t_{28} = 0.$$

Таким образом, общее число неизвестных параметров в обсуждаемой задаче можно снизить до 14. В данной статье будет рассмотрен случай, учитывающий первое ограничение $t_{27} = t_{28} = 0$.

Дальнейший алгоритм изучения нашего случая основан на некоторых особенностях квадратичной системы, связанных с наличием в паре $(\varepsilon_1, \varepsilon_2)$ из (1) значения $1/2$. Именно благодаря этому параметру уравнения

нашей системы имеют относительно простой вид, и в процессе решения системы многие из них разлагаются на линейные множители. Это позволяет, во-первых, не рассматривать весь набор уравнений, а ограничиться исследованием лишь части квадратичной системы. Мы работаем с 16 уравнениями из 25, имеющими наиболее простой вид. Во-вторых, удастся, по сути, автоматизировать процесс решения данной системы. Он основан на использовании символьных вычислений в пакете Maple. Подробно данный алгоритм решения был описан в [7], поэтому здесь приводиться не будет.

3. Система квадратичных уравнений

Искомые неизвестные величины:

$$m_1, m_2, m_3, m_4, t_1, t_3, t_9, t_{17}, t_{18}, t_{21}, t_{22}, t_{23}, t_{25}, t_{31}. \quad (5)$$

Система состоит из 16 квадратичных комплексных уравнений относительно 14 неизвестных вещественных параметров. Например:

$$\begin{aligned} Z_1: & -4t_{22}t_{21}\varepsilon - 2it_{22}^2\varepsilon - it_{21}^2 + 2it_{21}^2\varepsilon - 2t_9t_{21}\varepsilon - 2it_9t_{22}\varepsilon = 0 \\ Z_5: & -t_{21}^2 + 4it_{22}t_{21}\varepsilon - 8it_{22}t_{21}\varepsilon^2 + 4t_{21}^2\varepsilon - 4t_{21}^2\varepsilon^2 + \\ & 2t_9t_{22}\varepsilon + 4t_9t_{22}\varepsilon^2 - 4it_9t_{21}\varepsilon^2 + 2it_9t_{21}\varepsilon + 2t_{22}^2\varepsilon + 4t_{22}^2\varepsilon^2 = 0 \\ Z_{16}: & 4t_{25}t_{21}\varepsilon^2 + 4t_{25}t_{21}\varepsilon + 12t_9t_{17}\varepsilon - 8t_9t_{17}\varepsilon^2 - 4t_{18}t_{21}\varepsilon - \\ & - 12it_{18}t_{22}\varepsilon^2 - 4im_2t_{18}\varepsilon - 8im_2t_{18}\varepsilon^2 - 12it_3t_{21}\varepsilon^2 - \\ & - 6it_{18}t_{22}\varepsilon - 8it_{17}t_{21}\varepsilon - 24im_3t_{21}\varepsilon^2 - 8t_{18}t_{21}\varepsilon^2 - 8m_3t_{22}\varepsilon + \\ & + 16m_3t_{22}\varepsilon^2 - 20t_{17}t_{22}\varepsilon^2 - 2it_{17}t_{21} - 4t_3t_{22}\varepsilon + 8t_3t_{22}\varepsilon^2 - \\ & - 8m_3t_9\varepsilon + 8t_3t_9\varepsilon^2 - 8m_2t_{17}\varepsilon^2 + 6t_{17}t_{22}\varepsilon + 4it_3t_{21}\varepsilon + \\ & + 24t_{17}t_{21}\varepsilon^2 + 8im_3t_{21}\varepsilon + 16m_3t_9\varepsilon^2 + t_{25}t_{21} - 4m_2t_{17}\varepsilon - \\ & - 4t_3t_9\varepsilon + it_3t_{21} + 2im_3t_{21} = 0 \end{aligned}$$

Опишем кратко результаты работы алгоритма.

Всего была получена 1881 алгоритмическая цепочка. Конечным звеном каждой из таких цепочек является либо решение системы, либо набор неразложимых на множители уравнений.

В том числе было получено 878 нулевых (тривиальных) решений и 893 набора (5), совпадающих с другими аналогичными решениями.

Основным результатом является получение 53 различных решений системы.

Отметим еще, что 57 нетривиальных решений из остального множества являются частными случаями основных 53 наборов (5). Полученные решения являются основой для дальнейшего построения

матричных алгебр Ли, их интегрирования и построения списка искомым однородных поверхностей.

Замечание. Далеко не все полученные решения приводят к матричным алгебрам Ли, так как система квадратичных уравнений – это лишь совокупность необходимых (но, вообще говоря, не достаточных) условий для того, чтобы набор изучаемых матриц был алгеброй Ли.

Заключение

Большинство полученных решений привели к семейству матричных алгебр Ли следующего вида:

$$\left(\begin{array}{cccc} A & 0 & 0 & 1 \\ 0 & m & 0 & 0 \\ 4i & 0 & 2m & 0 \\ 0 & 0 & 0 & 0 \end{array} \right), \left(\begin{array}{cccc} 0 & 0 & 0 & i \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{array} \right), \left(\begin{array}{cccc} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 2i(1+2\varepsilon) & 0 & 0 \\ 0 & 0 & 0 & 0 \end{array} \right),$$

$$\left(\begin{array}{ccc} 0 & 0 & 0 \\ 0 & 0 & i \\ 0 & 2-4\varepsilon & 0 \\ 0 & 0 & 0 \end{array} \right), \left(\begin{array}{ccc} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{array} \right), \quad (6)$$

где $A, m \in R, 0 < \varepsilon \neq \frac{1}{2}$. Этому семейству алгебр соответствует семейство аффинно-однородных поверхностей из следующей теоремы.

Теорема 2. Любая аффинно-однородная поверхность M , имеющая алгебру Ли $\mathfrak{g}(M)$ с базисом вида (6), локально аффинно эквивалентна одной из следующих поверхностей ($x_1 = \operatorname{Re}(z_1)$):

$$v = x_1^2 + |z_2|^2 + \varepsilon(z_2^2 + \bar{z}_2^2),$$

$$v = (1+x_1)^\alpha + |z_2|^2 + \varepsilon(z_2^2 + \bar{z}_2^2),$$

$$v = e^{x_1} + |z_2|^2 + \varepsilon(z_2^2 + \bar{z}_2^2),$$

$$v = -\ln(1+x_1) + |z_2|^2 + \varepsilon(z_2^2 + \bar{z}_2^2),$$

$$v = (1+x_1)\ln(1+x_1) + |z_2|^2 + \varepsilon(z_2^2 + \bar{z}_2^2).$$

Получение этих поверхностей можно считать одним из основных теоретических результатов данной статьи, полученных алгоритмическими методами.

Отметим еще, что всего на основе разработанных алгоритмов было получено 8 типов алгебр Ли.

Список литературы

1. Лобода, А. В. Об аффинной однородности поверхностей трубчатого типа в \mathbb{C}^3 / А.В. Лобода, Т. Т. З. Нгуен // Тр. МИАН. – Москва, 2012. – Т. 279. – С. 102-119.
2. Бондаренко, В. С. Описание одного класса 6-мерных матричных алгебр Ли / В. С. Бондаренко, А. П. Бубнов, А. В. Лобода // Сборник студенческих научных работ ФКН ВГУ. – Вып. 9. – Воронеж, 2015. – С. 44-50.
3. Чурсин, А. Г. О матричных элементах одного семейства матричных алгебр Ли / А. Г. Чурсин, Е. А. Павлова, А. В. Лобода // Сборник студенческих научных работ ФКН ВГУ. – Вып. 9 – Воронеж, 2015. – С. 286-291.
4. Нгуен, Т. Т. З. Построение 5-мерных матричных алгебр Ли с помощью пакета MAPLE / Т. Т. З. Нгуен // Вестник ВГУ Сер. Физика. Математика, 2012. – № 1. – С. 162-170.
5. Лобода, А. В. О полном списке аффинно-однородных поверхностей $(\epsilon, 0)$ -типов в пространстве \mathbb{C}^3 / А. В. Лобода, А. В. Шиповская // Известия ВУЗов. Математика. – 2015. – № 6. – С. 75-82.
6. Atanov, A. V. Affine homogeneous strictly pseudoconvex hypersurfaces of the type $(1/2, 0)$ in \mathbb{C}^3 [Электронный ресурс] / A. V. Atanov, A. V. Loboda, A. V. Shipovskaya. – Режим доступа : [ArXiv.org/abs/1401.2252](https://arxiv.org/abs/1401.2252)
7. Павлова, Е. А. Компьютерные алгоритмы решения некоторых систем квадратичных уравнений / Е. А. Павлова, А. В. Лобода // Сборник студенческих научных работ ФКН ВГУ. – Вып. 7. – Воронеж, 2013. – С. 105-113.

ОБФУСКАЦИЯ JAVA БАЙТ-КОДА

Р. И. Петриев

Студент

Д. И. Соломатин

Старший преподаватель

Введение

Одной из частей информационной безопасности является защита программного обеспечения. Одним из методов защиты является *обфускация* или запутывание кода – приведение исходного текста или исполняемого кода программы к виду, сохраняющему ее

функциональность, но затрудняющему анализ, понимание алгоритмов работы и модификацию при декомпиляции. «Запутывание» кода может осуществляться на уровне алгоритма, исходного текста и/или ассемблерного текста. Для создания запутанного ассемблерного текста могут использоваться специализированные компиляторы, использующие неочевидные или недокументированные возможности среды исполнения программы. Существуют также специальные программы, производящие обфускацию, называемые обфускаторами.

В представленной работе рассматривается задача обфускации на уровне байт-кода приложений, разработанных на языке Java. Данная задача актуальна, т. к. скомпилированный стандартным компилятором языка Java код может быть практически полностью восстановлен к исходному виду существующими декомпиляторами.

Среди существующих обфускаторов для языка Java можно выделить следующие: *Stringer*, *Zelix KlassMaster*, *Excelsior JET* и *ProGuard*. Первые три дают неплохой результат, но являются платными. Последний же бесплатен, но является больше оптимизатором, нежели обфускатором. Другие подобные приложения, как правило, содержат в себе ошибки, перестали давно поддерживаться или имеют слишком простой функционал, которого недостаточно.

Реализация обфускатора «с нуля» требует значительных усилий. Поэтому в данной работе использовалась библиотека для анализа и преобразования Java байт-кода – *Soot* [1]. Частью *Soot* является проект *JBCO* (*Java ByteCode Obfuscator*) [2]. В *JBCO* уже есть определенный набор преобразований байт-кода, в дополнении к которым в рамках данной работы было создано ещё два: обфускация синхронизации и шифрование локальных строк.

1. Обфускация синхронизации

Для рассмотрения обфускации синхронизации необходимо понимать, что собой представляют *synchronized*-методы и *synchronized*-блоки. В Java они необходимы для разграничения доступа к ресурсам потоков в многопоточной среде по времени. Их реализация основана на так называемых *мониторах*.

Монитор – это средство обеспечения контроля доступа к ресурсу. У монитора может быть максимум один владелец в каждый текущий момент времени. В Java у каждого экземпляра объекта есть монитор. Используется он следующим образом: любой нестатический *synchronized*-метод при своем вызове прежде всего пытается захватить монитор того объекта, у которого он вызван. Если это удалось – метод исполняется, иначе поток останавливается и ждет, пока монитор будет отпущен другим потоком.

Synchronized-блок описывается следующим образом:

```

Object sync = new Object();
...
synchronized(sync) {
    ...
}

```

В этом случае захватывается монитор у объекта `sync`.

Для объявления `synchronized`-метода достаточно указать для него соответствующий модификатор:

```

public synchronized void someMethod() {
    ...
}

```

В этом случае захватывается монитор у объекта `this`. Для статических методов, у которых нет ссылки на `this`, будет использован объект класса `Class`. Когда виртуальная машина Java (JVM) выполняет `synchronized`-метод (для таких методов в `method_info` проставлен флаг `ACC_SYNCHRONIZED`), она автоматически устанавливает блокировку на соответствующий объект и снимает блокировку после вызова метода или при возникновении исключения.

В отличие от `synchronized`-методов для `synchronized`-блоков запрос и освобождение блокировки, а также сопутствующая обработка возможных исключений, описываются явно в байт-коде:

```

0: aload_0          // загружаем this в стек
1: dup              // дублируем ссылку
2: astore_1         // сохраняем дубликат
3: monitorenter    // захватываем this
4: aload_0
5: getfield         // загружаем поле у this
6: nop
7: iconst_m1        // загружаем константу
8: aload_1          // загружаем this для освобождения
9: monitorexit     // освобождаем this
10: ireturn          // возвращаем значение
11: astore_2         // здесь начинается блок catch и
                    // сохраняется ссылка на объект исключения
12: aload_1          // загружаем this для освобождения
13: monitorexit     // освобождаем this
14: aload_2         // загружаем ссылку на исключение
15: athrow          // выбрасываем перехваченное исключение

```

Для реализации обфускации `synchronized`-метода используется преобразование метода к блоку. Сначала необходимо снять флаг `ACC_SYNCHRONIZED` с метода, убрав его из `method_info`. Затем необходимо создать `synchronized`-блок, который включает в себя весь метод. При преобразовании `synchronized`-метода в блок надо учитывать, что в методе может быть несколько точек возврата и везде необходимо предусмотреть освобождение объекта. В библиотеке Soot есть отличное решение этой проблемы – `ExceptionalUnitGraph`. Он позволяет найти все точки возврата [3, 4].

При обфускации важно искать «золотую середину» между сложностью защиты и производительностью.

Для достижения лучшей степени защиты необходимо добавлять элементы, которые создают неоднозначность при выполнении метода. Например, добавить условий (оператор `if` или `switch`). Условие можно сделать нетривиальным, например, с использованием случайных чисел, но при этом условие будет выполняться всегда однозначно. В случае применения оператора `switch` самым простым решением является передача ему случайного числа, которое будет из такого диапазона, что всегда попадет в ветку `default` или набор других. Для большего эффекта, лучше добавить элементы случайности в порядок и содержимое обфусцирующего кода, чтобы не было однозначного пути восстановления.

Для обфускации `synchronized`-блоков необходимо работать с `monitorenter` и `monitorexit`. Преобразование находит эти инструкции и начинает добавлять вокруг них обфусцирующий код, о котором говорилось выше, и перемещать их в нестандартные места. При этом главное, чтобы приложение после этого работало корректно.

Рассмотрим простейший пример. Java код:

```
public void setSyncVal(int val) {
    synchronized(this) {
        syncVal = val;
    }
}
```

Этот код просто устанавливает значение переменной в блоке синхронизации. Байт-код после обфускации (приводится только для `monitorenter`) выглядит следующим образом (отступами выделен добавленный код):

```
0: aload_0
1: astore_2
2: sipush      500
5:  istore_3    // лок. переменная для условий и switch
6:  sipush      500
9:  istore     4 // лок. переменная для условий и switch
11: iload_3
12: lookupswitch { // добавляем switch для усложнения
                    // вычисления порядка исполнения
                    54: 75
                    default: 69
                }
32: iload_3
33: iload      4
35: iadd
36: iload      4
38: iadd
39: istore_3
40: aload_0
41: iload_1
```

```

42: putfield   #9          // field syncVal:I
45: aload_2
46: monitorexit
47: goto      55
50: astore_1
51: aload_2
52: monitorexit
53: aload_1
54: athrow
55: return
56: aload_0
57: monitorenter          // перенесен в конец метода после return
58:   iload_3
59:   iload      4
61:   if_icmpge  32       // дополнительное условие
64:   iload_3
65:   iload      4
67:   iadd
68:   istore_3
69:   iload_3
70:   iload      4
72:   if_icmpge  56       // дополнительное условие
75:   goto      11

```

В Soot процесс обфускации происходит так: после построения структуры программы внутри Soot, наступает очередь анализа/преобразования кода. В данном случае будет преобразование на уровне метода, поэтому класс наследуется от `BodyTransformer`. Soot будет по очереди применять к текущему методу различные преобразования, вызывая метод `internalTransform` с соответствующими параметрами [5].

```

Iterator stmtIt = units.snapshotIterator();
while (stmtIt.hasNext()) {
    Stmt stmt = (Stmt) stmtIt.next();
    if (stmt instanceof EnterMonitorStmt) {
        obfuscateStmt(body, units, stmt);
        monitorsTransformed++;
    }
    if (weight > 6) { // параметр дает возможность изменять
                     // "силу" обфускации
        if (stmt instanceof ExitMonitorStmt) {
            obfuscateStmt(body, units, stmt);
            monitorsTransformed++;
        }
    }
}
}

```

Код ищет инструкции `monitorenter` и `monitorexit` для выполнения преобразований. Этот код относится к Jimple-представлению байт-кода (стандартное полное представление). Переменная `units` типа `PatchingChain<Unit>` является коллекцией инструкций метода. Соответственно `Unit` – это базовый класс инструкции, а `Stmt` – выражения.

В байт-коде рассмотренного выше примера видно, что `monitorenter` находится после `return`. Для перемещения вниз инструкции просто удаляем её из тела метода и добавляем в конец, где `stmt` и есть `monitorenter`. Для байт-кода это нормальная ситуация несмотря на то, что непосредственно в языке Java добавление инструкций после `return` недопустимо. Важно вставить в конце переход обратно в то место, где изначально находилась инструкция. Иначе на этапе верификации класса возникнет ошибка и приложение перестанет работать.

Также стоит учитывать, что можно применить и другие преобразования. Например, `Packing Local Variables into Bitfields` в данном случае очень сильно осложнит анализ даже человеком, не говоря уже об автоматизации процесса.

В Soot также есть специальный класс `LookupSwitchStmt` для `switch` (вообще существует две инструкции для этого оператора: `lookupswitch` и `tableswitch`). В нашем случае для создания необходимо определить:

- локальную переменную со значением;
- набор возможных значений;
- набор соответствующих значениям инструкций, к которым будет осуществлен переход для выполнения блока операций для этого значения;
- инструкцию для значения по умолчанию (в Java-коде секции `default` может и не быть, но на самом деле компилятор сгенерирует его в любом случае).

Далее рассматривается код после `return`.

Для «прыжка» от одной инструкции к любой другой в методе используется `goto` [6]. Но в нашем случае делать переход из одного места в другое с его помощью нельзя. Проблема в том, что некоторые умные декомпиляторы умеют восстанавливать исходный порядок инструкций в байт-коде, убирая лишние `goto` и перемещая блоки в исходные места.

Как можно заметить, оператор `switch` в байт-коде отличается сильно от его представления в языке высокого уровня. По сути `switch` это тот же `goto`, но с несколькими возможными вариантами целей для «прыжков». То же самое и с оператором `if`.

2. Шифрование локальных строк

В `.class`-файлах строки хранятся в открытом виде, что может весьма помочь реверс-инжинирингу. Также в некоторых приложениях могут быть запросы к базам данных, что дает информацию о структуре базы данных, или же текст запросов к серверу.

Чтобы этот метод работал, необходимо заменить все значения строк на зашифрованные, а затем вставить дешифрующий алгоритм в месте использования значения.

Алгоритм должен соответствовать определенному минимальному набору требований:

- должен быть почти уникальным для всех строк;
- должен использовать несколько операций с разными ключами;
- не должен слишком снижать производительность;
- может быть скрыт в подпроцедуре или перемещен в другое место.

Под «почти уникальным» подразумевается, что теоретически в очень большом приложении есть вероятность появления очень схожих алгоритмов. Уникальность зависит от:

- количества операций в методе;
- количества различных видов операций;
- количества ключей.

Если эти требования учтены, то схожих методов будет очень мало или их не будет вообще. Это необходимо для того, чтобы значительно усложнить анализ кода и сделать крайне затруднительным создание алгоритма для автоматического дешифрования.

Также алгоритмы могут использовать разное кол-во разных операций в разной последовательности. В процессе реализации потребуется несколько случайных чисел, чтобы задать реализацию алгоритма для текущей строки. Дешифрование не должно слишком сильно повлиять на производительность, иначе приложение может стать слишком медленным. Здесь возможна оптимизация – найти циклы и вынести дешифрование за их пределы. Для того, чтобы ещё больше усложнить восстановление кода, алгоритм можно перенести в другое место, можно расщепить на части и распределить их по методу.

Ниже описаны действия для выражений вызова метода, которые были реализованы в рамках данной работы:

- получение выражений вызова;
- поиск строковых констант в аргументах;
- создание переменной с зашифрованным значением и замена константы;
- добавление перед вызовом алгоритма дешифрования.

В случае с другими типами инструкций, например, присваивания, работа идет также с константными значениями. Соответственно, при шифровании необходимо запоминать порядок блоков, чтобы при дешифровании поставить их в обратном порядке.

Заключение

На данный момент описанные преобразования байт-кода полностью реализованы. Тестирование возможности восстановления кода после их

применения проводилось следующими декомпиляторами: Fernflower, Java Decompiler, ProGuard Deobfuscator и JD-GUI. Ни один из них не смог полноценно восстановить исходный код.

Список литературы

1. Soot : A framework for analyzing and transforming Java and Android Applications [Электронный ресурс]. – Режим доступа : <http://sable.github.io/soot>
2. Collberg, C. A taxonomy of obfuscating transformations / C. Collberg, C. Thomborson, D. C. Low. – Auckland, New Zealand : University of Auckland, 1997. – 36 p.
3. Collberg, C. Manufacturing cheap, resilient and stealthy opaque constructs / C. Collberg, C. Thomborson, D. C. Low. – Auckland, New Zealand : University of Auckland, 1998. – 13 p.
4. JBCO: the Java ByteCode Obfuscator [Электронный ресурс]. – Режим доступа : <http://www.sable.mcgill.ca/JBCO>
5. Einarsson, Arni. A Survivor's Guide to Java Program Analysis with Soot / Arni Einarsson, Janus Dam Nielsen. – version 1.1. – Aarhus, Denmark : BRICS, Department of Computer Science University of Aarhus, 2008. – 47 p.
6. Cappaert, Jan. Code Obfuscation Techniques for Software Protection / Jan Cappaert. – Leuven : Katholieke Universiteit Leuven, 2012. – 132 p.

ОЦЕНОЧНАЯ ХАРАКТЕРИСТИКА СЕТЕВОГО ТРАФИКА ЯЧЕЙСТОЙ СЕТИ В СИСТЕМЕ ОРNET

А. А. Пешков
Студент
Ю. Б. Нечаев
Профессор

Введение

В настоящее время сети передачи данных продолжают активно развиваться, в том числе все большее распространение получают сети с ячеистой топологией. Технология Wireless Mash или ячеистые сети расширяет функциональность беспроводного доступа в Интернет и позволяет реализовывать точки доступа с охватом и порогом снижения пропускной способности на порядок более высоким, чем у привычных хот-спотов. Благодаря возможности обеспечения защищенного беспроводного

покрытия как внутри помещений, так и на улицах, в городской местности или в крупных населенных пунктах и районах, Wireless Mesh может быть использована для быстрого развертывания, в частности, сети связи для целей внутренней безопасности или в случаях чрезвычайных ситуаций в городе.

Особенностями Wireless Mesh является:

- создание зон сплошного информационного покрытия большой площади;
- масштабируемость сети (увеличение площади зоны покрытия и плотности информационного обеспечения) в режиме самоорганизации;
- использование беспроводных транспортных каналов для связи точек доступа в режиме "каждый с каждым";
- устойчивость к потере отдельных элементов.

В силу этих особенностей к протоколам маршрутизации в ячеистых сетях предъявляются следующие требования:

1. протоколы должны быть распределенными;
2. обеспечивать надежную доставку пакетов в условиях постоянно изменяющейся топологии сети;
3. обеспечивать малое время построения маршрута в условиях постоянно изменяющейся топологии сети;
4. обладать механизмами оперативного обнаружения разрыва маршрута и его восстановления;
5. не допускать образования петель в маршрутах;
6. рассылать при функционировании как можно меньший объем служебной информации;
7. обеспечивать высокую производительность сети при различных ее размерах [1].

В качестве среды моделирования я буду применять систему Ornet Modeler. Данная система предлагает пользователям графическую среду для создания, выполнения и анализа событийного моделирования сетей связи. Это удобное программное обеспечение может быть использовано для большого ряда задач, например, типичные – создание и проверка протокола связи, анализ взаимодействий протокола, оптимизация и планирование сети. Также возможно осуществить с помощью пакета проверку правильности аналитических моделей и описание протоколов [2].

Моделирование сети

Рассмотрим созданную беспроводную ячеистую сеть (рис 1). Она состоит из 20 мобильных узлов и располагается на площади 500x625 м.

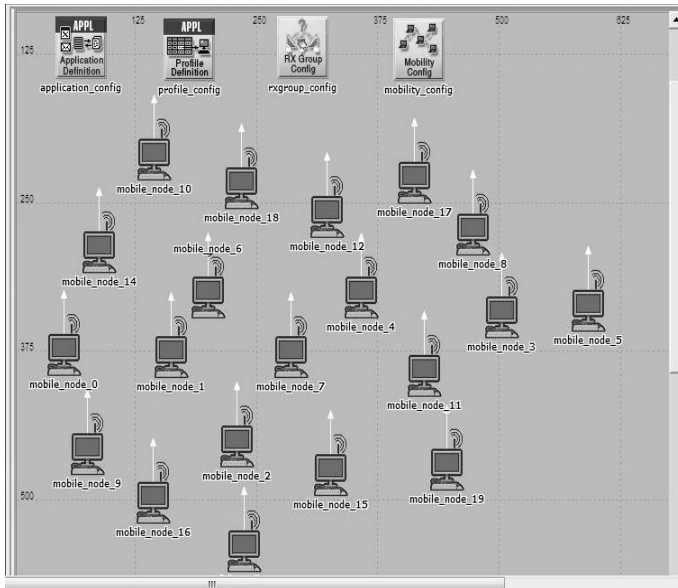


Рис. 1. Схема смоделированной сети

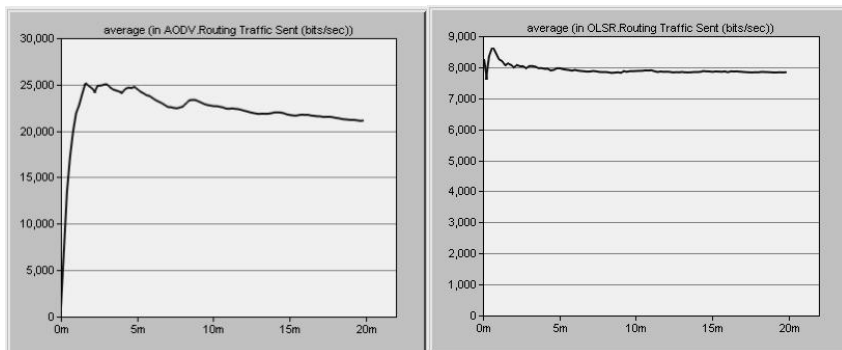
Элементы сети настроены на работу по протоколам маршрутизации AODV (Ad hoc On-Demand Distance Vector) и OLSR (Optimized Link-State Routing). Эти два протокола используются для передачи данных в мобильных сетях с самоорганизующейся топологией.

Различия этих протоколов состоят в том, что AODV – динамический протокол, разработанный для использования в сетях различного размера. Представляет собой комбинацию двух протоколов: DSR и DSDV, от которого он наследует концепцию hello-сообщений. Это необходимо для поддержания узлом актуального списка своих соседей, что позволяет ускорить процесс рассылки запросов на построение маршрута.

OLSR – статический протокол, который является попыткой адаптации классического LSR протокола к использованию в условиях беспроводных сетей. Основным нововведением является концепция многоточечных ретрансляторов, которая оптимизирует процесс широковещательной рассылки, значительно сокращая объемы рассылаемой информации [3].

Работа данных протоколов тестируется в построенной сети и на основе полученных данных, сделаем вывод о влиянии выбора протокола на сетевой трафик.

По результатам проведенных симуляций работы сети получаем графики параметров сети для протокола AODV и протокола OLSR (рис. 2).



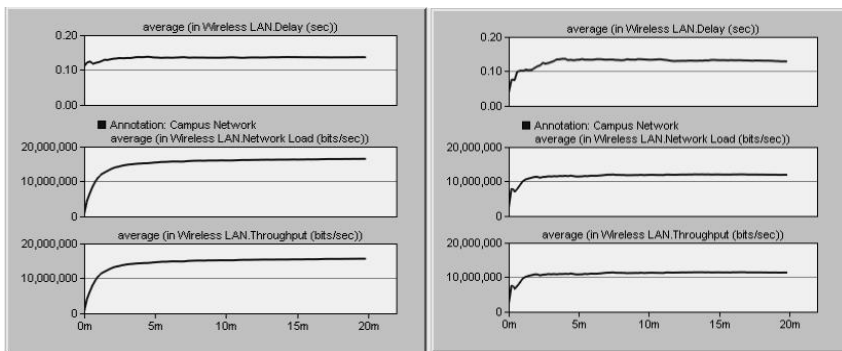
а б

а – протокол AODV, б – протокол OLSR

Рис. 2. Отправленный трафик

Сравнив графики отправленного трафика (рис. 2) можно сделать вывод об объеме переданных данных. Из графиков видно, что объем переданной информации, при одинаковых параметрах сети, по протоколу AODV в три раза больше.

Далее проведем сравнение параметров временной задержки, сетевой нагрузки и пропускной способности (рис 3).



а б

а – протокол AODV, б – протокол OLSR

Рис. 3. Параметры работы сети

Полученные результаты исследования для удобства занесем в таблицу.

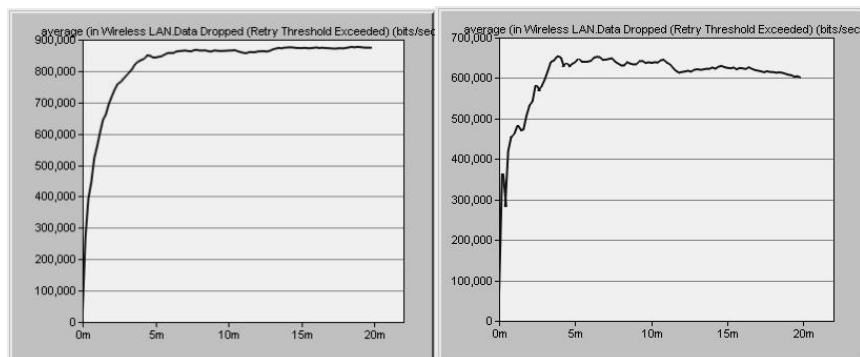
Таблица

Параметры работы сети

| Параметр | AODV | OLSR |
|------------------------------|-------------|-------------|
| Временная задержка, с | 0.14 | 0.13 |
| Сетевая нагрузка, Мб/с | 17 | 12 |
| Пропускная способность, Мб/с | 15 | 11 |

Из таблицы видно, что разница во временной задержке не столь велика, а значения сетевой нагрузки и пропускной способности существенно различаются.

Следующими исследованными параметрами являются параметры потери информации (рис. 4).



а

б

а – протокол AODV, б – протокол OLSR

Рис. 4. Потеря данных

Исходя из соотношений переданной и потерянной информации можно сделать вывод о потерях: для протокола AODV потери составили 3%, а для протокола OLSR – 8%.

Заключение

Данная статья посвящена ячеистым сетям передачи информации и влиянию протоколов маршрутизации на сетевой трафик в ячеистых сетях. Результаты моделирования свидетельствуют о том, что протокол

маршрутизации оказывает существенное влияние на основные сетевые параметры, поэтому одной из основных задач на этапе моделирования сети должен быть выбор протокола построения маршрутов, основанный на требованиях к сети. При высокой нагрузке целесообразно использовать протокол AODV, который, несмотря на большие временные задержки, чем у OLSR, позволяет минимизировать потери информации и не создает высокой нагрузки на сеть за счёт собственной служебной информации.

Список литературы

1. Управление, вычислительная техника и информатика / В. М. Винокуров, [и др.] // Доклады ТУСУРа. Часть 1. – 2010. – № 2 (22). – С. 288-294.
2. Тарасов, В. Н. Анализ и оптимизация локальных сетей и сетей связи с помощью программной системы OPNET MODELER / В. Н. Тарасов, А.Л. Коннов, Ю.А. Ушаков // Вестник ОГУ. – 2006. – №6. – С. 197-204.
3. Блэк, Ю. Сети ЭВМ : протоколы стандарты, интерфейсы / Ю. Блэк. – Москва : Мир, 1990. – 385 с.

КОМПЬЮТЕРНАЯ ОЦЕНКА БИОЛОГИЧЕСКИХ РИТМОВ И АДАПТАЦИОННЫХ ВОЗМОЖНОСТЕЙ СТУДЕНТОВ ВОРОНЕЖСКОГО МЕДИЦИНСКОГО УНИВЕРСИТЕТА

А. Б. Пилюгин

Студент

А. Ю. Лукаш

Студент

Н. С. Троицкая

Студент

Ю. А. Пахомова

Студент

О. Н. Попова

Студент

© Пилюгин А. Б. (ВГУ), Лукаш А. Ю., Троицкая Н. С., Пахомова Ю. А., Попова О. Н. (ВГМУ им Н.Н. Бурденко), Борзунов С. В., Кургалин С. Д. (ВГУ), Друганова Л. П., Фертикова Т. Е. (ВГМУ им Н.Н. Бурденко), 2015

С. В. Борзунов
Доцент
С. Д. Кургалин
Профессор
Л. П. Друганова
Доцент
Т. Е. Фертикова
Доцент

Введение

Неразумное отношение человека к своему здоровью, к окружающей природе часто является следствием незнания биологических законов и адаптивных возможностей. Чтобы сохранить здоровье и работоспособность, интеллектуальный, эмоциональный и физический потенциал, необходима большая просветительская работа, в результате которой человек сможет грамотно и рационально планировать свою рабочую деятельность. Знание хронобиотипа может помочь в составлении режима рабочего дня и определении временных пиков активности для конкретного студента [1].

1. Цель и задачи исследования

Цель исследования состояла в определении хронобиологического типа при помощи компьютерной программы методом тестирования, а также в выявлении и сопоставлении показателей индивидуальной минуты (ИМ) со стандартными величинами.

Задачи исследования — разработать компьютерную программу, позволяющую:

1. определить хронобиологический тип человека;
2. определить биологический возраст человека
3. изучить воздействие биологических ритмов на физическую работоспособность;
4. рассмотреть факторы, способствующие нарушению биологических ритмов;
5. рассчитать ИМ и определить адаптивные возможности организма.

2. Материал и методы исследования

В ходе исследования были использованы анкетирование, статистическая и графическая обработка, расчет ИМ при помощи секундомера, и разработана компьютерная программа для анализа полученных данных.

Для обработки результатов анкетирования был разработан программный комплекс, который определяет:

- принадлежность к одному из трех основных хронобиотипов: утренний или «Жаворонки», вечерний или «Совы» и индифферентный или «Голуби»;
- Биологический возраст;
- Индивидуальную минуту.

Рассмотрим подробнее особенности программного комплекса. Приложение реализовано на языке С#. Используется база данных под СУБД SQLite ver3. Блок-схемы различных модулей представлены на рис.1-2. Некоторые из диаграмм сущностей базы данных изображены на рис. 3-4.

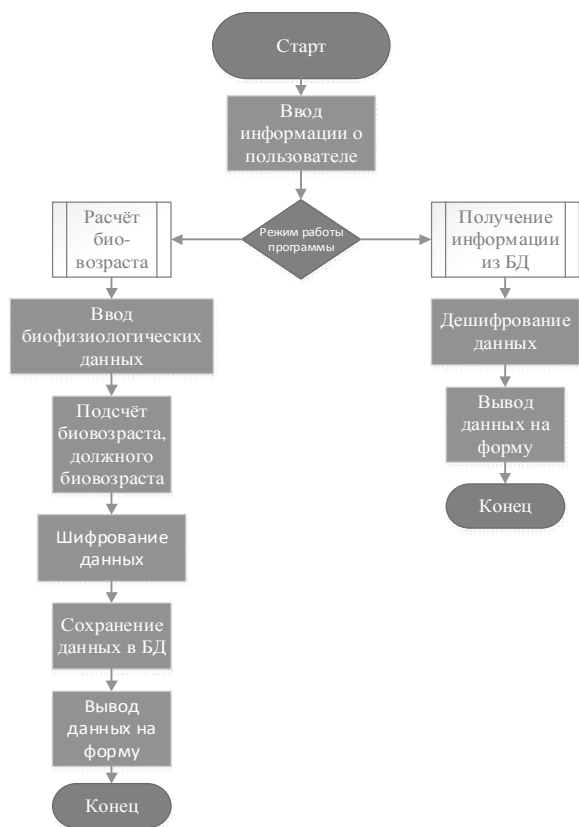


Рис. 1. Блок-схема модуля "Биовозраст"

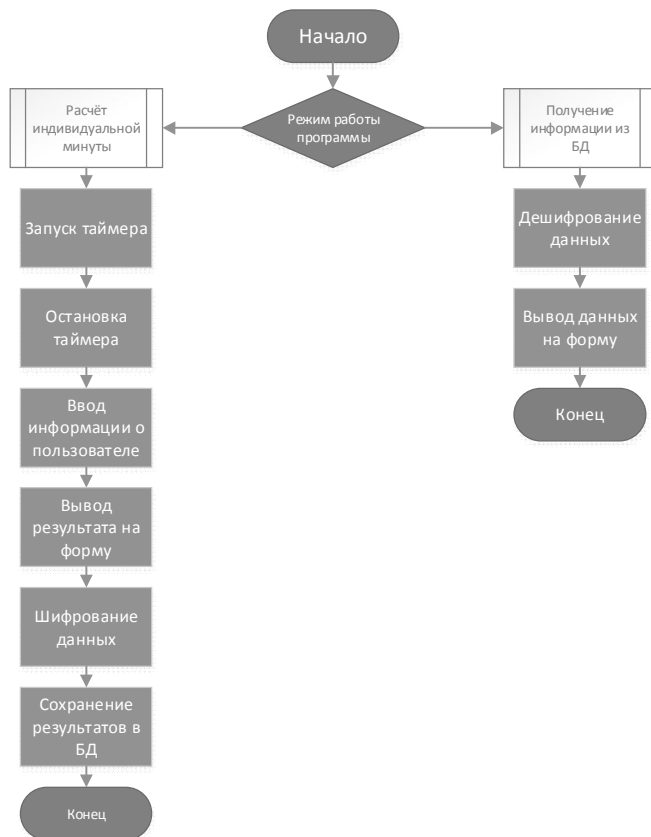


Рис. 2. Блок-схема модуля "Индивидуальная минута"

Для безопасного хранения данных используется шифрование методом Плейфера [2]. Генерируется матрица 6×6 , состоящая из букв русского алфавита и специальных символов и случайно перемешивается. Чтобы зашифровать сообщение, необходимо разбить его на пары символов (биграммы), например «привет» становится «пр ив ет», и отыскать эти биграммы в таблице. Два символа биграммы соответствуют углам прямоугольника в ключевой матрице. Определяем положения углов этого прямоугольника относительно друг друга. Затем, руководствуясь следующими 4 правилами, зашифровываем пары символов исходного текста:

– Если два символа биграммы совпадают (или если остался один символ), добавляем после первого символа «\$», зашифровываем новую пару символов и продолжаем.

– Если символы биграммы исходного текста встречаются в одной строке, то эти символы замещаются на символы, расположенные в ближайших столбцах справа от соответствующих символов. Если символ является последним в строке, то он заменяется первым символом этой же строки.

– Если символы биграммы исходного текста встречаются в одном столбце, то они преобразуются в символы того же столбца, находящиеся непосредственно под ними. Если символ является нижним в столбце, то он заменяется первым символом этого же столбца.

– Если символы биграммы исходного текста находятся в разных столбцах и разных строках, то они заменяются символами, находящимися в тех же строках, но соответствующие другим углам прямоугольника.

Для расшифровки необходимо использовать инверсию этих четырёх правил, откидывая символы «\$», если они не несут смысла в исходном сообщении.



Рис. 3. Диаграмма сущностей БД для биовозраста

| биотип | |
|-----------------|--|
| id | |
| пол | |
| возраст | |
| хронобиотип | |
| время адаптации | |
| описание | |

Рис. 4. Диаграмма сущностей БД для биотипа и индивидуальной минуты

3. Результаты

Разработанный программный комплекс применяется на кафедре физики, математики и медицинской информатики для вычисления таких показателей, как:

- биологический возраст;
- индивидуальная минута;
- хронобиотип.

Результаты анкетирования показали, что испытуемые оценивают себя следующим образом:

- четко утренний тип – 4 человека (8%);
- скорее, утренний тип, чем вечерний – 10 человек (20%);
- индифферентный тип – «Голубь» – 12 человек (24%);
- скорее, вечерний тип, чем утренний – 10 человек (20%);
- четко вечерний тип – «Сова» – 14 человек (28%).

Данные, полученные в ходе компьютерного тестирования на предмет хронобиологической принадлежности, показали более точные результаты, чем подсчеты, проведенные вручную, а также выявили несоответствия в самоопределении испытуемых по биотипу. По результатам компьютерной программы было выявлено:

- «Жаворонок» – 2 человека (4%);
- «Голубь» – 33 человека (66%);
- «Сова» – 15 человек (30%).

У здоровых людей величина ИМ является относительно стойким показателем, характеризующим эндогенную организацию времени и адаптационные способности организма.

У лиц с высокими способностями к адаптации ИМ равна 62,90-69,71 с, а у людей низкими – в среднем 47,0-46,2 с.

По результатам ИМ были выявлены следующие показатели среди студентов от 18 до 22 лет:

- ИМ превышает 1 минуту физического времени – 68%;
- ИМ меньше 1 минуты физического времени – 32%.

4. Заключение

В соответствии с поставленной целью был создан программный комплекс, который определяет принадлежность к одному из трех основных хронобиотипов, биологический возраст и индивидуальную минуту испытуемых. Разработанный программный комплекс применяется на кафедре физики, математики и медицинской информатики ВГМУ им Н.Н. Бурденко для определения адаптивных возможностей организма человека.

Список литературы

1. Доскин, В. А. Биоритмы, или Как стать здоровым / В. А. Доскин. – Москва : Эксмо, 2014. – 240 с.
2. Бабаш, А. В. История криптографии / А. В. Бабаш, Г. П. Шанкин. – Москва : Гелиос АРВ, 2002. – 240 с.

СХЕМА РАБОТЫ ИНТЕРПРЕТАТОРА ЯЗЫКА ПРОГРАММИРОВАНИЯ С ДИНАМИЧЕСКИ РАСШИРЯЕМЫМ СИНТАКСИСОМ

М. В. Полев

Студент

Д. И. Соломатин

Старший преподаватель

Введение

В процессе разработки программного обеспечения программисты ограничены логикой и возможностями языка программирования, который они используют. Случается так, что инструментария используемого языка программирования недостаточно для решения поставленной задачи.

Попытка решения данной проблемы привела к созданию языково-ориентированной парадигмы программирования. Она заключается в решении конкретных специфических задач специально написанными для них предметно-ориентированными языками (Domain Specific Language).

Каждый DSL направлен на решение одной специфической задачи. При этом расширение языка, на котором ведется разработка является более удобным, чем создание нового языка. Под расширением подразумеваем добавление в язык конструкций, позволяющих лаконично описать и

решить поставленную задачу. Для удобства разработки расширения должны носить динамический характер, то есть производиться в процессе выполнения программного кода [1].

Возможность расширения языка программирования должна быть предусмотрена исполнителем (в частном случае интерпретатором) этого языка. Соответственно, исполнитель должен иметь специфичную структуру, позволяющую расширить язык еще во время разбора его исходного кода. В данной статье обзревается упрощенная схема интерпретатора динамически расширяемого языка программирования.

1. Схема работы интерпретатора статического языка

Рассмотрим упрощенно процесс работы интерпретатора языка программирования. Интерпретатор, как правило, состоит из 2-х основных частей: синтаксического анализатора и исполнителя (в частных случаях исполнитель можно рассматривать как виртуальную машину). Входная программа на языке, понимаемым интерпретатором (язык L) разбирается синтаксическим анализатором и преобразуется во внутреннее представление, удобное для дальнейшего исполнения, в простейшем варианте – AST-дерево (Abstract Syntax Tree). Выполнение программы осуществляется исполнителем уже по внутреннему представлению входной программы. Суть процесса отражена на рис. 1.

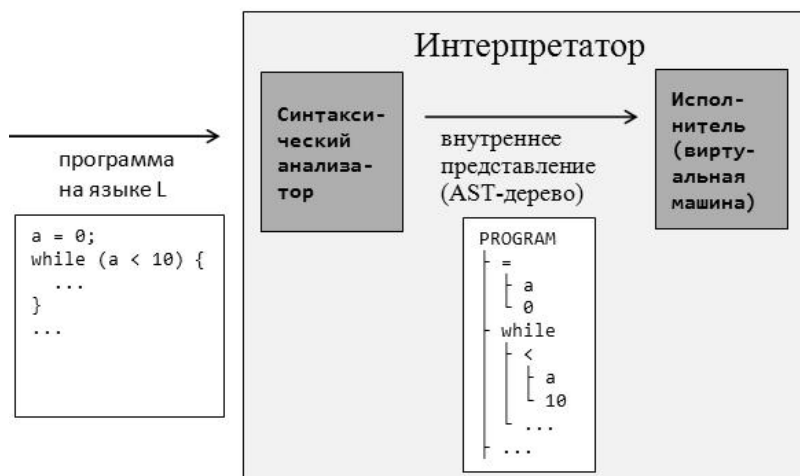


Рис. 1. Упрощенная схема работы интерпретатора

Стоит отметить, что исполнитель также работает с языком L в том смысле, что может выполнять только конструкции языка L в представлении, очищенном от конкретного синтаксиса языка L.

Синтаксический анализатор интерпретатора строится, как правило, с помощью специальной утилиты для построения синтаксических анализаторов, которая на вход получает грамматику языка L с вставками кода, отвечающими за построение AST-дерева, а на выходе генерирует код синтаксического анализатора. Для утилиты построения синтаксических анализаторов также можно говорить о входном языке – языке грамматик. Обозначим этот язык как язык G. Схема работы утилиты для построения синтаксических анализаторов показана на рис. 2.

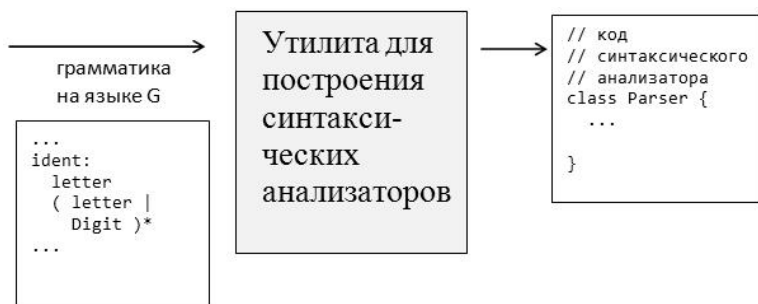


Рис. 2. Схема работы утилиты построения синтаксических анализаторов

2. Выбор типа грамматик

Обязательным условием для возможности расширения языка L является построение синтаксического анализатора этого языка на основе грамматики, допускающей расширение. Необходимо использовать класс порождающих строго однозначный разбор грамматик, замкнутый относительно операций добавления новых правил в грамматику и изменения существующих [2].

Широко используемые для построения синтаксических анализаторов классы грамматик, такие как LL(k) и LR(k) не удовлетворяют этому требованию, так как в результате расширения синтаксиса языка программирования может получиться цепочка лексем языка, при анализе которой невозможно будет сделать однозначный выбор правила грамматики, которое стоит применить к этой цепочке.

Для реализации поставленной в работе задачи был выбран тип PEG-грамматик, предложенный в 2002 г. Брайаном Фордом в работе [3].

2. Схема работы интерпретатора расширяемого языка

Теперь рассмотрим, какова должна быть общая схема работы интерпретатора языка программирования с расширяемым синтаксисом. Так как синтаксис языка задается с помощью грамматик, то, следовательно, интерпретатор должен понимать два языка – исходный

язык L и язык грамматик G. При этом язык G является статическим, в то время как язык L может меняться (дополняться) во время работы интерпретатора. Отсюда следует требование, что, как синтаксический анализатор, так и исполнитель языка L должны быть представлены некой динамической структурой, допускающей изменения в процессе работы интерпретатора. Общая структура и схема работы такого интерпретатора показана на рис. 3.

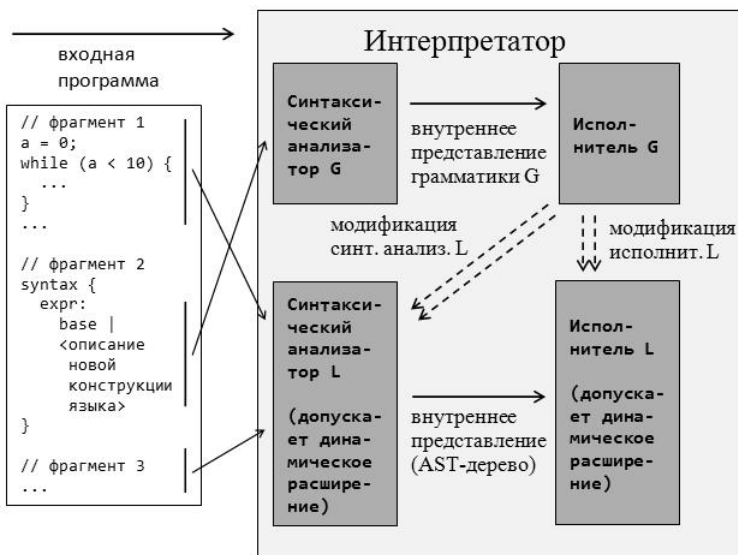


Рис. 3. Схема работы интерпретатора синтаксически расширяемого языка

Входная программа для рассматриваемого интерпретатора кроме непосредственно кода, может содержать фрагменты, предназначенные для расширения языка L (на рис. 3 это фрагмент 2). Как только синтаксический анализатор языка L наталкивается на конструкцию, отвечающую за расширения языка, он передает управление синтаксическому анализатору языка G, который в свою очередь, вызывает исполнитель языка G, модифицирующий синтаксический анализатор L и, возможно, исполнитель L. То есть после разбора фрагмента 2 у нас входной язык уже не L, а его расширение, которое можно обозначить как L'. Таким образом, фрагмент 3 может быть уже написан на языке L', т. е. с использованием расширения, описанного во фрагменте 2.

Стоит сказать, что в данной схеме не указано, каким именно образом осуществляется расширение исполнителя L. Здесь возможны различные варианты, например, описание правил выполнения расширения на самом

языке L или подключение к исполнителю стороннего модуля, отвечающего за выполнение расширения.

3. Реализация

В рамках данной работы автором был разработан прототип языка программирования с возможностью динамического расширения. Прототип включает в себя интерпретатор, который идеологически и функционально соответствует схеме интерпретатора расширяемого языка, описанной выше, и исходный код языка, удовлетворяющий грамматикам, описанным в интерпретаторе.

Язык G представлен как язык грамматик, удовлетворяющий PEG грамматике. Является статическим и встроен в интерпретатор языка L. Фрагмент языка G представлен на рис.4.

```
NUMBER:
  a = '-'? b = ([ '0'-'9' ])+
  -> (a^ b "number")
;
STRING:
  b = '"' a = (IDENTIFIER / CHAR) c = '"'
  -> (b^ "string" a, c)
;
CHAR:
  a = ('(' / ')' / ';' / '{' / '}')
  -> (a^ "char")
;
IDENTIFIER:
  a = ( ([ 'A'-'Z' ] / '_' / [ 'a'-'z' ] ) / [ '0'-'9' ] )+
  -> (a^ "ident")
;
```

Рис. 4. Фрагмент языка грамматик

В результате синтаксического анализа языка G интерпретатором будет получено внутреннее представление языка G в виде абстрактного синтаксического дерева, фрагмент которого представлен на рис.5.

В дальнейшем это дерево будет использовано в синтаксическом анализаторе исходного текста языка, фрагмент которого изображен на рис. 6. На фрагменте можно видеть конструкцию расширения языка L, описанную на самом языке L, которая добавляет в язык оператор «fog» используя уже имеющуюся возможность языка L понимать оператор «while», а так же использование этого оператора непосредственно в программе.

анализа абстрактное синтаксическое дерево будет использоваться непосредственно исполнителем языка L для выполнения исходного текста.

Заключение

В статье были рассмотрен упрощенный принцип построения интерпретатора расширяемого языка программирования. На основе проведенного анализа и полученной схемы был реализован прототип динамически расширяемого языка программирования. Идеи, представленные в работе, были подтверждены на практике при проверке возможности расширить грамматику созданного прототипа языка.

Методы, используемые в работе, не являются единственными возможными решениями поставленной задачи. Принцип, представленный в данной статье, является базовым для динамического расширения языков программирования, но реализация может различаться в зависимости от предпочтений конкретного разработчика.

Список литературы

1. Ахо, А. Теория синтаксического анализа, перевода и компиляции. Том 1 / А. Ахо, Дж. Ульман. – Москва : Мир, 1978. – 612 с.
2. Соломатин, Д. И. Объектно-ориентированный подход к описанию и расширению синтаксиса языков программирования / Д. И. Соломатин // Вестник Воронеж. гос. ун-та. Сер. системный анализ. – 2008. – № 2. – С. 51-55.
3. Ford, B. Packrat Parsing: a Practical Linear-Time Algorithm with Backtracking / B. Ford; Dept. of Electrical Engineering and Computer Science. – Cambridge : Massachusetts Institute of Technology, 2002. – 112 p.

АЛГОРИТМ ВЫДЕЛЕНИЯ ГРАНИЦ РЕЧЕВЫХ ФРАГМЕНТОВ В АУДИОСИГНАЛАХ

А. Г. Полтаева

Студент

П. С. Лысачёв

Старший преподаватель

Введение

Задача выделения границ речевых фрагментов в аудиосигналах при наличии шума является одной из наиболее важных задач обработки речи. Её решение позволяет существенно сократить объём обрабатываемой

информации, а следовательно и объём ресурсов, затрачиваемых на дальнейшую обработку.

Проблема выделения речевых фрагментов сложна – за исключением случаев большого отношения сигнал/шум, когда даже энергия наиболее слабых звуков речи превышает энергию шума. Но такие условия, как правило, не встречаются в реальных ситуациях. В реальности неречевые фрагменты могут включать в себя тишину, шум, другие акустические сигналы; они могут быть как стационарными, так и нестационарными. Это значительно усложняет задачу выделения границ речи.

1. Характеристики аудиосигнала

Для решения задачи выделения границ речи в аудиосигнале нужно определить, какие характеристики следует использовать. Как волна, звук характеризуется амплитудой и спектром частот. Спектр есть совокупность значений частот составляющих синусоидальных звуковых волн. Амплитуда определяет силу звука, его энергию, частота – высоту звука. На основе амплитуды и спектра частот можно получить различные параметры сигнала, позволяющие делать выводы относительно характера сигнала.

Различные параметры характеризуют сигнал с разных сторон, поэтому логично использовать набор параметров для классификации фрагментов сигнала. В описываемом алгоритме используются такие характеристики сигнала, как энергия Гиджера [7, 8], среднее число пересечений нуля [2], мера спектральной плоскостности [4], мера изменчивости сигнала [6], усреднённая спектральная дельта-функция автокорреляции [7] и доминирующая частота [3].

Энергия Гиджера рассчитывается по формуле:

$$TE_n = \left[\sum_{k=0}^{K/2} (k\Delta f)^2 |X_n^k|^2 \right]^{0.5}, \quad (1)$$

где n – номер кадра, k – индекс частоты, $k\Delta f$ – частота, X_n^k – амплитудный спектр текущего кадра. При этом для решения задачи выделения речевых фрагментов частоты $k\Delta f$ лучше брать не от 0 до $K\Delta f/2$, а от 300 Гц до 4000 Гц – поскольку именно в этом диапазоне может находиться речь; таким образом, шум, находящийся вне этого диапазона, не повлияет на расчёт данной характеристики.

Однако энергия Гиджера не позволяет обнаружить глухие звуки, поскольку их энергия значительно меньше энергии вокализованных сегментов речи. Для этого вместе с данной характеристикой можно использовать функцию среднего числа пересечений нуля.

Функция среднего числа переходов через нуль предоставляет информацию относительно количества изменений знака сигнала. Чем больше раз сигнал переходит через нуль, тем быстрее изменяется сигнал, а значит, сигнал содержит высокочастотную информацию.

Данная характеристика определяется следующим образом:

$$Z_n = \frac{1}{2N} \sum_{m=n-N+1}^n |\text{sign}[s(m)] - \text{sign}[s(m-1)]|, \quad (2)$$

где $\text{sign}[s(m)]$ – функция, равная 1, если аргумент больше или равен нулю, и -1 – в противном случае.

При низком уровне шума этих двух характеристик может быть достаточно для выделения границ речи, однако в реальных условиях процент ошибок при обнаружении речи возрастает. Это определяет необходимость использования дополнительных характеристик.

Доминирующая частота определяется формулой:

$$F_n = \arg \max(X_n^k), \quad (3)$$

где X_n^k – амплитудный спектр кадра. Ширина полосы частот, занимаемой речевым сигналом, называется частотным диапазоном речи. Эта полоса заключается в диапазоне частот от 100 Гц до 8000 Гц, но при обработке аудиосигнала ограничиваются интервалом частот от 300 Гц до 3400 Гц.

Мера изменчивости сигнала описывает степень нестационарности сигнала. Данная характеристика рассчитывается через предыдущие R кадров исследуемого сигнала s_n относительно кадра n :

$$L_n = \frac{1}{K} \sum_{k=1}^K \left[\xi_n^k - \bar{\xi}_n \right]^2, \quad (4)$$

$$\xi_n^k = \sum_{m=n-R+1}^n \frac{S_m^k}{\sum_{l=n-R+1}^n S_l^k} \log \frac{S_m^k}{\sum_{l=n-R+1}^n S_l^k}, \quad (5)$$

$$\bar{\xi}_n = \frac{1}{K} \sum_{k=1}^K \xi_n^k, \quad (6)$$

где S_m^k – краткосрочная оценка спектра, рассчитывается с использованием метода Бартлетта-Уэлча [9], который усредняет оценки спектра для M последовательных кадров. Смысл характеристики в том, что мера энтропии ξ_n^k сильнее изменяется в зависимости от k , если кадр содержит речь, чем когда кадр содержит только шум.

Мера спектральной плоскостности вычисляется как отношение среднего геометрического $Gm_n(k)$ и среднего арифметического $Am_n(k)$ спектра для кадра n . Эта величина измеряется по логарифмической шкале и принимает значения в диапазоне $(-\infty, 0]$:

$$LSFM_n = \sum_{k=0}^K \log_{10} \frac{Gm_n(k)}{Am_n(k)}. \quad (7)$$

Спектр здесь также оценивается с использованием метода Бартлетта-Уэлча. Смысл характеристики в том, что спектр сигнала более организован в течение речевых фрагментов сигнала, чем во время шума. Высокое значение спектральной плоскостности означает, что спектр обладает примерно одинаковой мощностью на всех частотах, что близко к белому шуму. Низкое значение спектральной плоскостности означает, что мощность спектра сосредоточена в небольшом количестве частот.

Формула для расчёта усреднённой спектральной дельта-функции автокорреляции:

$$md_n = \left[\sum_{l=0}^L |\Delta R_n^P(l)| \right]^{1/2}, \quad (8)$$

где $l=0..L$, $L=K/2$, L – число периодов, по которым рассчитывается коэффициент автокорреляции (число лагов автокорреляции). Спектральная дельта-функция автокорреляции рассчитывается по формуле:

$$\Delta R_n^P(l) = \frac{\sum_{q=-Q}^Q q R_n^P(l+q)}{\sum_{q=-Q}^Q q^2}, \quad (9)$$

где $Q = 3$, которое может принимать значения от 2 до 5. Значение спектральной функции автокорреляции для кадра n вычисляется по формуле:

$$R_n^P(l) = \sum_{k=0}^{K/2-1-l} |X_n^k|^2 |X_n^{k+l}|^2, \quad (10)$$

где $|X_n^k|$ – амплитудный спектр сигнала, K – количество значений спектра.

Для речевых кадров спектральная функция автокорреляции содержит больше пиков с большими амплитудами, чем для неречевых кадров. Усреднённая спектральная дельта-функция автокорреляции как параметр,

значение которого зависит от числа пиков и их амплитуд, может использоваться для обнаружения речи.

2. Алгоритм выделения границ речевых фрагментов

Идея алгоритма заключается в том, что для каждого кадра сигнала рассчитываются все необходимые характеристики и сравниваются с пороговыми значениями, после чего кадр считается речевым в том случае, если более двух характеристик удовлетворяют пороговым условиям.

Энергия Тиджера рассматривается вместе с функцией среднего числа пересечений нуля: проверяются значения Z_n вокруг выделенных с помощью энергии Тиджера границ речи; в том случае, если, по крайней мере, для 3 кадров из ближайших 15 Z_n превышает пороговое значение, все 15 кадров помечаются, как речь. Это позволяет обнаружить глухие звуки на границах фраз.

Пороговые значения рассчитываются исходя из предположения, что первые несколько кадров аудиосигнала не содержат речь. Далее эти значения постоянно обновляются в зависимости от сигнала. Формулы для расчёта пороговых значений энергии Тиджера и усреднённой спектральной дельта-функции автокорреляции представлены в [7], среднего числа пересечений нуля – в [2], долгосрочной меры спектральной плоскостности – в [4], долгосрочной меры изменчивости сигнала – в [6]. Доминирующая частота для речевого кадра должна принимать значения от 300 Гц до 3400 Гц, её пороговые значения не пересчитываются.

3. Результаты эксперимента

Алгоритм, описанный в данной статье, был реализован на языке программирования C#. В качестве исходных данных были взяты файлы формата WAVE. Сигналы разбивались на кадры длиной 20 мс с перекрытием 10 мс. Результаты работы алгоритма для аудиосигнала, содержащего фразу «Раз два три», представлены на рис. 1-рис. 2. Рисунки демонстрируют устойчивость алгоритма к шумам даже достаточно большой мощности.

Тестирование алгоритма было проведено на различных данных: не содержащих шума, содержащих различные виды стандартных шумов с разным соотношением сигнал/шум, а также на реальных аудиосигналах. В конечном итоге были сопоставлены результаты алгоритмов, описанных соответственно в [2], [3], [6], [4], [7], с результатами данного алгоритма.

В ходе тестирования работы алгоритмов были получены значения таких характеристик точности, как ошибки первого рода, ошибки второго рода, ошибки в начале речи, ошибки в конце речи, абсолютная точность (процент всех верно классифицированных кадров), точность определения речи, точность определения шума.

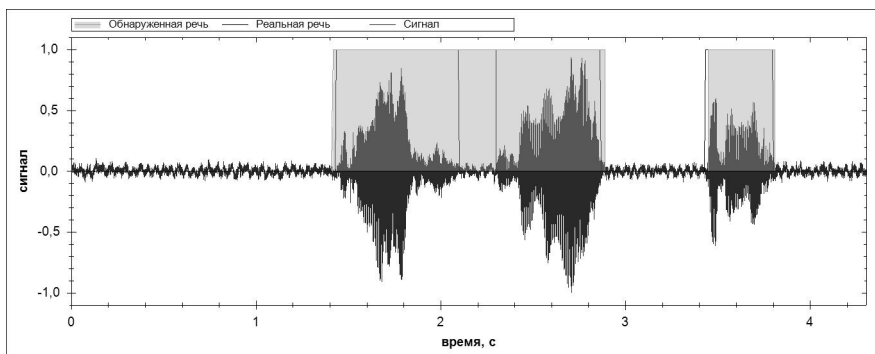


Рис. 1. Результат работы алгоритма

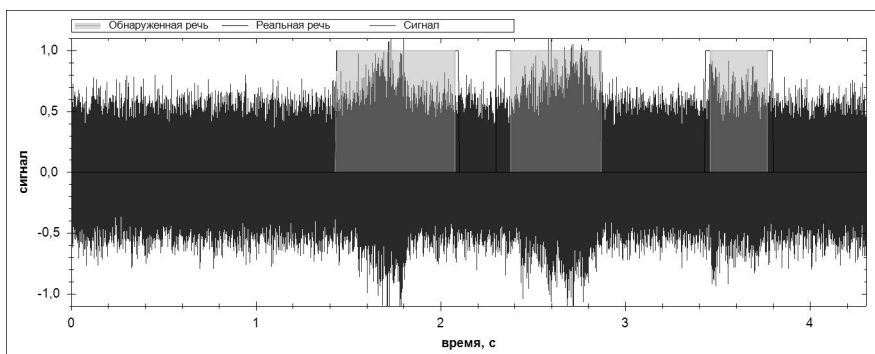


Рис. 2. Результат работы алгоритма на зашумлённых данных (гауссовский белый шум)

Лучшим считается алгоритм, обеспечивающий большие значения абсолютной точности и точности речи одновременно: отсутствие речевых потерь важно, однако большое количество лишних кадров, помеченных речевыми, может лишить смысла процесс выделения границ речи, предназначенный для сокращения обрабатываемой информации. Большая часть ошибок должна быть сосредоточена у границ речи, поскольку тогда сохраняются шансы восстановить граничные слова.

Результаты тестирования алгоритмов на незашумлённых исходных данных представлены в таблице 1. Из них следует, что, хотя первый и пятый алгоритмы дают преимущество в точности определения речи, они больше других теряют при определении отсутствия речи. Таким образом, описанный алгоритм (№6) в наибольшей степени удовлетворяет

поставленным условиям, сочетая высокие уровни определения речи и пауз, а также преимущественное сосредоточение ошибок на границах речевых фрагментов.

Таблица 1

Тестирование алгоритмов на незашумлённых данных

| № | Ошибки I рода, % | Ошибки II рода, % | Ошибки в начале, % | Ошибки в конце, % | Точность, % | Точность речи, % | Точность пауз, % |
|---|------------------|-------------------|--------------------|-------------------|-------------|------------------|------------------|
| 1 | 26,75 | 0,48 | 21,32 | 42,32 | 88,63 | 99,52 | 73,25 |
| 2 | 11,61 | 4,83 | 14,93 | 52,31 | 91,62 | 95,18 | 88,39 |
| 3 | 19,58 | 7,77 | 22,77 | 39,41 | 86,16 | 92,23 | 80,42 |
| 4 | 9,53 | 34,44 | 14,12 | 20,84 | 73,37 | 65,56 | 90,47 |
| 5 | 27,16 | 0,79 | 28,14 | 28,23 | 87,18 | 99,21 | 72,84 |
| 6 | 10,99 | 3,90 | 26,43 | 42,28 | 92,47 | 96,10 | 89,00 |

Результаты тестирования алгоритмов на реальных данных представлены в таблице 2. Наивысшую точность обеспечивает именно последний алгоритм, сочетающий различные характеристики, в то время как показатели первого и пятого резко ухудшаются.

Таблица 2

Тестирование алгоритмов на реальных данных

| № | Ошибки первого рода, % | Ошибки второго рода, % | Ошибки в начале речи, % | Ошибки в конце речи, % | Точность, % | Точность речи, % | Точность пауз, % |
|---|------------------------|------------------------|-------------------------|------------------------|-------------|------------------|------------------|
| 1 | 20,20 | 23,65 | 15,24 | 15,17 | 76,34 | 76,35 | 79,80 |
| 2 | 20,03 | 38,34 | 10,98 | 16,18 | 69,69 | 61,66 | 79,97 |
| 3 | 26,94 | 25,38 | 12,59 | 20,53 | 72,57 | 74,62 | 73,06 |
| 4 | 20,94 | 23,47 | 14,03 | 20,33 | 76,16 | 76,53 | 79,06 |
| 5 | 56,78 | 3,51 | 11,23 | 8,607 | 68,29 | 96,49 | 43,22 |
| 6 | 30,63 | 12,42 | 15,52 | 18,77 | 76,52 | 87,58 | 69,37 |

Заключение

Данная статья посвящена проблеме выделения фрагментов речи в аудиосигнале. Описан алгоритм выделения фрагментов речи на основе дополняющих друг друга характеристик аудиосигнала. В ходе эксперимента было установлено, что алгоритм устойчив к разным видам и уровням шумов и обладает преимуществом над другими алгоритмами в

плане результата, но время его выполнения также превышает время работы других алгоритмов. Однако в том случае, если подобные задержки не принципиальны для конечной системы, данный алгоритм позволяет решать проблему выделения речи с достаточно высокой точностью.

Список литературы

1. Рабинер, Л. Р. Цифровая обработка речевых сигналов / Л. Р. Рабинер, Р. В. Шафер. – Москва : Радио и связь, 1981. – 496 с.
2. Rabiner, L. R. An Algorithm for Determining the Endpoints of Isolated Utterances / L. R. Rabiner, M. R. Sambur // Bell System Technical Journal. – 1975. – Vol. 54. – Issue 2. – P. 297-315.
3. Moattar, M. H. A simple but efficient real-time voice activity detection algorithm / M. H. Moattar, M. M. Homayounpour // 17th European Signal Processing Conference, Glasgow, Scotland, August 24-28, 2009. – EUSIPCO. EURASIP, 2009. – P. 2549-2553.
4. Ma, Y. Efficient voice activity detection algorithm using long-term spectral flatness measure [Электронный ресурс] : электрон. журн. / Y. Ma, A. Nishihara // EURASIP Journal on Audio, Speech, and Music Processing. – Режим доступа: <http://asmp.eurasipjournals.com/content/2013/1/21>
5. VAD techniques for real-time speech transmission on the Internet / A. Sangwan [и др.] // 5th IEEE International Conference on High-Speed Networks and Multimedia Communications, Jeju Island, Korea, July 3-5, 2002. – IEEE, 2002. – P. 46-50.
6. Ghosh, P. K. Robust voice activity detection using long-term signal variability / P. K. Ghosh, A. Tsiartas, S. Narayanan // Journal IEEE Transactions on Audio, Speech, and Language Processing. – 2011. – Vol. 19. – Issue 3. C. 600-613.
7. Ouzounov, A. A. Telephone Speech Endpoint Detection Using Mean-Delta Feature / A. A. Ouzounov // Cybernetics and Information Technologies. – 2014. – Vol. 14. – No 2. – C. 127-139.
8. Ouzounov, A. A. Robust Feature for Speech Detection / A.A. Ouzounov // Cybernetics and Information Technologies. – 2004. – Vol. 4. – No 2. – P. 3-14.
9. Manolakis, D.G. Statistical and Adaptive Signal Processing: Spectral Estimation, Signal Modeling, Adaptive Filtering and Array Processing / D. G. Manolakis, V. K. Ingle, S. M. Kogon. – New York : Artech House, 2005. – 796 p.

МУЛЬТИПЛАТФОРМЕННОЕ НЕПРЕРЫВНОЕ ТЕСТИРОВАНИЕ ПРОГРАММНЫХ ПРОДУКТОВ ФИРМЫ UCS

Р. И. Полянских

Студент

А. В. Распопов

Доцент

Введение

Тестирование системы — процесс исследования программного обеспечения (ПО) с целью получения информации о качестве продукта.

Вопросы качества ПО крайне актуальны. Программы, содержащие ошибки, могут быть причиной утечки информации, финансовых потерь, простоя. Неотъемлемой частью процесса обеспечения качества является тестирование на предмет обнаружения ошибок.

По существующим оценкам [1], затраты на тестирование программ составляют до 50% всего бюджета программного продукта. Таким образом, тестирование — очень трудоемкий и дорогостоящий процесс. Стоит отметить, что с расширением функциональности ПО расходы на тестирование нелинейно возрастают. Поэтому, задача автоматизации процессов тестирования, а так же снижения затрат на их поддержку не теряет своей актуальности.

Существует множество готовых решений в области автоматизированного тестирования, однако в силу особенности специфической графики тестируемого ПО продуктов фирмы UCS и отсутствие поддержки других платформ, они не удовлетворяют требованиям, предъявляемым к качеству тестирования. Кроссплатформенных решений для программных продуктов относительно немного, и лишь малая их часть находится в открытом доступе. Поэтому было решено разработать решение для непрерывного мультиплатформенного автоматизированного тестирования.

1. Виды тестирования, непрерывная интеграция

Главный критерий принятия решения об автоматизации — возможность сэкономить время и затраты на тестирование.

Автоматизированное тестирование ПО — это процесс верификации программного обеспечения, при котором основные функции и шаги теста, такие как запуск, инициализация, выполнение, анализ и выдача результата,

выполняются автоматически при помощи инструментов для автоматизированного тестирования [2].

Основные преимущества такого подхода:

- Повторяемость
- Быстрота выполнения
- Выполнение без вмешательства
- Меньшие затраты на поддержку
- Отчеты

Непрерывная интеграция (Continuous Integration – CI) – это процесс, состоящий из непрерывного компилирования, тестирования, анализа и развертывания исходного кода [3]. Данная методика предполагает использование системы управления версиями (как например CVS, Subversion, git) для хранения исходных кодов и всего остального, что необходимо для сборки и тестирования проекта. В системах с непрерывной интеграцией это означает запуск новой сборки всякий раз, когда изменяется текст программы в репозитории системы управления версиями исходного кода. Это позволяет минимизировать вмешательство тестировщика в процесс тестирования. Одной из таких систем непрерывной интеграции, используемой для тестирования продуктов UCS является система Jenkins..

2. Оптимизация затрат внедрения автоматизированной системы тестирования

Ручное тестирование. Совокупность затрат на тестирование это оценка расходов на использование ручного труда тестировщика на конкретной итерации проекта. Сюда может также входить количество конфигураций отдельных тестов, к примеру: один тест на 2 платформы.

Совокупность затрат на ручное тестирование можно выразить следующим образом:

$$Em = m * Nc * tc * f \cos t(t) * n , \quad (1)$$

где Em – совокупность затрат на ручное тестирование, m – количество тестов, выполняемых вручную, Nc – количество конфигураций, tc – время выполнения тестов вручную, $f \cos t$ – стоимость работы тестировщика на данном этапе проекта, n – количество итераций.

Автоматизированное тестирование. Совокупность затрат будет состоять из начальных расходов на приобретение оборудования, сюда же входит стоимость специальных средств на автоматизацию (ПО, лицензий); а так же из работы автоматизатора в единицу времени. Время работы автоматизатора можно представить как сумму времени на разработку и поддержку всех автотестов, а также непосредственно времени

тестирования. Исходя из этого, затраты на внедрение и поддержку автоматизированного тестирования (Ea) можно выразить, как:

$$Ea = C + N * (ta + te + tm) * fc(t), \quad (2)$$

где C – фиксированные затраты на автоматизацию (стоимость лицензий, стоимость оборудования); Nc – количество конфигураций тестирования; ta – время на разработку всех автотестов; te – время выполнения автотестов; tm – время на поддержку тестов, fc – стоимость работы автоматизатора в единицу времени.

Используя данные багтрекинговой системы (Redmine) был произведен расчет и построен график оценки ресурсных затрат на тестирование программного продукта (рис. 1).



Рис. 1. Оценка ресурсных затрат на тестирование

Из рис. 1 видно, что переход от ручного тестирования к автоматизированному несомненно выгоден. Однако вначале требуются достаточно серьезные усилия на разработку ядра тестовой системы, после этого разработка новых тестов проста. Точка пересечения N показывает, на каком этапе разработки эффективен переход от ручного к автоматизированному тестированию. Так как с ростом количества конфигураций увеличиваются и расходы, то ставится задача мультиплатформенного непрерывного тестирования.

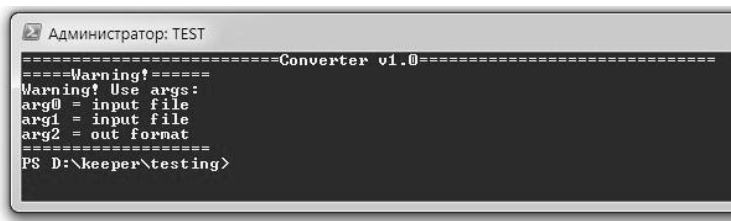
3. Разработка решения для мультиплатформенного тестирования

Для мультиплатформенного решения был разработан скрипт-конвертор, позволяющий вести разработку на одном языке сценариев (например, PowerShell под Windows) и тестировании ПО на различных операционных системах.

Основной функцией конвертора является поиск и замена регулярных выражений, встречающихся в сценарии.

```
.\converter.ps1 in.ps1 test1.sh linux
```

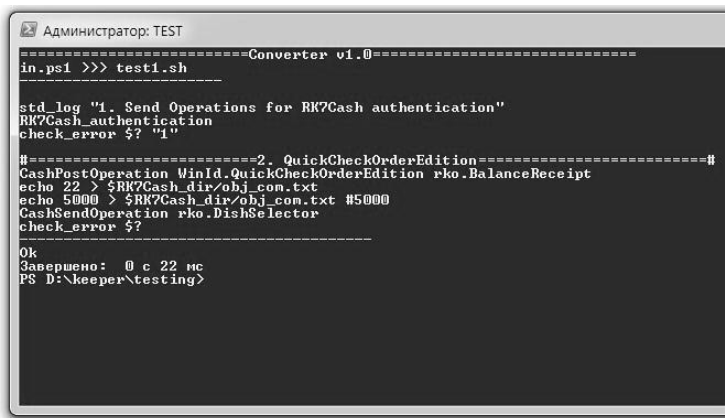
Здесь фигурируют три переменных, они же и являются параметрами запуска конвертора. Допустимы следующие параметры запуска: [исходный скрипт] [выходной скрипт] [win/linux] – ОС для конвертации. В случае отсутствия одного или нескольких параметров выдается предупреждающее сообщение пользователю и работа конвертора прекращается с пояснением ошибки (рис. 2):



```
Администратор: TEST
=====Converter v1.0=====
-----Warning!-----
Warning! Use args:
arg0 = input file
arg1 = input file
arg2 = out format
=====
PS D:\keeper\testing>
```

Рис. 2. Запуск конвертора без параметров

Пример успешной работы скрипта представлен на рис. 3.



```
Администратор: TEST
=====Converter v1.0=====
in.ps1 >>> test1.sh
-----
std_log "1. Send Operations for RK7Cash authentication"
RK7Cash_authentication
check_error $? "1"
#-----2. QuickCheckOrderEdition-----#
CashPostOperation WinId.QuickCheckOrderEdition rko.BalanceReceipt
echo 22 > $RK7Cash_dir/obj_com.txt
echo 5000 > $RK7Cash_dir/obj_com.txt #5000
CashSendOperation rko.DishSelector
check_error $?
-----
Ok
Завершено: 0 с 22 мс
PS D:\keeper\testing>
```

Рис. 3. Успешное преобразование скрипта конвертором

Данный конвертор применим для тестовых сценариев, написанных для тестирования различных продуктов фирмы UCS (R-Keeper, Premiera, Kitchen Display System). Таким образом, разработка тестовых сценариев сведена к разработке их всего лишь под одну ОС. При этом возможна и обратная конвертация скрипта под другую операционную систему.

4. Оценка качества и степень внедрения решения

В тестировании участвовали 3 программных продукта фирмы UCS (R-Keeper, Premiera, Kitchen Display System). В ходе исследования было выяснено, что внедрение решения мультиплатформенного тестирования с использованием скрипта-конвертора существенно сокращает затраты на разработку и поддержку новых тестовых сценариев (рис. 4).

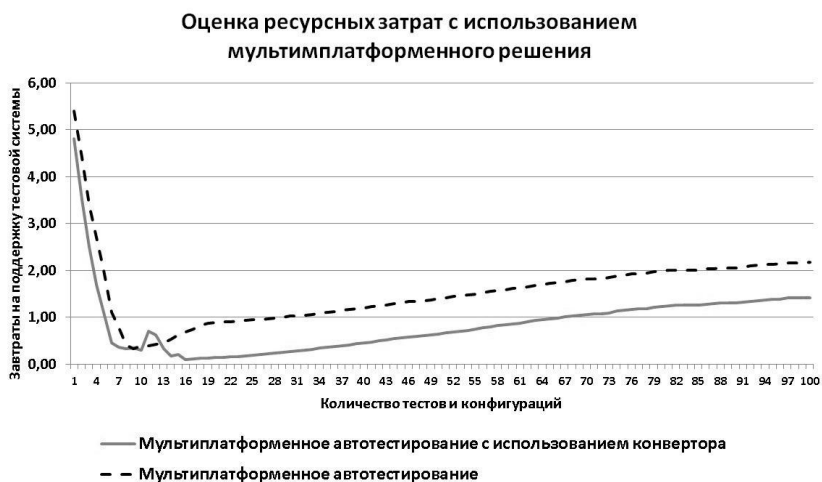


Рис. 4. Оценка ресурсных затрат с использованием конвертора

Здесь можно увидеть небольшой подъем графика для решения с использованием конвертора, обусловленный разработкой самого скрипта-конвертора. Однако после его внедрения необходимость разработки тестовых сценариев на 2 разных языках под две различные ОС отпала. Теперь достаточно вести разработку всего лишь на одном языке, затем выполнить конвертацию и получить аналогичный скрипт под другую ОС.

Заключение

В ходе исследования эффективности внедрения автоматизированной тестовой системы можно сделать выводы о том, что автоматизация тестирования – неотъемлемая часть любой разработки, которая позволяет экономить не только время и затраты на тестирование, но и повысить

качество выпускаемого продукта путем тестирования его на различных платформах, благодаря мультиплатформенному решению автоматизированной системы тестирования.

Система автотестирования находится на сервере тестирования Jenkins и обеспечивает тестирование продуктов в режиме реального времени. Следующим этапом планируются различные улучшения, основываясь на отзывах пользователей и требованиях разработчиков.

Список литературы

1. Орлов, С. А. Технологии разработки программного обеспечения : учебник для вузов / С. А. Орлов, Б. Я. Цилькер. – 4-е изд. – Санкт-Петербург : Питер, 2012. – 608 с.
2. Сеницын, С. В. Верификация программного обеспечения / С. В. Сеницын, Н. Ю. Налютин. – Москва : БИНОМ, 2008. – 368 с.
3. Дюваль, Поль М. Непрерывная интеграция: улучшение качества программного обеспечения и снижение риска / Поль М. Дюваль, Стивен М. Матиас III., Эндру Гловер. – Москва : Вильямс, 2008. – 240 с.

ПРОГНОЗИРОВАНИЕ ФИНАНСОВЫХ РЫНКОВ С ПОМОЩЬЮ МОДЕЛИ ВЕГЕ-ИЗИНГА

В. А. Потапова

Студент

М. Е. Семенов

Профессор

Введение

Современный финансовый рынок характеризуется значительной сложностью протекающих на нем процессов. С одной стороны финансовый рынок достаточно хаотичен, поскольку его эволюция определяется волей большого количества людей, а с другой в нем действуют устойчивые механизмы, определяемые коллективным поведением участников.

В этой связи построение формальных моделей, позволяющих лучше понять структуру и поведение рынка, как единого целого, так и его составляющих, долгое время привлекали и продолжают привлекать внимание практиков и исследователей.

В действительности же работа на финансовом рынке сводится к покупке и продаже валюты и получению выгоды. Правило "покупай

дешевле и продавай дороже" является, пожалуй, точным описанием сути работы на валютном рынке.

Чтобы приобретенная валюта приносила доходы, нужно уметь правильно управлять ею. В современном мире существует множество экономико-математических моделей, посредством которых решаются те или иные задачи во всех сферах деятельности человека, в частности на финансовом рынке. Важным моментом является прогнозирование последующих событий. В настоящее время становятся особенно актуальными работы, позволяющие хотя бы в минимальной степени смоделировать и объяснить законы финансового рынка. Эти работы важны и для инвесторов, интересующихся возможностью прогнозирования поведения цен финансовых активов, и для регулирующих органов, которых интересует возможность влияния на рынок так, чтобы он наилучшим образом соответствовал целям развития экономики.

1. Вычисление показателя Херста и R/S-анализ

Данная характеристика широко применима к анализу временных рядов в экономике и на рынках капитала, так как исключительно устойчива и позволяет отличать случайные ряды с независимыми приращениями от неслучайных рядов, ее предложил английский ученый Гарольд Херст [1]. С помощью алгоритма R/S-анализа можно выявить следующие свойства временного ряда: значение показателя Херста H и соответствующий ему «цвет шума»; оценка меры устойчивости временного ряда; наличие долговременной памяти и оценка её глубины; наличие циклов.

Приведем алгоритм вычисления постоянной Херста [2].

Пусть имеется ряд наблюдений $E = \{e_i\} \ i=1,2,\dots,N$ где N – общее число наблюдений.

Выберем фиксированный временной промежуток $T = n$ и организуем цикл по $k=1$ до $N-n$. При $k=1$ выбираем первые n наблюдений e_1, e_2, \dots, e_n и вычисляем среднюю M_{1n} . Временной ряд накопленных отклонений определяется как, где $m < n$:

$$X_{1m} = \sum_{i=1}^m (e_i - M_{1n}). \quad (1)$$

Далее, при $k=2$ выбираем серию наблюдений e_2, e_3, \dots, e_{n+1} . Как и выше, вычисляем среднее M_{2n} и X_{2m} :

$$X_{2m} = \sum_{i=2}^{m+1} (e_i - M_{2n}). \quad (2)$$

Повторяя процесс вычислений для всего цикла по $k=1$ до $N-n$, получим $N-n$ значений кумулятивных отклонений $X_{km} \ (k=1, \dots, N-n)$.

Следующим шагом вычисляем размах последовательности ряда накопленных отклонений $\{X_{km}\}$ как:

$$R_n = \max(X_{km}) - \min(X_{km}). \quad (3)$$

Для возможности сравнения различных типов рядов Херст предложил разделить этот размах на стандартное отклонение исходных наблюдений.

Пусть S – выборочное стандартное отклонение ряда $\{e_i\}$, $i=1..n$, вычисляемое по следующей формуле:

$$S_n = \sqrt{1/n} \sum_{i=1}^n (e_i - M_{i,n}). \quad (4)$$

Тогда нормализуя размах R_n путем деления на S , получаем величину R_n/S – безразмерную вариацию ряда, которая будет расти с увеличением временного промежутка $T = n$.

Херст ввел следующее соотношение:

$$R/S = (an)^H, \quad (5)$$

где a – некоторая константа; n – число наблюдений; H – показатель Херста.

Прологарифмировав последнее равенство, получим:

$$\ln(R/S) = H \ln n + \ln a. \quad (6)$$

Откладывая по оси x $\ln n$, а по оси y – $\ln(R/S)$ получаем точки, по которым, используя метод наименьших квадратов, строим линейную регрессию. Наклон полученной линии регрессии и даст значение постоянной Херста.

Применительно к финансовым данным можно использовать следующую трактовку: показатель Херста H измеряет влияние информации на временной ряд данных. Значение $H = 0.5$ («белый шум») подразумевает случайное блуждание, что является подтверждением гипотезы эффективного рынка. В этом случае события некоррелированы, все новости уже впитаны и обесценены рынком. В противоположность этому при $H > 0.5$ («черный шум») события сегодня будут иметь значение завтра, то есть полученная информация продолжает учитываться рынком некоторое время спустя. Это не просто автокорреляция, когда влияние информации быстро падает (кратковременная память), а это долговременная память. Она обуславливает информационное влияние в течение больших периодов времени и характеризуется длиной цикла. Если же $H < 0.5$ («розовый шум»), то мы имеем дело с антиперсистентным рядом. Такой ряд волатилен, то есть более изменчив, чем ряд случайный. Он состоит из частых реверсов «спад-подъем».

2. Гипотеза когерентного рынка

Модель была разработана Тонисом Веге в 1990 году [3]. Первоначально модель Веге-Изинга была призвана для описания ферромагнитных материалов во внешнем магнитном поле, в нашей модели она соответствует однородным группам трейдеров.

Сначала мы строим гистограмму из котировок какого-либо индекса финансового рынка. Затем, определяем функцию распределения рыночных доходностей:

$$f_{st}(q) = \frac{c}{Q(q)} \exp \left[2 \int_{-1/2}^q \frac{K(q)}{Q(q)} dq \right], \quad (7)$$

где q – переменная, описывающая преобладающие тенденции на рынке (позитивные или негативные), $q \in [-1/2, 1/2]$, c – это нормирующая константа, $K(q)$ - коэффициент дрейфа, которая задается следующей формулой:

$$K(q) = \sinh(k \cdot q + h) - 2 \cdot q \cdot \cosh(k \cdot q + h), \quad (8)$$

$Q(q)$ - коэффициент диффузии:

$$Q(q) = \frac{1}{n} \cdot [\cosh(k \cdot q + h) - 2 \cdot q \cdot \sinh(k \cdot q + h)]. \quad (9)$$

Функция (7) меняет свою форму при изменении ее управляющих параметров k , h и N . Значение параметра показателя толпы k можно получить с помощью показателя Херста, алгоритм вычисления которого описан выше [1]. Расчет степеней свободы N и фундаментального смещения h производится путем подгонки уравнения (1).

От значений управляющих параметров зависит, в каком рыночном состоянии будет находиться рынок:

1. Эффективный рынок ($k \approx 1.8$, $h \approx 0$, $N = 186$);
2. Переходные состояния рынка ($k \approx 2.0$, $h \approx 0$, $N = 186$);
3. Хаотический рынок ($k \approx 2.2$, $h \approx 0$, $N = 186$);
4. Когерентные рынки:
 - а) когерентный медвежий рынок ($k \approx 2.2$, $h \approx -0.2$, $N = 186$);
 - б) когерентный бычий рынок ($k \approx 2.2$, $h \approx 0.2$, $N = 186$).

В итоге мы сравниваем получившуюся гистограмму с функцией распределения вероятностей и выбираем те значения параметров, при которых разница минимальна.

3. Тестирование системы торговли, основанной на распознавании фазы рынка

Предложим простую систему торговли, основанную на распознавании фазы рынка. Основная идея системы основана на избегании периодов, когда рынок ведет себя как случайный или хаотический, и попытке

инвестировать только тогда, когда поведение рынка напоминает когерентное.

Введем следующие торговые правила:

– покупка, когда $H > 0,58$ и $h > 0,0012$;

– продажа, если h становится отрицательным или нейтральным и H приближается к 0,5.

Значения H и h выбраны путем оптимизации стратегии.

Параметры модели вычислялись для последовательных месячных промежутков времени. Анализ проводился с августа 2014 по апрель 2015 года (рисунок).



Рисунок. Значение индекса EUR/USD с августа 2014 по апрель 2015

Полученные результаты торговли приведены ниже:

– 01.08.2014 – 01.09.2014. $H = 0,8$, $N = 100$, $i = 0,008$;

Фаза переходного состояния рынка. Покупку осуществлять в данной фазе нельзя;

– 01.09.2014 – 01.10.2014. $H = 0,6$, $N = 50$, $h = -0,002$;

Рынок переходит в фазу эффективного состояния. Данное состояние характеризуется высоким риском и низкой доходностью, поэтому покупать индекс не стоит;

– 01.10.2014 – 01.11.2014. $H = 0,5$, $N = 20$, $h = 0$;

Фаза эффективного рынка, удерживаем наблюдательную позицию;

– 01.11.2014 – 01.12.2014. $H = 0,65$, $N = 10$, $h = -0,004$;

Без изменений;

– 01.12.2014 – 01.01.2015. $H = 0,87$, $N = 15$, $h = -0,02$;

Наблюдаем переход рынка в когерентное «медвежье» состояние, хоть показатель Херста и удовлетворяет нашим условиям торговли, но параметр фундаментального смещения нет, поэтому не предпринимаем никаких действий;

– 01.01.2015 – 01.02.2015. $H = 0.66$, $N = 150$, $h = 0$;

Фаза переходного состояния рынка;

– 01.02.2015 – 01.03.2015. $H = 0.9$, $N = 50$, $h = 0.01$;

Наблюдаем переход состояния в фазу когерентного бычьего рынка. Управляющие параметры удовлетворяют нашей системе торговли, поэтому принимаем решение о покупке индекса по цене 1.05;

– 01.03.2015 – 01.04.2015. $H = 0.58$, $N = 40$, $h = -0.004$;

Рынок возвращается в состояние эффективного. Необходимо продать индекс по цене 1.12;

– 01.04.2015 – 01.05.2015. $H = 0.7$, $N = 250$, $h = -0.002$;

Без позиций. Рынок находится в фазе переходного состояния;

Таким образом, можно подвести итог нашей торговли. За период с августа 2014 по апрель 2015 года: индекс EUR/USD понизился с 1.34 до 1.12 (на 0.22 единиц значений индекса, то есть, если использовать стратегию «покупай и держи», то торговля оказалась бы невыгодной, прибыль не была бы получена, а только грозили нам убытки). За этот же период торговля по системе с учетом фазы рынка за счет попытки избегания случайных и хаотичных рынков, то есть с меньшим риском, дала накопленную прибыль 0.07 единиц индекса (таблица).

Таблица

Результат торговли по стратегии избегания хаотичных и случайных фаз рынка

| Индекс | Цена покупки | Цена Продажи | Прибыль/убыток |
|---------|--------------|--------------|----------------|
| EUR/USD | 1.05 | 1.12 | +0.07 |

Заключение

Данная статья посвящена выполнению теоретических и прикладных исследований в области прогнозирования финансовых рынков. Был вычислен показатель Херста, позволяющий отличать случайные ряды с независимыми приращениями от неслучайных рядов. В рамках гипотезы когерентных рынков в модели Веге-Изинга были численно подсчитаны значения числа участников рынка, степень настроя инвесторов и величина, отражающая внешние экономические условия. С помощью этих

показателей можно определить, в каком состоянии находится рынок, и предпринять соответствующие действия торговли. Приведенная в работе стратегия на примере динамики индекса EUR/USD за период с августа 2014 года по апрель 2015 года показывает практическую пригодность подхода, основанного на учете фазы рынка для торговли.

Список литературы

1. Hurst, H. E. Long-term Storage of Reservoirs / H. E. Hurst // Transactions of the American Society of Civil Engineers. – London, 1951. – Vol. 116. – P. 770-779.
2. Яновский, Л. П. Принципы, методология и научное обоснование прогнозов урожая по технологии «ЗОНТ»: монография / Л. П. Яновский. – Воронеж : Воронеж. гос. аграр. ун-т, 2000. – 376 с.
3. Vaga, T. The Coherent Market Hypothesis / T. Vaga // Financial Analysts Journal. – 1991. – Vol. 46. – № 6. – P. 36-49.

ПРИКЛАДНОЕ ПРОГРАММНОЕ СРЕДСТВО МОДЕЛИРОВАНИЯ МЕТОДА ВЫРАВНИВАНИЯ РЕСУРСОВ

Е. В. Резниченко
Студент
М. Г. Матвеев
Профессор

Введение

Согласно своду знаний по управлению проектами, проект – это некоторая задача с определенными исходными данными и требуемым результатом, обуславливающими способ её решения. Проект включает в себя замысел, средства его реализации и получаемые в процессе реализации результаты. Таким образом, целью проекта является создание уникальных продуктов, услуг или результатов.

Одним из важнейших управляемых аспектов проекта является материальный ресурс и, соответственно, система управления ресурсами выступает в ряду подсистем управления проектами в качестве основной. Проект, как система, функционирует в пределах жизненного цикла и имеет «вход», в качестве которого выступают все ресурсы проекта, комплекс взаимосвязанных процессов обработки этих ресурсов с учетом ограничений и критерияльных показателей с целью обеспечить их

оптимальное использование для достижения конечной цели управления проектом и «выход», являющийся результатом проекта.

В рамках управления проектом потоки материальных ресурсов отражают эффективность процессов управления проектом или сигнализируют о нарушениях в запланированном ходе работ. Таким образом, имея такую цель, как повышение эффективности функционирования системы управления проектом необходимо осуществлять управление ресурсами проекта.

1. Оптимизация распределения ресурсов

При анализе профилей использования ограниченных ресурсов определяется соответствие между пределами их потребления (возможностью) и потребностью для выполнения работы. В ряде случаев возникает ресурсный конфликт. Ресурсный конфликт – это ситуация, когда потребность в каком-либо ресурсе превышает его максимальный предел потребления [1]. Примером могут служить одновременно выполняемые работы и использование одного и того же ресурса.

Для оптимизации распределения ресурсов и, в частности, разрешения ресурсных конфликтов, используются методы выравнивания, которые учитывают пределы потребления ресурсов и позволяют использовать их наиболее эффективно [2]. Выравнивание ресурсов устраняет пики в использовании ресурсов и устанавливает уровень использования ресурсов ниже максимального предела, смещая выполнение некоторых работ на более поздние даты.

2. Цель эксперимента

Цель эксперимента в данной статье – создать программное средство, позволяющее строить сетевой график ограниченного по времени проекта с использованием одного типа ресурсов и проводить его автоматизированный анализ. В случае, когда потребность в конкретном типе ресурсов колеблется и управление затрудняется, а их использование может быть весьма неэффективным, то решать эту проблему необходимо, используя метод выравнивания ресурсов, который уравнивает или сглаживает потребность в ресурсах.

В данной задаче проект рассматривается как совокупность большого числа операций, зависящих друг от друга. Некоторые операции могут выполняться параллельно, некоторые лишь после окончания выполнения предыдущей операции.

Дробление операций не допускается. Это значит, что, если операция внесена в график, то полагают, что она будет непрерывно осуществляться до ее окончания; следовательно, нельзя начать операцию, потом остановить ее на какое-то время, а затем закончить ее.

Программное средство должно предоставлять следующие возможности:

1. построение вершинного графа проекта;
2. вычисление критического пути проекта и выделение его на графе;
3. построение схемы потребности в ресурсах и использование ресурсов (диаграмма Ганта);
4. построение графика загрузки ресурсов.

3. Анализ задачи

Проект представляет собой список операций. Каждая операция имеет одного или нескольких предшественников, длительность, количество затрачиваемых ресурсов, значение раннего старта, раннего финиша, позднего старта и позднего финиша.

Сетевой график – это динамическая модель производственного процесса, отражающая технологическую зависимость и последовательность выполнения комплекса операций, увязывающая их свершение во времени с учетом затрат ресурсов и стоимости операций. В вершинном типе сетевых графов операции представлены узлами графа, а стрелками изображаются их взаимосвязи [3].

Критической является операция, для которой не существует резерва времени между наиболее ранним сроком ее начала и наиболее поздним сроком ее окончания.

Путь в вершинном графе, соединяющий эти операции, называется критическим путем.

Метод выравнивания ресурсов применяется в случае, когда использование ресурса становится неэффективным из-за колебания потребности в нем. В сущности, данный метод приводит к задерживанию некритических операций, используя простои для снижения пика потребностей и восполняя их нехватку.

4. Пример использования приложения

После получения необходимых данных, а именно списка операций с их предшественниками и параметрами, заполняются все поля приложения (рис. 1).

После заполнения данных нужно нажать кнопку «Готово».

Результатом данного действия будет являться построенный вершинный граф проекта (рис. 2) с выделенным красным цветом критическим путем, построенная схема потребности и использования ресурсов (рис. 3), график загрузки ресурсов (рис. 4) наглядно отображающая применение метода выравнивания ресурсов, выровненный график загрузки ресурсов, показывающий результат применения метода выравнивания ресурсов.

Операции Количество дней: 20

| Название | Кол-во ресурсов | Длительность | Предшественники | Ранний старт | Ранний финиш | Поздний старт | Поздний финиш |
|----------|-----------------|--------------|-----------------|--------------|--------------|---------------|---------------|
| A | 3 | 2 | | 0 | 2 | 0 | 2 |
| B | 2 | 3 | A | 2 | 5 | 2 | 13 |
| C | 5 | 4 | A | 2 | 4 | 2 | 6 |
| D | 3 | 2 | A | 2 | 4 | 2 | 13 |
| E | 5 | 3 | C | 6 | 9 | 6 | 9 |
| F | 1 | 4 | E | 9 | 13 | 9 | 13 |
| G | 3 | 3 | F | 13 | 16 | 13 | 16 |
| H | 3 | 2 | F | 13 | 15 | 13 | 20 |
| I | 2 | 4 | F | 13 | 17 | 13 | 17 |
| J | 2 | 3 | B, D, G, H, I | 17 | 20 | 17 | 20 |

Добавить операцию
Готово

Рис. 1. Форма использования приложения на конкретном примере

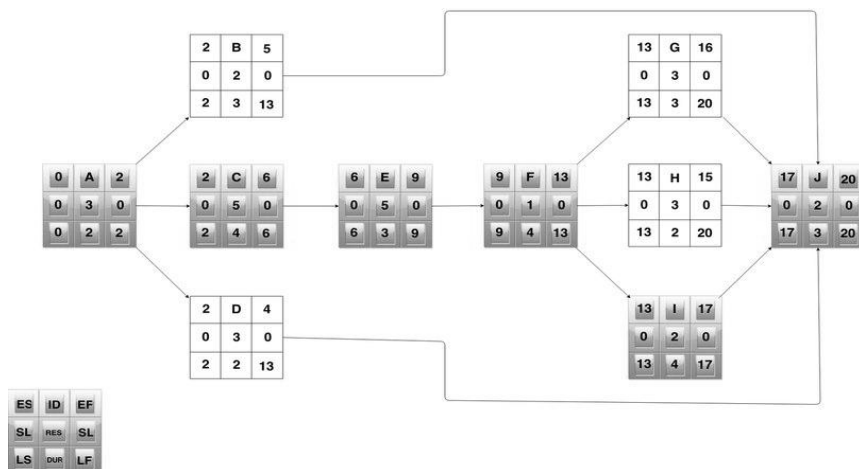


Рис. 2. Вершинный граф проекта

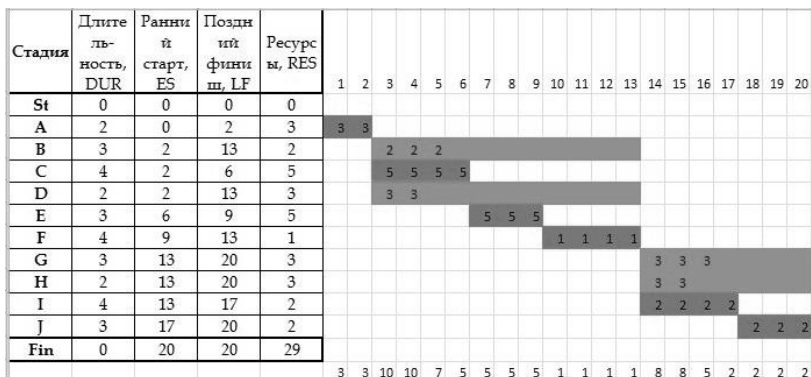


Рис. 3. Схема использования ресурсов

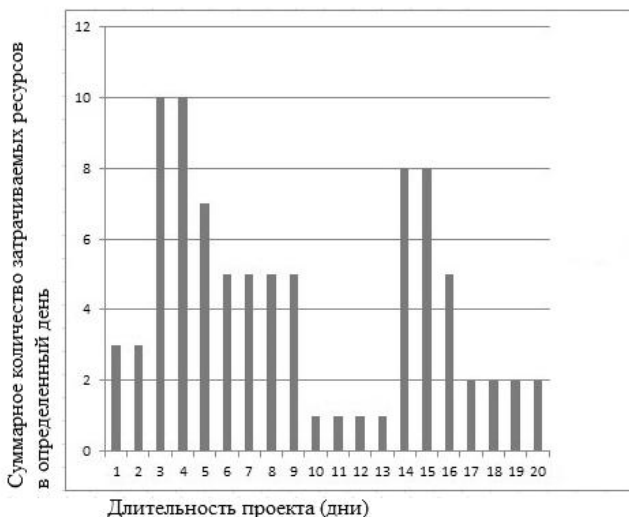


Рис. 4. График загрузки ресурсов

График загрузки ресурсов отображает зависимость длительности проекта и суммарного количества затрачиваемых ресурсов в определенный день.

Изучение графика позволяет заметить колебания потребности в ресурсах. Поскольку было заявлено, что проект ограничен по времени, целью будет сокращение пика потребностей в ресурсах и, таким образом, повышение степени их использования. Изучение схемы использования

ресурсов показывает, что 4 операции имеют простой, который можно использовать для сокращения пика — операции В, D, С, Н (рис. 5).

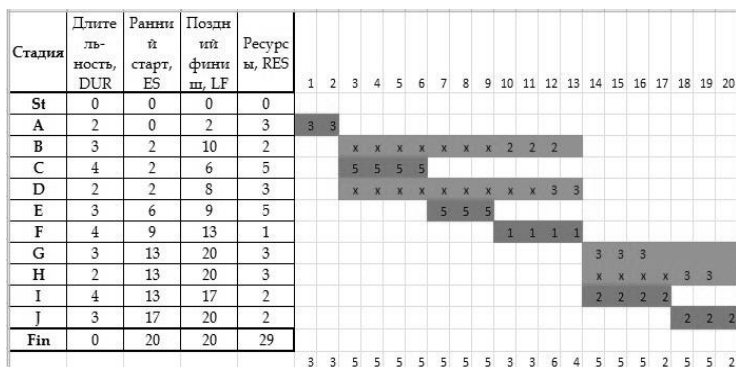


Рис. 5. Схема использования ресурсов

Любая из этих операций может быть задержана, чтобы сократить пик потребности в ресурсах. Выбор был сделан в пользу операций, которые имеют наименьший риск опоздания (В, D и С, поскольку они имеют наибольший простой). Это объясняется тем, что с такими операциями связан наименьший риск задержания проекта.

Можно заметить, что использование метода выравнивания ресурсов позволило сократить пики нагрузки в нагруженных операциях, при этом позволило увеличить нагрузку в незагруженных операциях. Таким образом, график был выровнен (рис. 6).

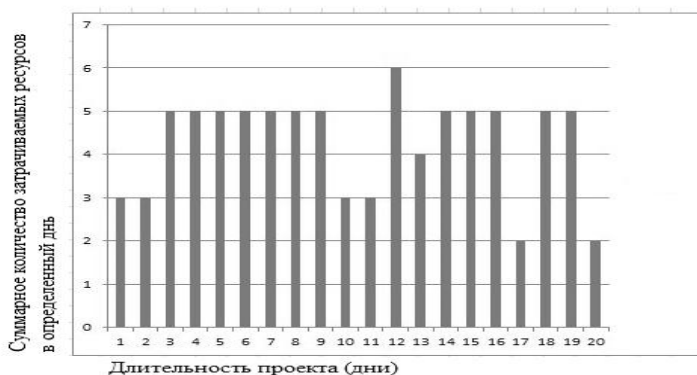


Рис. 6. График загрузки ресурсов

Заключение

Разработанное приложение позволяет наглядно отображать выполнение операций, а также знать и контролировать критический путь проекта, что позволяет не затягивать его реализацию, применять метод выравнивания ресурсов, позволяющий сокращать пики нагрузки в нагруженных операциях и увеличивать нагрузку в незагруженных операциях.

Таким образом, применение метода выравнивания ресурсов на практике позволяет значительно облегчить процесс управления проектом.

Список литературы

1. Кудрявцев, Е. М. Microsoft Project. Методы сетевого планирования и управления проектом / Е. М. Кудрявцев. – Москва : ДМК Пресс, 2005. – 240 с.
2. Брукс, Ф. Мифический человеко-месяц / Ф. Брукс, Х. Чапел. – Москва : Символ-Плюс, 2010. – 304 с.
3. Эддоус, М. Методы принятия решений / М. Эддоус, Р. Стэнсфилд. – Москва : Аудит, ЮНИТИ, 1997. – 590 с.

ПРИМЕРЫ АФФИННО-ОДНОРОДНЫХ ПОВЕРХНОСТЕЙ (ϵ, ϵ)-ТИПОВ В ПРОСТРАНСТВЕ \mathbb{C}^3

Е. В. Ремизова
Студент
А. В. Лобода
Профессор

1. Общая схема изучения однородности

Содержание данной работы связано с задачей полного описания аффинно-однородных вещественных гиперповерхностей 3-мерного комплексного пространства \mathbb{C}^3 . Эта сложная задача эффективно решается в последние годы (см. [1]-[3]) за счет выделения различных частных типов среди всех изучаемых поверхностей. Разбиение всех поверхностей на типы связано с использованием аффинных канонических уравнений рассматриваемых аналитических многообразий. Согласно [1], мы рассматриваем вещественно-аналитическую поверхность $M \subset \mathbb{C}^3$, заданную вблизи начала координат уравнением специального вида

$$v = (|z_1^2| + |z_2^2|) + (\varepsilon_1 z_1^2 + \varepsilon_2 z_2^2) + (\varepsilon_1 \bar{z}_1^{-2} + \varepsilon_2 \bar{z}_2^{-2}) + \sum_{k+l+2m \geq 3} F_{klm}(z, \bar{z}, u). \quad (1)$$

Здесь $z_1, z_2, w = u + iv$ – координаты в пространстве \mathbb{C}^3 , F_{klm} – однородный многочлен степени k по z , степени l по \bar{z} , степени m по u .

Типом поверхности с уравнением (1) будем называть пару $(\varepsilon_1, \varepsilon_2)$ с определенными условиями на них. Свойство однородности уже полностью изучено (см. [2]-[3]) в случае нескольких типов, ниже рассматриваются только поверхности $(\varepsilon, \varepsilon)$ -типа, где $\varepsilon \in \mathbb{R}$, $\varepsilon \neq 0$, $\varepsilon \neq 1/2$.

Мы следуем общей схеме изучения однородности, изложенной, например, в [2] и связанной с описанием матричных алгебр Ли, имеющих некоторые особенности в строении своих элементов. Все изучаемые алгебры состоят из матриц вида

$$\begin{pmatrix} A_1 & A_2 & A_3 & p \\ B_1 & B_2 & B_3 & s \\ a & b & c & q \\ 0 & 0 & 0 & 0 \end{pmatrix}. \quad (2)$$

а первым шагом схемы является оценка размерности таких алгебр Ли, отечающих поверхностям изучаемого типа.

Главный фрагмент в задаче поиска таких алгебр связан с исследованием системы квадратичных уравнений, описывающих условия замкнутости алгебры Ли относительно коммутатора $[Z_1, Z_2] = Z_1 Z_2 - Z_2 Z_1$.

Всего таких уравнений имеется 50. В уже описанных типах однородности удавалось обходиться (см., например, [3]) изучением лишь малых подсистем такой системы, выбранных из соображений их визуальной простоты. В нашем случае приходится изучить вопрос более детально.

2. Алгебра размерности 7

По аналогии с [4] несложно показать, что для аффинно-однородных поверхностей $(\varepsilon, \varepsilon)$ типа размерность $5 \leq \dim \leq 7$. При этом с точностью до аффинных преобразований имеется лишь 1-параметрическое семейство аффинно-однородных поверхностей

$$v = (|z_1^2| + |z_2^2|) + \varepsilon(z_1^2 + z_2^2 + \bar{z}_1^{-2} + \bar{z}_2^{-2}). \quad (3)$$

Базис семейства таких алгебр, зависящего от параметра ε , приведен ниже:

$$\begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 2i(1+2\varepsilon) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 & i \\ 0 & 0 & 0 & 0 \\ 2(1-2\varepsilon) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 2i(1+2\varepsilon) & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix},$$

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & i \\ 0 & 2(1-2\varepsilon) & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

Как и в рассмотренных других типах поверхностей, наиболее сложной для изучения является ситуация 5-мерных алгебр.

3. Квадратичные системы для случая 5-мерных алгебр

Базис пятимерной алгебры имеет общий вид:

$$\begin{pmatrix} A1_1 & A2_1 & A3_1 & 1 \\ B1_1 & B2_1 & B3_1 & 0 \\ 2i(1+2\varepsilon) & 0 & 2m_1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} A1_2 & A2_2 & A3_2 & i \\ B1_2 & B2_2 & B3_2 & 0 \\ 2(1-2\varepsilon) & 0 & 2m_2 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \quad (4)$$

$$\begin{pmatrix} A1_3 & A2_3 & A3_3 & 0 \\ B1_3 & B2_3 & B3_3 & 1 \\ 0 & 2i(1+2\varepsilon) & 2m_3 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} A1_4 & A2_4 & A3_4 & 0 \\ B1_4 & B2_4 & B3_4 & i \\ 0 & 2(1-2\varepsilon) & 2m_4 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} A1_5 & A2_5 & A3_5 & 0 \\ B1_5 & B2_5 & B3_5 & 0 \\ 0 & 0 & 2m_5 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

Система квадратичных уравнений, отвечающих 5-мерной алгебре Ли требуемого вида, связывает вещественные и мнимые части элементов 2×2 -блоков базисных матриц вида (4).

Квадратичные уравнения, возникающие в общей схеме, удобно представить в виде равенств нулю 50 квадратичных форм. В случае поверхностей $(\varepsilon, \varepsilon)$ -типа все эти формы зависят от 14 упомянутых вещественных элементов 2×2 -блоков (а остальные элементы выражаются через них на начальных шагах общей схемы).

При этом удобно переобозначить обсуждаемые 14 неизвестных и разбить их на две группы по 7 величин в каждой. Обозначим первые 7 переменных как x_i , последние 7 как y_i . После такого переобозначения 25 из 50 изучаемых квадратичных форм оказываются билинейными по двум группам переменных (и имеют, тем самым, "блочнo-антидиагональные" матрицы), еще 25 распадаются в "прямые" суммы (соответствующие матрицам "блочнo-диагонального" вида):

$$Q_{dia} = \sum a_m x_i x_j + b_m y_i y_j, \quad Q_{anti} = \sum c_m x_i y_j.$$

Отсюда естественным образом вытекает рассмотрение двух больших случаев: равенство нулю первой или второй группы переменных. В обоих случаях все 25 уравнений, соответствующих антидиагональным формам, становятся тождественно нулевыми. Таким образом, остается система из 25 уравнений относительно 7 неизвестных x_i и аналогичная система относительно y_i . Далее, рассмотрен случай $\forall i = 1..7 \quad x_i = 0$.

Примеры уравнений в системе:

$$z[1]: 2\varepsilon p_8 p_{10} - 4\varepsilon p_{10} p_{11} - 4\varepsilon p_{11} p_{12} + p_8 p_{10} + p_{10} p_{11} - 2p_{11} p_{12} = 0,$$

$$z[2]: -2\varepsilon p_9 p_{10} + 4\varepsilon p_{11}^2 + 4\varepsilon p_{11} p_{14} - p_9 p_{10} - 2p_{10}^2 + p_{10} p_{13} - \\ - 2p_{11}^2 + 2p_{11} p_{14} = 0,$$

$$z[4]: -2\varepsilon p_9 p_{11} - 4\varepsilon p_{10} p_{11} - 4\varepsilon p_{10} p_{14} + 2\varepsilon p_{11} p_{13} + 2\varepsilon p_{13} p_{14} - p_9 p_{11} - \\ - p_{10} p_{11} - 2p_{10} p_{14} - p_{11} p_{13} + p_{13} p_{14} = 0,$$

$$z[12]: -8\varepsilon^2 p_8 p_{10} - 12\varepsilon^2 p_8 p_{12} + 4\varepsilon^2 p_8 p_{13} - 12\varepsilon^2 p_9 p_{11} - 12\varepsilon^2 p_9 p_{14} + \\ + 16\varepsilon^2 p_{10} p_{11} + 12\varepsilon^2 p_{10} p_{14} + 4\varepsilon^2 p_{11} p_{12} - 4\varepsilon p_8 p_{10} - 12\varepsilon p_8 p_{12} - 6\varepsilon p_9 p_{11} - \\ - 12\varepsilon p_9 p_{14} - 4\varepsilon p_{10} p_{11} + 6\varepsilon p_{10} p_{14} - 4\varepsilon p_{11} p_{12} + 2\varepsilon p_{11} p_{13} - 3p_8 p_{12} - p_8 p_{13} - \\ - 3p_9 p_{14} - p_{10} p_{11} - 3p_{11} p_{12} - p_{11} p_{13} = 0.$$

4. Примеры алгебр и однородных поверхностей

Было найдено 7 типов решений системы. Каждому решению соответствует свой базис алгебры (полученные наборы матриц удовлетворяют определению алгебры). Пять из них являются подалгебрами 7-мерной алгебры, таким образом, они тоже будут порождать поверхности типа квадрики (3). Две алгебры, однако, соответствуют более интересным поверхностям.

Пример 1. Алгебра $v1-(1-2)$, являющаяся подалгеброй 7-мерной алгебры (4):

$$\begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 2i(1+2\varepsilon) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} m_2 & m_4 & 0 & i \\ -m_4 & m_2 & 0 & 0 \\ 2(1-2\varepsilon) & 0 & 2m_2 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 2i(1+2\varepsilon) & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix},$$

$$\begin{pmatrix} m_4 & -m_2 & 0 & 0 \\ m_2 & m_4 & 0 & i \\ 0 & 2(1-2\varepsilon) & 2m_4 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix},$$

Пример 2. Алгебра $v1-(1-6)$

$$\begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 2i(1+2\varepsilon) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 & i \\ 0 & 0 & 0 & 0 \\ 2(1-2\varepsilon) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & it_{24} & 0 & 1 \\ 0 & 2i(1+2\varepsilon) & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix},$$

$$\begin{pmatrix} \frac{-t_{24}}{1+2\varepsilon} & 0 & 0 & 0 \\ 0 & -t_{24} & 0 & i \\ 0 & 2(1-2\varepsilon) & \frac{-2t_{24}}{1+2\varepsilon} & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

После ее интегрирования получаем поверхность:

$$v = |z_1^2| + \varepsilon(z_1^2 + \bar{z}_1^2) + C |i - tz_2|^{\frac{2}{1+2\varepsilon}} + \frac{1+2\varepsilon}{t^2} (2y_2t - 1), \quad (t = t_{24} \neq 0).$$

Пример 3. Алгебра $v1-(2a)$

$$\begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -it^5 & -it(t^2 - 1) & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & i(1+2\varepsilon) & 0 & 0 \\ 0 & 0 & 0 & 0 \\ t^5(-1+2\varepsilon) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix},$$

$$\begin{pmatrix} 2t^2\varepsilon & 0 & 0 & 0 \\ 0 & t^2(1+2\varepsilon) & 0 & 0 \\ it^3(1-t^2) & 2it\varepsilon - it(t^2 - 1) & t^2(-1+2\varepsilon) & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} i & 0 & 0 & 0 \\ 0 & i & 0 & 0 \\ 0 & 0 & i & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

После ее интегрирования получаем поверхность:

$$v = \frac{y_2}{x_2} u + \frac{1}{tx_2} |z_2|^{1+2\varepsilon} \left(2Ct - (t^2 - 2t - 1) |z_2|^{1+2\varepsilon} \right) + \frac{-t}{2(1+2\varepsilon)x_2|z_2|^2} B(x, y),$$

$$B(x_1, y_1, x_2, y_2) = 2t^4(x_2^2\alpha_+ + 2\varepsilon x_1 x_2 y_1 y_2 - y_2^2\alpha_-) +$$

$$+ 2(x_1x_2 + y_1y_2)(1 + 2\varepsilon)(t^2 - 1) |z_2|^2,$$

$$\alpha_+ = |z_1^2| + \varepsilon(z_1^2 + \bar{z}_1^2), \quad \alpha_- = -|z_1^2| + \varepsilon(z_1^2 + \bar{z}_1^2).$$

Заключение

Изучение большой и сложно устроенной системы квадратичных уравнений оказывается возможным разбить на более простые случаи. В рамках одного из них получено полное описание всех возможных алгебр Ли и построены соответствующие этим алгебрам аффинно-однородные поверхности.

Список литературы

1. Лобода, А. В. Об одном семействе аффинно-однородных вещественных гиперповерхностей 3-мерного комплексного пространства / А. В. Лобода, А. С. Ходарев // Известия ВУЗов. Сер. Математика. – 2003. – № 10. – С. 38-50.
2. Лобода, А. В. Об аффинной однородности поверхностей трубчатого типа в \mathbb{C}^3 / А. В. Лобода, Т. Т. З. Нгуен // Москва : Тр. МИАН. – 2012. – Т. 279 – С. 102-119.
3. Лобода, А. В. О полном списке аффинно-однородных поверхностей $(\varepsilon, 0)$ -типов / А. В. Лобода, А. В. Шиповская // Известия ВУЗов. Математика. – 2015. – № 6. – С. 75-82.
4. Лобода А. В. О размерностях групп аффинных преобразований, транзитивно действующих на вещественных гиперповерхностях \mathbb{C}^3 / А. В. Лобода // Вестник ВолГУ. Серия 1 : Математика. Физика. – 2014. – №4 (23). – С. 11-35.
5. Ремизова, Е. В. Использование канонических уравнений в задаче об аффинной однородности / Е. В. Ремизова, А. В. Лобода; под ред. Д. Н. Борисова // Сборник студенческих работ ВГУ. Выпуск 8 – 2014. – С. 155-163.

ПРОГНОЗИРОВАНИЕ ВРЕМННЫХ РЯДОВ РЕГРЕССИОННЫМИ МОДЕЛЯМИ С НАСТРАИВАЕМЫМИ КОЭФФИЦИЕНТАМИ

О. О. Решетова

Студент

М. Е. Семенов

Профессор

Введение

В настоящее время финансовый рынок характеризуется значительной сложностью протекающих на нем процессов. Стандартные методы моделирования и прогнозирования финансовых временных рядов часто дают неудовлетворительные результаты. Следует отметить разрыв между действительными экономическими реалиями и экономическими теориями.

В этой связи построение моделей, позволяющих точно спрогнозировать значения финансовых рядов, как никогда актуально.

В настоящей работе проведено исследование временных финансовых рядов, на основе которого разработана регрессионная модель с адаптивно настраиваемыми (посредством искусственных нейронных сетей коэффициентами).

Задача прогнозирования сложных динамических систем является трудно формализуемой [1]. Прогноз таких систем опирается исключительно на выявление скрытых закономерностей в накопленных данных. Прогнозирование финансовых временных рядов – типичный пример такого рода задач. В научной литературе приводится большое количество математических моделей и методик для анализа финансовых рынков. Самыми распространенными для анализа считаются модели авторегрессии, скользящего среднего, а также интегрированные модели авторегрессии и скользящего среднего. В условиях априорной стационарности временных рядов эти методы дают вполне удовлетворительные результаты, однако это свойство (стационарность) наблюдается, в лучшем случае на небольших временных интервалах [2].

Другой современный подход заключается в использовании искусственных нейронных сетей. Как правило, методы, основанные на них достаточно хорошо адаптируются к изменяющимся характеристикам временных рядов.

Поэтому в настоящей работе используется метод прогнозирования, основанный на авторегрессионной модели, коэффициенты которой

адаптивно настраиваются посредством использования искусственных нейронных сетей [3].

Для настройки коэффициентов были использованы различные варианты нейронных сетей, такие как радиальные, многослойный персептрон [4], но наилучший результат удалось добиться используя искусственные нейронные сети с гистерезисной функцией активации.

Отличительной особенностью нейронной сети с ГФА является то, что в силу свойств гистерезиса, нейронная сеть обладает ассоциативной способностью и памятью.

1. Гистерезисная функция активации

Построим ГФА на основе входо-выходных соответствий, формализуемых следующим дифференциальным уравнением:

$$\frac{dx}{dt} = f(u, x), \quad (1)$$

где $u = u(t)$ – вход, а $x = x(t)$ – выход ($t \geq 0$).

Линии уровня правой части этого дифференциального уравнения приведены на рис. 1 (кривая Г).

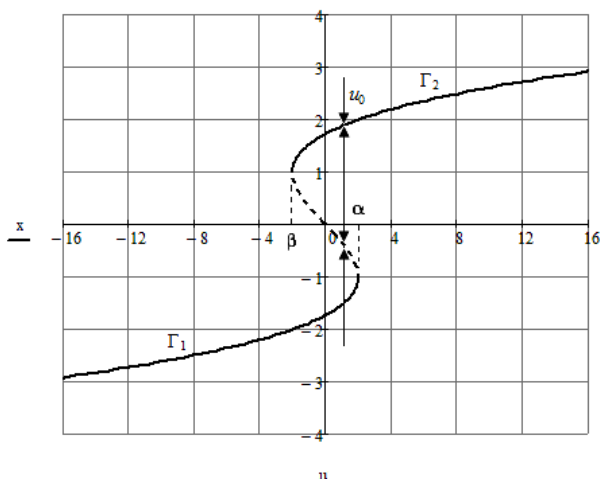


Рис. 1. Линии уровня $f(u, x) = 0$

В дальнейшем будем считать, что правая часть уравнения (1) определяется соотношением

$$f(u, x) = u - x^3 + 3x. \quad (2)$$

Очевидно, что решение данного уравнения, функция $x(t)$, является реакцией на воздействие $u(t)$ с запаздыванием $\delta = dx/dt$, то есть связь между входом и выходом является гистерезисным соотношением [4].

2. Описание модели

Ниже будем использовать модель описываемую следующим уравнением

$$y_t = \beta_0 + \beta_1 y_{t-1} + \beta_2 y_{t-2} + \dots + \beta_k y_{t-k} \quad (3)$$

(уравнение авторегрессии), где каждое значение будем считать адаптивно настраиваемым с помощью нейронной сети, на вход которой подается предыдущие значения временного ряда

$$\beta_i = a_{i1} y_{t-1} + a_{i2} y_{t-2} + \dots + a_{ik} y_{t-k}, \quad (4)$$

где $a_{i1}, a_{i2}, \dots, a_{ik}$ – весовые коэффициенты, вычисляемые нейросетевым способом.

При обучении основным критерием была минимизация функции ошибки

$$\sum (y_t - \hat{y}_t)^2 \rightarrow \min, \quad (5)$$

где y_t – значения прогнозируемого ряда, а \hat{y}_t – значение функции после адаптивной настройки коэффициентов, при помощи нейронной сети.

3. Результаты

В ходе исследований были рассмотрены различные модели построения нейронной сети, с различными функциями активации и различным количеством нейронов скрытого слоя. Ниже, будут показаны некоторые из них и описаны параметры настройки, при которой они были получены.

Линейная модель (рис. 2) представляется сетью без промежуточных слоев, которая в выходном слое содержит только линейные элементы. Веса соответствуют элементам матрицы, а пороги – компонентам вектора смещения.

Сеть типа радиальной базисной функции (рис. 3) имеет промежуточный слой из радиальных элементов, каждый из которых воспроизводит гауссову поверхность отклика. Выходной слой сети состоит из элементов с линейными функциями активации. Этот тип сетей не требует отдельного обучения.

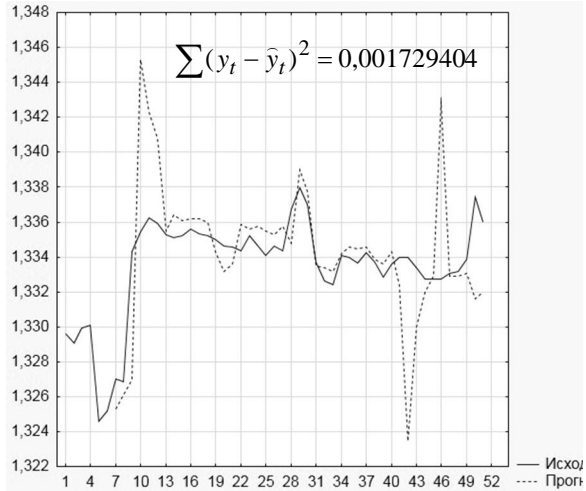


Рис. 2. Линейная нейронная сеть

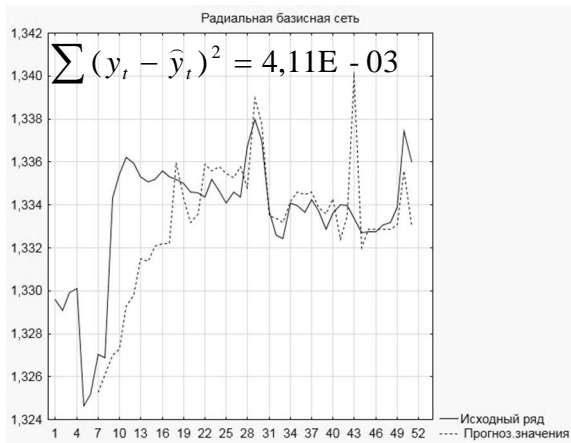


Рис. 3. Радиальная базисная сеть

Классическая многослойная НС (рис. 4) с пятью нейронами входного слоя, пятью нейронами скрытого и одним нейроном выходного слоя. В качестве функций активации входного и скрытого слоя использовалась линейная функция, для выходного слоя линейная функция «logsig».

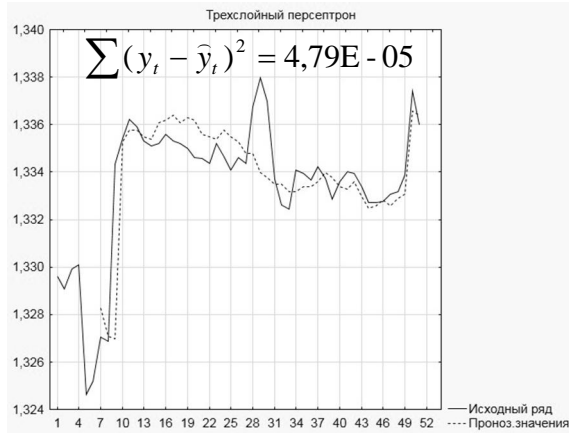


Рис. 4. Трехслойный персептрон

Последней приведем график нейронной сети с гистерезисной функцией активации (рис.5).

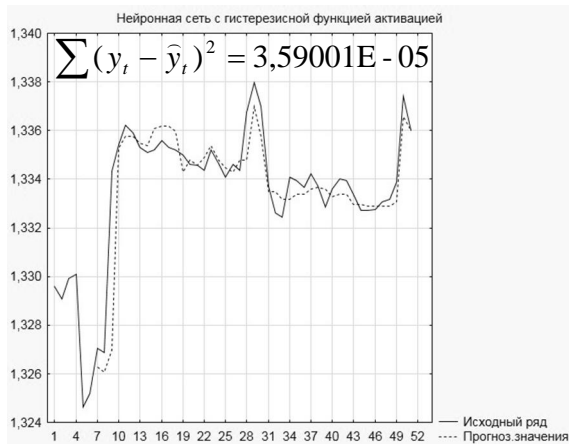


Рис. 5. Нейронная сеть с гистерезисной функцией активации

Опираясь на значения отклонения, а также следуя эмпирическим соображения, можно сделать вывод, что последняя конфигурация нейронной сети, благодаря правильной настройке параметров, позволяет строить наиболее точный прогноз по указанным данным.

Составим прогноз изменения значений функции временного ряда, в каждый такт времени. Данный прогноз строиться на основе архитектуры трехслойного персептрона, с теми же функциями активации (рис. 6).

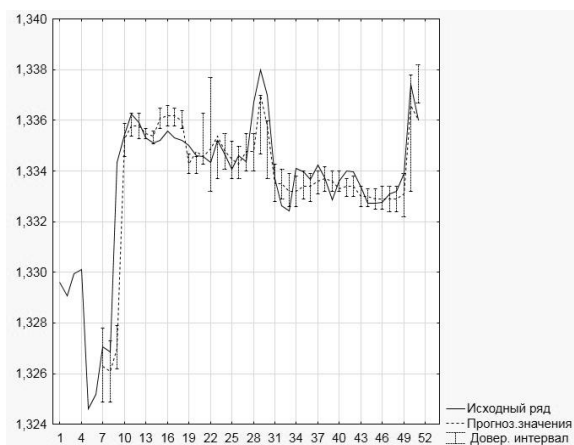


Рис. 6. Прогноз доверительных интервалов

Заключение

Проанализировав существующие модели, описывающие изменения котировок на финансовых рынках, было установлено, что одни из самых адекватных являются модели, построенные на основе алгоритмов обучения нейронных сетей. При этом оказалось, что наиболее подходящими (с точки зрения качества прогнозов) оказались нейронные сети с гистерезисной функцией активации.

Список литературы

1. Айвазян, С. А. Прикладная статистика и основы эконометрики: Учебник для вузов / С. А. Айвазян, В. С. Мхитарян. – Москва: ЮНИТИ, 2002. – 1022 с.
2. Низаметдинов, Ш. У. Анализ данных: учебное пособие / Ш. У. Низаметдинов, В. П. Румянцев. – Москва : НИЯУ МИФИ, 2012. – 288 с.
3. Клюкин, В. И. Нейросетевые структуры и технологии: учебное пособие/ В. И. Клюкин, Ю. К Николаенков. Издательско-полиграфический центр ВГУ, 2008. – 65 с.
4. Искусственные нейронные сети с гистерезисной функцией активации / А. М. Соловьев [и др.] // Теория и техника радиосвязи. – 2013. – № 2. – С. 102-110.

АФФИННО-ОДНОРОДНЫЕ ГИПЕРПОВЕРХНОСТИ ПАРАБОЛИЧЕСКОГО ТИПА С ИНДЕФИНИТНОЙ ФОРМОЙ ЛЕВИ

А. В. Руденко

Студент

А. В. Атанов

Ассистент

Введение

Задача описания поверхностей, однородных относительно действий различных групп преобразований (в частности, аффинных), активно решается с начала XX века (см., например, [1]). Соответствующие классификации поверхностей в одномерном и двумерном вещественных случаях известны, относительно недавно получены списки всех аффинно-однородных гиперповерхностей в пространствах R^3 и C^2 . Однако задача описания таких гиперповерхностей в C^3 остается открытой, к настоящему времени получены лишь некоторые фрагменты будущей классификации (см. [2],[3]).

В данной работе с использованием подхода, примененного в [2], [3], изучается свойство аффинной однородности одного типа индефинитных вещественных гиперповерхностей в C^3 . Строится классификация таких гиперповерхностей, для случая, когда размерность группы Ли аффинных преобразований, транзитивно действующей на них, больше 5.

1. Анализ младших компонент основного тождества

В начале дадим два определения, поясняющие активно используемые далее понятия.

Определение 1. Поверхность $M \subset C^3$ называется однородной относительно группы преобразований G , если G транзитивно действует на M , то есть некоторую точку из M можно перевести в любую другую точку этой же поверхности с помощью подходящего преобразования из группы G .

Определение 2. Под аффинной однородностью понимается существование в группе аффинных преобразований трехмерного комплексного пространства некоторой подгруппы Ли $G(M)$, транзитивно действующей на поверхности M .

Как было показано в [4], изучение аффинно-однородных indefинитных вещественных гиперповерхностей сводится к рассмотрению четырех типов, каждому из которых соответствует свое каноническое уравнение, определяющее вещественно-аналитическую indefинитную гиперповерхность пространства C^3 . В нашей работе рассматривается только один из упомянутых типов — параболический. Соответствующее каноническое уравнение в этом случае имеет вид

$$v = |z_1|^2 - |z_2|^2 + (z_1 + z_2)^2 + (\bar{z}_1 + \bar{z}_2)^2 + \sum_{r \geq 3} F_r(z, \bar{z}, u). \quad (1)$$

Здесь z_1, z_2, w — комплексные координаты в C^3 , $u = \text{Re}(w)$, $v = \text{Im}(w)$; $F_r(z, \bar{z}, u)$ — многочлен «веса» r , где $r = k + l + 2m$ (k — степень полинома $F_r(z, \bar{z}, u)$ по переменным $z = (z_1, z_2)$, l — по $\bar{z} = (\bar{z}_1, \bar{z}_2)$, m — по переменной u).

Отметим, что многочлен F_3 в разложении (1) за счет замены переменных можно сделать независимым от переменной u , поэтому в дальнейших рассуждениях будем считать, что $F_3 = F_3(z, \bar{z})$.

Мы изучаем однородность в локальном смысле, то есть свойство транзитивности должно выполняться лишь в малой окрестности фиксированной точки. В такой ситуации естественным является переход от самих преобразований к их инфинитезимальным (бесконечно малым, линеаризованным) аналогам. Строго говоря, такое действие означает переход от группы преобразований к соответствующей алгебре векторных полей, касательных к рассматриваемой поверхности. В силу линейности аффинных преобразований, упомянутые векторные поля тоже линейны и имеют следующий вид:

$$Z = (A_1 z_1 + A_2 z_2 + A_3 w + p) \frac{\partial}{\partial z_1} + (B_1 z_1 + B_2 z_2 + B_3 w + s) \frac{\partial}{\partial z_2} + (a z_1 + b z_2 + c w + q) \frac{\partial}{\partial w}, \quad (2)$$

где $A_k, B_k, a, b, c, p, s, q$ — комплексные константы.

Удобно также пользоваться матричным представлением поля (2):

$$\begin{pmatrix} A_1 & A_2 & A_3 & p \\ B_1 & B_2 & B_3 & s \\ a & b & c & q \\ 0 & 0 & 0 & 0 \end{pmatrix}. \quad (3)$$

Тогда линейное пространство g матриц, соответствующих линейным полям на поверхности M , образует матричную алгебру Ли, если оно замкнуто относительно операции коммутатора

$$[Z_1, Z_2] = Z_1 \cdot Z_2 - Z_2 \cdot Z_1. \quad (4)$$

Здесь $Z_j \cdot Z_k$ — обычное матричное умножение, а замкнутость g относительно коммутатора означает, что для любых двух матриц из g их коммутатор также принадлежит g .

Факт касания линейным векторным полем рассматриваемой однородной поверхности M записывается в виде основного тождества

$$\operatorname{Re}\{Z(\Phi)\}|_M = 0. \quad (5)$$

Суть использованного в [2, 3] подхода к изучению однородности гиперповерхностей заключается в получении информации о связях элементов матриц (3) друг с другом путем разложения аналитической функции, стоящей в левой части тождества (5), в ряд Тейлора в окрестности начала координат. Анализ весовых компонент до веса 4 включительно позволяет получить важную информацию об элементах матрицы (3).

В частности, анализ веса 0 позволяет установить, что константа q может быть только вещественной. Из рассмотрения веса 1 получаем следующие соотношения на элементы a и b :

$$a = 2i(2p + 2s + \bar{p}), \quad b = 2i(2p + 2s - \bar{s}).$$

Анализ компонент основного тождества, имеющих веса 2 и 3, дает существенно больше информации и позволяет установить следующий факт.

Теорема 1. Максимальная размерность группы Ли аффинных преобразований, транзитивно действующей на аффинно-однородной индефинитной вещественной гиперповерхности, задаваемой уравнением (1), допускает оценку

$$5 \leq \dim_R(g(M)) \leq 7.$$

Свободными элементами матрицы (3) остаются лишь $\operatorname{Re}(c)$ и $\operatorname{Im}(A_2)$. При этом система уравнений, возникающая при рассмотрении веса 3, позволяет выделить для более детального рассмотрения десять случаев, возникающих как ограничения на отдельные коэффициенты полинома F_3 в уравнении (1). При этом интересующий нас вариант аффинных преобразований, имеющих размерность группы Ли, указанную в теореме 1, больше 5, возможен только в четырех случаях из десяти (причем максимальная размерность 7 реализуется лишь в одном).

2. Классификация indefinitных гиперповерхностей

Дальнейший процесс классификации изучаемых гиперповерхностей состоит из двух этапов.

1) Выясняется, при каких значениях параметров совокупность заданных базисных матриц вида (3) является базисом алгебры Ли. Использование условия замкнутости относительно операции коммутатора позволяет составить для каждого из таких случаев системы нелинейных уравнений, решая которые мы сможем определить требуемые параметры. При этом приходится решать системы из многих десятков нелинейных уравнений.

2) Для каждого из набора полученных на первом этапе параметров строится гиперповерхность, касающаяся всех полей вида (3). Этот этап называется также интегрированием алгебры Ли.

Выполнение указанных шагов позволяет получить следующие классификационные результаты.

Теорема 2. Аффинно-однородная indefinitная гиперповерхность, определяемая уравнением (1) с 7-мерной группой (и алгеброй) является, с точностью до аффинных преобразований, квадратой

$$v = |z_1|^2 - |z_2|^2 + (z_1 + z_2)^2 + (\bar{z}_1 + \bar{z}_2)^2.$$

Теорема 3. Существуют четыре аффинно-однородные indefinitные гиперповерхности, определяемые уравнением (1) с 6-мерной группой Ли аффинных преобразований:

- 1)
$$v = 4(x_1 + x_2 + 1)\ln|x_1 + x_2 + 1| + 3t_1(x_1^2 + y_1^2 + x_1x_2 + y_1y_2)$$
- 2)
$$v = 2(t_1 - t_2)(t_1 + t_2)\ln|x_1^2 + y_1^2 - 2x_1 + 1| + 8t_1t_2\arctg\left(\frac{y_1}{x_1 - 1}\right) + 2t_1t_2(x_1y_2 - x_2y_1) - (t_1 - t_2)(t_1 + t_2)(x_1x_2 + y_1y_2)$$
- 3)
$$v = 2(t_1 - t_2)(t_1 + t_2)(x_1^2 + y_1^2 - 2x_1 + 1)\ln|x_1^2 + y_1^2 - 2x_1 + 1| + 8t_1t_2(x_1^2 + y_1^2 - 2x_1 + 1)\arctg\left(\frac{y_1}{x_1 - 1}\right) - 6(t_1^2 + t_2^2)(2t_1(x_1^2 + y_1^2 + x_1x_2 + y_1y_2) + t_2(x_1y_2 - x_2y_1))$$
- 4)
$$v = 2(1 + 6t_2y_1 + 6t_2y_2)\ln|1 + 6t_2y_1 + 6t_2y_2| - 9t_2^2(3x_1^2 + 4x_1x_2 + x_2^2 + 3y_1^2 + 4y_1y_2 + y_2^2)$$

Здесь t_1, t_2 — вещественные параметры.

Заключение

Была построена классификация аффинно-однородных вещественных гиперповерхностей параболического типа с индефинитной формой Леви для случая, когда размерность группы Ли аффинных преобразований, транзитивно действующей на этих поверхностях, больше пяти. Существует всего пять таких поверхностей, из которых одна имеет размерность соответствующей группы равную 7, а четыре остальных — равную 6. Для построения классификации была привлечена система символьной математики Maple, в которой были реализованы алгоритмы, соответствующие описанным выше этапам.

Список литературы

1. Cartan, E. Sur la géométrie pseudoconforme des hypersurfaces de deux variables complexes / E. Cartan // Ann. Math. Pura Appl. – 1932. – № 4 (11). – P. 17-90.
2. Лобода, А. В. Об аффинной однородности поверхностей трубчатого типа в S^3 / А. В. Лобода, Т. Т. З. Нгуен // Труды МИАН. – 2012. – Т. 279. – С. 102-119.
3. Атанов, А. В. Аффинно-однородные поверхности типа $(0,0)$ в пространстве S^3 / А. В. Атанов, А. В. Лобода // Матем. заметки. – 2015. – Т. 97. – № 2. – С. 309-313.
4. Данилов, М. С. Об аффинной однородности индефинитных вещественных гиперповерхностей пространства S^3 / М. С. Данилов, А. В. Лобода // Матем. заметки. – 2010. – Т.88. – № 6. – С. 866-883.

ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ УЛИЧНО-ДОРОЖНОЙ СЕТИ

С. В. Саввин

Студент

А. С. Беляев

Старший преподаватель

Введение

Успехи в области здравоохранения и развитие промышленности привели к значительному увеличению численности городского населения в XX веке. Например, за последние 100 лет число жителей Воронеж увеличилось более чем в 10 раз [1]. Ещё одной особенностью прошлого века стало развитие разнообразных средств для передвижения, среди которых центральное место занял автомобиль. С течением временем

доступность легковых автомобилей только росла, что привело к их большому распространению. В частности, в России за последние двадцать лет число личных автомобилей, приходящихся на тысячу человек населения, возросло более чем в 4 раза [2].

Таким образом, в начале нового тысячелетия дорожные системы городов столкнулись с огромными нагрузками, к которым они не были приспособлены. И если раньше транспортные проблемы были актуальны, в основном, для крупных мегаполисов, то в наши дни эта проблема актуальная и для сравнительно небольших городов.

Существуют различные пути решения транспортной проблемы, помимо непосредственного уменьшения числа автомобилей: строительство объездных дорог, расширение уже существующих путей, использование интеллектуальных средств управления дорожным потоком и т. д. Однако в основе принятия тех или иных решений лежит компьютерное моделирование улично-дорожных сетей, которое помогает оценить, как существующую ситуацию, так и способы её улучшения.

1. Цель

В данной статье рассматривается создание модели улично-дорожной сети с со светофорами на перекрёстках. Так как основная задача моделирования – оценка эффективности моделируемой системы, то в создаваемой модели должна быть предусмотрена возможность сбора разнообразной информации об автомобилях.

В последнее время всё большее распространение получают адаптивные (интеллектуальные) системы управления дорожным движением, которые изменяют режим работы светофоров в соответствии с текущей ситуацией на дороге. Поэтому для оценки подобных систем разрабатываемая модель должна позволять изменять режим работы светофоров во время своей работы.

2. Основные сущности

Улично-дорожная сеть является достаточно сложной системой, которая имеет случайные процессы. Таким образом, для создания модели целесообразно использовать методы имитационного моделирования.

Для создания имитационно модели необходимо первоначально определить основные сущности и связи между ними [3]. Для создаваемой модели основными сущностями будут являться транспортные потоки, перекрёстки, светофоры и транспортные средства.

Транспортный поток характеризуется координатами своего начала и конца, а также списком транспортных средств, которые движутся по нему в текущий момент времени.

Перекрёсток является пересечением нескольких потоков и содержит в себе информацию о светофорах, которые управляют этими потоками.

Каждый светофор управляет движением на одном транспортном потоке на некотором перекрёстке (число светофоров равно числу пересекающихся на перекрёстках потоков). Светофор характеризуется своим текущим сигналом (зелёным, жёлтым или красным) и временем до переключения.

Транспортное средство описывается своей одномерной координатой в рамках своего транспортного потока, скоростью, ускорением, номером полосы для движения и длиной. Другие параметры автомобилей определяются используемыми моделями движения.

3. Модели движения транспортных средств

Для большей достоверности создаваемой модели необходимо реализовать в ней реалистичное движения транспортных средств. Данная задача разбивается на две подзадачи:

- движение транспортных средств в пределах одной полосы для движения;
- перестроение транспортных средств между полосами.

Для решения первой подзадачи использовалась математическая модель разумного водителя (Intelligent Driver Model) [4]. Данная модель относится к классу моделей с лидером. Лидером для автомобиля является находящееся непосредственно перед ним транспортное средство. На каждом шаге моделирования осуществляется перерасчёт ускорения в зависимости от расстояния до лидера и его скорости, а также параметров автомобиля:

- максимальной скорости движения;
- желаемого ускорения;
- комфортного торможения;
- минимального расстояния до лидера;
- временного интервала до лидера.

Важной особенностью модели разумного водителя является тот факт, что с её помощью можно моделировать остановку перед запрещающим сигналом светофора. Для этого необходимо поставить в соответствие каждому светофору с красным сигналом фиктивный неподвижный автомобиль, что и приведёт к плавной остановке транспортных средств перед перекрёстком.

Для реализации перестроений применялась модель MOBIL (Minimizing Overall Braking Induced by Lane change) [5]. Особенностью этой модели является то, что она основана на вычислении ускорений при

помощи модели разумного водителя. Такой подход позволяет значительно сократить число используемых параметров.

Согласно модели MOBIL, перестроение осуществляется, если удовлетворены критерии безопасности (можно ли совершить перестроение) и заинтересованности (есть ли смысл в совершении перестроения).

Модель MOBIL характеризуется двумя параметрами:

- максимально возможное торможение;
- коэффициент «вежливости» (от 0 до 1; при 0 транспортное средство заботится только о своей выгоде, при 1 – только о выгоде остальных автомобилей).

4. Моделирующий алгоритм

После определения математической модели системы необходимо описать её моделирующий алгоритм. Данная имитационная модель имеет три вида активных объектов:

- транспортные потоки, которые создают транспортные средства;
- светофоры, которые переключают свои сигналы;
- транспортные средства, которые движутся по транспортным потокам.

На каждом шаге моделирования данные объекты действуют по алгоритмам, представленным на рис. 1.

5. Реализация и пример использования модели

Имитационная модель реализована на языке Scala и запускается при помощи виртуальной машины Java версии не ниже 7. Для распараллеливания вычислений была использована библиотека Akka, а для визуализации работы – библиотека Swing.

Для упрощения сбора различной информации о транспортных средствах реализована возможность создавать для событий пользовательские функции-обработчики. Всего предусмотрено три вида событий:

- события модели (запуск, завершение шага моделирования, остановка);
- события светофоров (переключение сигнала);
- события транспортных средств (создание, перестроение, остановка, проезд перекрёстка и т.д.).

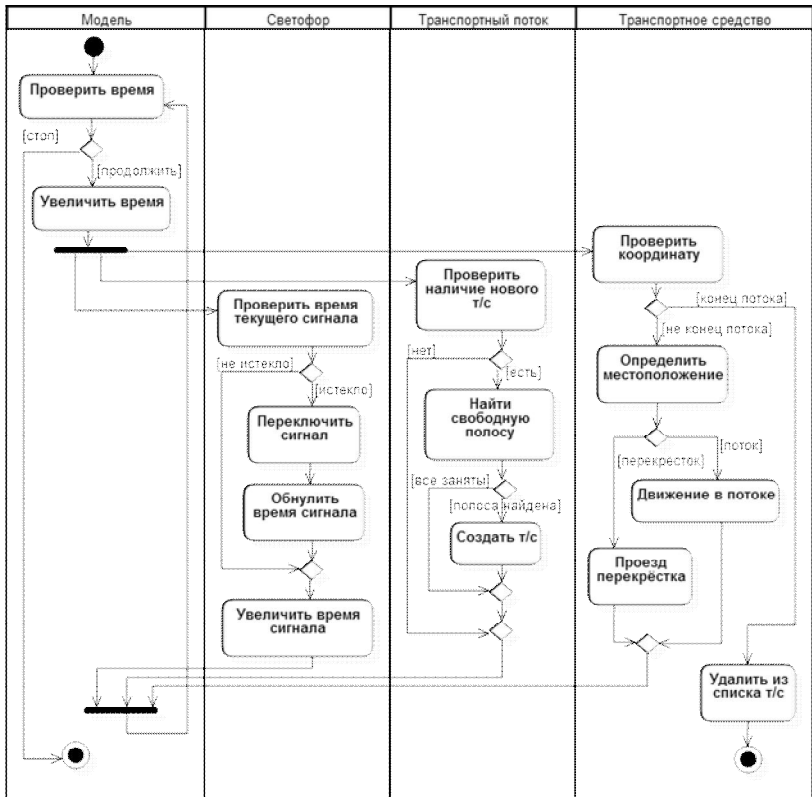


Рис. 1. Диаграмма активности каждого шага моделирования

В качестве примера использования созданной модели рассмотрим небольшую уличную сеть, состоящую из трёх улиц и двух перекрёстков. Все улицы имеют одинаковую и постоянную интенсивность движения (0,6 автомобилей/сек) через свои концы и по три полосы для движения в обе стороны. Все светофоры имеют одинаковый режим работы: две фазы по 30 секунд каждая. Визуализация работы такой модели показана на рис 2. В этот момент времени на горизонтальной улице горят запрещающие сигналы светофоров, и поэтому перед перекрёстками скапливаются очереди из автомобилей.

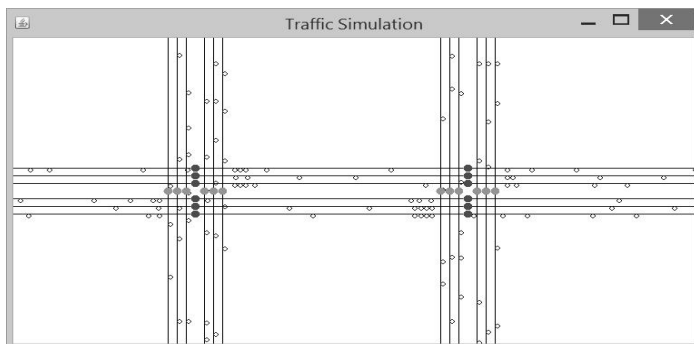


Рис. 2. Пример работы модели

Заключение

Данная статья посвящена созданию имитационной модели улично-дорожной сети со светофорами на перекрёстках, которая позволяет собирать разнообразную информацию о движении транспортных средств путём обработки событий. Приведены используемые математические модели движения транспортных средств по полосе и перестроения между полосами, а также используемый моделирующий алгоритм. Созданная модель позволяет динамически управлять режимом работы светофоров, и, следовательно, в дальнейшем может использоваться для моделирования работы интеллектуальных систем управления дорожного движения.

Список литературы

1. Шулепова, Э. А. Историко-культурное наследие Воронежа: материалы Свода памятников истории и культуры Российской Федерации / Э. А. Шулепова. – Воронеж : Центр духовного возрождения Чернозёмного края, 2000. – 575 с.
2. Число собственных легковых автомобилей на 1000 человек населения по субъектам Российской Федерации. Федеральная служба государственной статистики [Электронный ресурс]. – Режим доступа : http://www.gks.ru/scripts/db_inet/dbinet.cgi?pl=1316016
3. Сирота, А. А. Компьютерное моделирование и оценка эффективности сложных систем / А. А. Сирота. – Москва : Техносфера, 2006. – 280 с.
4. Treiber, M. Traffic Flow Dynamics / M. Treiber, A. Kesting. – New York : Springer, 2013. – 503 p.
5. Treiber, M. General Lane-Changing Model MOBIL for Car-Following Models / M. Treiber, A. Kesting, D. Helbing // Transportation Research Record Journal of the Transportation Research Board. – 2007. – N 7. – P. 19-29.

РАЗРАБОТКА И РЕАЛИЗАЦИЯ МЕТОДА ПРОГНОЗИРОВАНИЯ РАСПРЕДЕЛЕННОЙ ПО ГЛУБИНЕ ТВЕРДОСТИ СТАЛИ

И. Ю. Савченко

Студент

М. Г. Матвеев

Профессор

Введение

Как известно, металлургическая промышленность является одной из ведущих, уступая лишь топливно-энергетической. Предприятия и организации металлургии размещены более чем в 20 регионах страны. Выплавка стали осуществляется по спецификациям (Таблица 1), которые предоставляет заказчик. Спецификации включают в себя информацию о форме изделия по окончании выплавки, диапазон, в котором могут изменяться химические примеси и значения прокаливаемости – твердость выплавленной стали на различной глубине от поверхности [1, 2].

Таблица 1

Пример заказной спецификации

| Допустимый диапазон изменения химического состава, % | | | Допустимый диапазон изменения твердости, HRC | | |
|--|------|-------|--|------|------|
| Состав | min | max | h, мм | min | max |
| C | 0,18 | 0,37 | 3 | 52,9 | 57,5 |
| Mn | 0,2 | 0,8 | 5 | 52,9 | 57,5 |
| Si | 0,15 | 0,35 | 7 | 52,1 | 57,5 |
| P | 0 | 0,03 | 9 | 50,9 | 56,3 |
| S | 0,03 | 0,035 | 11 | 49,2 | 55,9 |
| Cr | 0,9 | 1 | - | - | - |
| Ni | 0 | 0,2 | - | - | - |
| Al | 0,02 | 0,06 | - | - | - |
| Cu | 0 | 0,3 | - | - | - |

Для управления процессом выплавки большинство металлургических предприятий имеет сегодня ту или иную систему математических моделей, описывающих зависимость распределенной по глубине твердости от количественного состава химических компонент [3, 4].

1. Аналитическая часть

На сегодняшний день множество предприятий чёрной металлургии при управлении выплавкой стали использует систему прогнозирования распределенной по глубине h твердости, основанная на кусочно-линейной аппроксимации нелинейной зависимости распределенной твердости $y(h)$ от химического состава $\bar{x} = (x_1, x_2, \dots, x_n)$ [3]. В этой системе весь допустимый диапазон изменения концентраций химических компонент разбит на интервалы.

Основываясь на статистические данные при анализе плавов, для определенных интервалов химических элементов, получены линейные регрессионные зависимости, которые имеют следующий вид.

$$\bar{y}(h) = \bar{a}_0 + A\bar{x},$$

где $\bar{x} = (x_1, x_2, \dots, x_n)$ – вектор значений концентраций химических компонент плавки; \bar{a}_0, A – вектор и матрица оценок параметров регрессионной модели соответственно.

Процесс управления твердостью стали выполняется по следующему сценарию. Эксперт – высококвалифицированный специалист, далее оператор, опытным путем, опираясь на данные из заказной спецификации, подбирает наиболее подходящий состав химических элементов. Для прогноза твердости, в момент подбора коэффициентов химических примесей, оператор использует одну из 8 моделей системы [3].

В итоге выбирается именно та модель, которая по мнению эксперта удовлетворяет критерию близости сочетания концентраций химических примесей.

Основными недостатками существующего подхода к управлению являются:

- высокая вероятность ошибочного принятия решения экспертом при выборе наиболее адекватной модели для прогноза;
- большая доля трудозатрат при выполнении перебора нужных концентрации химических примесей и возможность выбора концентраций, не подходящих под описание, но приближённых к заданному распределению твёрдости.

2. Реализация

Чтобы устранить вышеупомянутые недостатки системы, нужно решить следующие задачи:

- автоматизировать процесс выбора наиболее подходящей модели прогнозирования твердости стали с точки зрения заказной спецификации;
- автоматизировать решение задачи выбора требуемого химического состава, обеспечивающего заданную в спецификации

твердость стали, в условиях стохастичности параметров существующих моделей

Для того чтобы выполнить решение задачи прогнозирования, иными словами, чтобы построить математическую модель прогноза, основываясь на статистических данных технологических измерений твердости на различных глубинах и с различной концентрацией химических компонентов, могут быть использованы методы универсальной аппроксимации, основанные на нечетких продукционных правилах, в частности, модель Токаги-Суджено-Канга (TSK) [4].

Для разработки программного средства была использована треугольная функция принадлежности (рис.1).

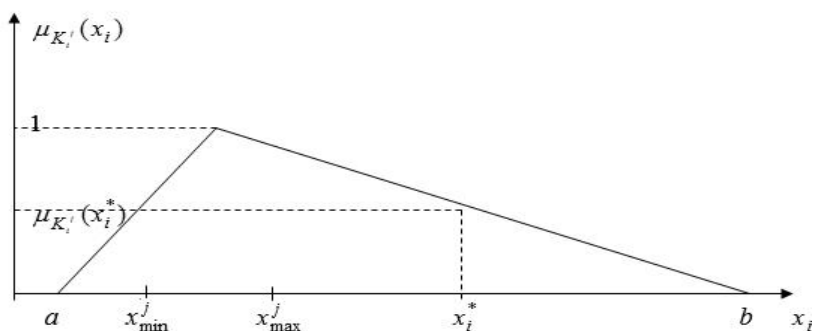


Рис. 1. Треугольная функция принадлежности

Функция принадлежности рассчитывается следующим образом:

$$\mu_{K_i^j}(x_i) = \begin{cases} \frac{2(x_i - a)}{x_{\min}^j + x_{\max}^j - 2a}, & \text{если } a \leq x_i \leq \frac{x_{\min}^j + x_{\max}^j}{2} \\ \frac{2(x_i - b)}{x_{\min}^j + x_{\max}^j - 2b}, & \text{если } \frac{x_{\min}^j + x_{\max}^j}{2} < x_i \leq b \\ 0, & \text{если } x_i \leq a \text{ или } x_i \geq b \end{cases}$$

где x^j – концентрация химической примеси, которую выставляет эксперт; x_{\min}^j, x_{\max}^j – границы диапазона подмножества (в нашем случае, диапазон для химических примесей, предоставляемый заказчиком); a, b – границы диапазона множества (в нашем случае, множество, которое

получается в результате суммирования диапазонов каждого хим. элемента из 8-ми моделей).

На основе описанного выше метода было реализовано программное средство, которое позволяет сделать прогноз твердости стали на заданных глубинах (рис. 2). Принцип работы программы заключается в следующем:

- выбирается количество уровней для прогноза и назначаются значения глубин прокаливаемости (область «Прокаливаемость», изображенная на рис. 2);

- устанавливаются значения ограничивающего коридора, исходя из спецификации заказчика (область «Ограничение», изображенная на рис. 2);

- выставляются коэффициенты, которые будут участвовать в прогнозе и изменение которых позволит управлять прогнозом (область «Хим. примеси», изображенная на рис. 2);

- производится визуализация прогноза твердости (монотонно-убывающая функция, расположенная между двумя ограничивающими кривыми, изображенная на рис. 2).

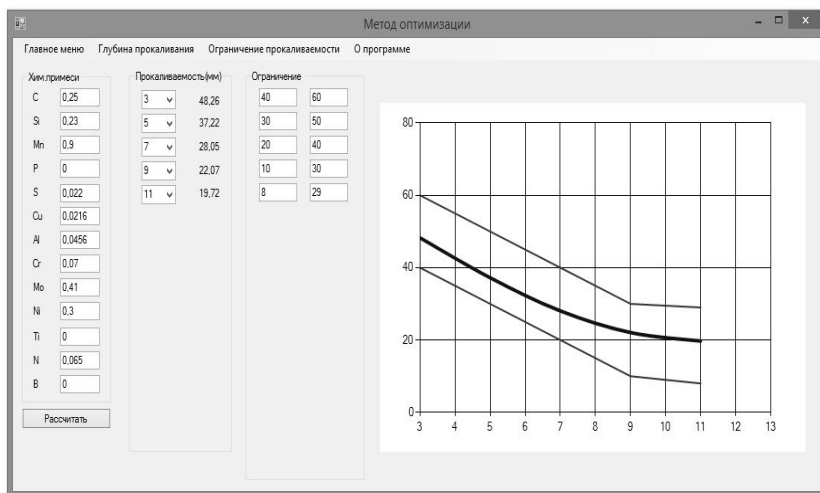


Рис. 2. Прогнозирование твердости стали

Заключение

В итоге было реализовано программное средство, позволяющее минимизировать возникновение ошибок при выборе той или иной модели для выплавки стали, а также сократить трудозатраты, которые раньше расходовались на подбор концентраций химических примесей.

Список литературы

1. Контроль качества термической обработки стальных полуфабрикатов и деталей: Справочник / Под ред. проф. В. Д. Кальнера. – Москва : Машиностроение, 1984. – 384 с.
2. Лахтин, Ю. М. Металловедение и термическая обработка металлов: учеб. пособие / Ю. М. Лахтин. – Москва : Металлургия, 1976. – 258 с.
3. Совершенствование статистического контроля механических свойств листового проката из стали 17Г1С-У / А. М. Степашин, В. В. Зырянов, С. Е. Павлов [и др.] // Сталь. – 2005. – № 3. – С. 96-98.
4. Оптимизация прокаливаемости и состава термоулучшаемой борсодержащей стали / М. В. Бобылев, А. М. Ламухин, О. А. Кувшинников [и др.] // Сталь. – 2002. – № 7. – С. 68-71.

АВТОМАТИЗАЦИЯ ПРОЦЕССА «ОТПУСК МАТЕРИАЛА» В СИСТЕМЕ SAP ERP С ТОЧКИ ЗРЕНИЯ ПРОЦЕССА ЗАКУПКИ

Ю. И. Сафронова

Студент

А. П. Толстобров

Доцент

Введение

SAPR/3 – это многокомпонентная программная система, разработанная по технологии клиент/сервер, которая используется для управления внешними и внутренними ресурсами предприятия.

Данная система позволяет отражать все бизнес-процессы, происходящие на предприятии, в реальном времени, синхронизировать их между различными модулями системы и предоставлять пользователям актуальную и точную информацию в любой момент времени.

Главной особенностью SAP является то, что данная система конфигурируема, то есть при внедрении на реальное предприятие, ее необходимо настроить под конкретные бизнес-процессы [1]. В большинстве случаев для того или иного процесса в системе уже существует некая логика, которую достаточно просто включить. Но иногда требуется изменить стандартные решения для более точного отражения процессов работы предприятия.

1. Анализ стандартного функционала

В данной статье будет рассмотрена настройка бизнес-процесса модуля Material Management (ММ) под определенные требования. Модуль ММ (Управление материальными потоками) является одним из основных компонентов в Логистике. Он включает в себя такие аспекты, как закупка, получение и хранение материала, планирование потребности в материалах, инвентаризация и многие другие процессы. Данный модуль тесно связан с такими модулями как Финансы, Контроллинг, Продажи, Производство [2].

Стандартная цепочка закупок предполагает наличие следующих основных элементов, представленных на рис. 1:

1. *Заявка на закупку.* Бизнес (планировщик) определяет потребность в материале на некую дату. Если по данным, предоставленным в отчетах SAP, необходимого запаса к нужной дате не будет, то оформляется заявка на закупку этого материала. В заявке фиксируется количество материала, дата поставки и завод, где возникла потребность [3].

2. *Заказ на поставку.* По созданной ранее заявке и конкретному договору с поставщиком создается заказ на поставку. В заказе четко определяются условия поставки: поставщик, цены, количество, условия доставки (даты, место разгрузки, способ перемещения груза).

Созданный заказ утверждается и подписывается с двух сторон – со стороны продавца и покупателя.

3. *Поступление материала.* В момент реального поступления материала на склад бизнес формирует документ поступления материала. В нем указывается реальное количество поступившего груза, дата и место отгрузки.

4. *Оплата.* После прихода груза в системе формируется счет-фактура на оплату материала. В ней указывается заказ на поставку, договор, реквизиты поставщика, дата и сумма оплаты. По данному документу будут сформированы последующие документы оплаты, по которым и будет выполнен перевод средств с одного счета на другой.

Поступивший материал хранится на определенном складе, впоследствии он может быть продан, передан в производство, списан на конкретных работников предприятия.

В данной работе более подробно будет рассмотрен случай списания материала.

По стандарту, предусмотренному в SAP списать материал можно в транзакции MIGO в один шаг. Для этого достаточно ввести номенклатурный номер и партию материала, его количество и место возникновения затрат (куда поступит списанный материал).



Рис. 1. Стандартная цепочка закупки в SAP

С точки зрения модуля ММ такое списание материала будет практически не отслеживаемым и невозможным для контроля и планирования. Поэтому далее будет рассмотрен способ настройки системы, позволяющий более точно фиксировать все возникающие потребности в материалах и закрывать их вовремя.

2. Постановка задачи

Были сформулированы следующие требования к работе системы:

1. отслеживание длинной цепочки прохождения документов перед фактическим отпуском материала (заявка, заказ, исходящая поставка и документ списания материала);
2. автоматическое создание исходящей поставки из заказа на списание (системой);
3. возможность согласования и отказа от списания материала ответственными лицами на первом этапе (заявка);
4. отражение потребности при прогоне ППМ (планирование потребности в материалах);
5. документирование сопровождения этапов бизнес-процесса (печатные формы и электронные документы).

В результате новый бизнес-процесс должен представлять из себя следующую цепочку, представленную на рис 2.



Рис. 2. Планируемая цепочка закупки/списания в SAP

3. Реализация

Для реализации работы такой цепочки в системе SAP необходимо выполнить следующие настройки:

- создание новых видов документов для заявки на списания, заказа на списание, исходящей поставки;
- настройка связей между новыми видами документов для корректной работы цепочки;
- настройка поведения системы в зависимости от действий пользователя на каждом из шагов;
- настройка автоматического создания исходящей поставки из заказа на списание в момент его первого сохранения;
- настройка автоматического выбора партии материала в исходящую поставку;
- настройка возможности согласования/отказа от списания на этапе создания заявки на списание;
- включение заявок на списание в список потребностей, фиксируемых при прогоне ППМ;
- разработка и настройка печатной формы М-11 для исходящей поставки.

В результате выполненных разработок и настроек был сформирован следующий алгоритм работы системы, представленный на рис. 3.

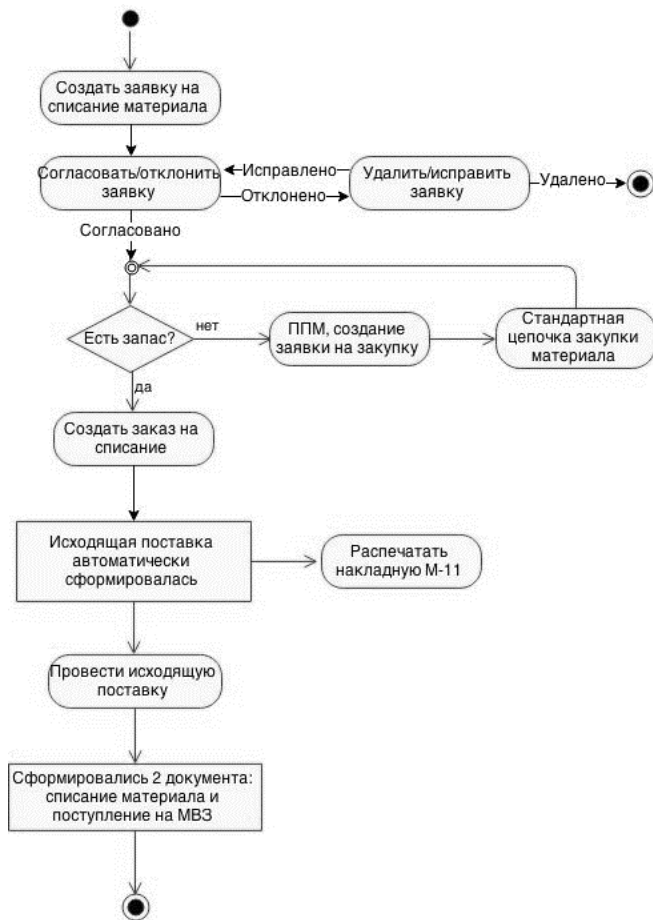


Рис. 3. Алгоритм реализованного бизнес-процесса «Отпуск материала»

Заключение

В данной статье рассматривался модуль Material Management системы SAP ERP, анализировался стандартный функционал процессов закупки и списания материала с точки зрения этого модуля и рассматривался способ настройки системы под определенные требования.

Новый бизнес-процесс полностью соответствует заявленным требованиям, в результате получены следующие результаты:

- до фактического отпуска материала создаются новые документы, которые позволяют отслеживать ошибки на ранних этапах и блокировать

заявку в случае отказа в согласовании на отпуск материала, качественно планировать потребности в материалах;

– отпуск материала сопровождается электронными и печатными формами;

– исключены ошибки при выборе партии для списания, настроен автоматический выбор партии по принципу FIFO («стек», первая поступившая партия списывается первой).

В подведении итогов можно сказать, что система SAP позволяет реализовать наиболее точное отражение всех процессов, происходящих на предприятии. Если стандартный функционал SAP не отражает какой-либо процесс в полном объеме, его можно изменить или создать абсолютно новый. Благодаря этому любое предприятие крупного и среднего бизнеса может внедрить данную систему в свое производство для повышения его эффективности.

Список литературы

1. SAP MM Handbook (SAP Books) / L. Kogent [at al.]. – Burlington : Jones & Bartlett Learning, 2010. – 678 p.

2. Schulz, O. Using SAP: A Guide for Beginners and End Users / O. Schulz. – Quincy : Galileo Press, 2012. – 388 p.

3. Фатерян, А. Краткое руководство по внедрению SAP R/3 [Электронный ресурс] : электрон. журн. / А. Фатерян. – Режим доступа : <http://fatheryan.narod.ru/SAPconsultantguide.html>

АНАЛИЗ И ИССЛЕДОВАНИЕ ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ ДЛЯ РЕШЕНИЯ ЗАДАЧИ ПРОГНОЗИРОВАНИЯ ВРЕМЕННЫХ РЯДОВ

А. Б. Тимофеев,

Студент

Е. А. Сирота

Доцент

Введение

Задача прогнозирования временных рядов является особенно актуальной в последнее время, когда стали доступными мощные средства сбора и обработки информации.

На сегодняшний день широко применяются различные методы прогнозирования, такие как статистические методы, алгоритмы экстраполяции экспериментальных данных и другие.

Искусственные нейронные сети являются альтернативой традиционным методам решения нелинейных задач прогнозирования временных рядов, из-за наличия у них обобщающих свойств, распараллеливания ими процесса обработки информации, а также таких свойств как отказоустойчивость, адаптивность, нелинейность [1].

1. Теоретический анализ

Целью настоящей статьи является анализ и исследование искусственных нейронных сетей для решения задачи прогнозирования временных рядов, в частности, задачи потребления гемокомпонентов лечебными профилактическими учреждениями, которая подробно представлена в [2].

Исходя из того, что задача прогнозирования является частным случаем задачи регрессии, следует, что она может быть решена следующими типами нейронных сетей: многослойный персептрон (МП), радиально-базисной сетью, обобщенно-регрессионной сетью, сетью Вольтерри, сетью Элмана и иерархической нейронной сетью [1, 3].

На первом этапе задача прогнозирования решалась с помощью сети многослойный персептрон.

Многослойными персептронами называют нейронные сети прямого распространения. Входной сигнал в таких сетях распространяется в прямом направлении, от слоя к слою. Многослойный персептрон в общем представлении состоит из следующих элементов:

- множества входных узлов, которые образуют входной слой;
- одного или нескольких скрытых слоев вычислительных нейронов;
- одного выходного слоя нейронов.

Далее для решения задачи прогнозирования была использована обобщенно-регрессионная нейронная сеть (НС) GRNN.

Нейронная сеть GRNN предназначена для решения задач обобщенной регрессии, аппроксимации функций и анализа временных рядов. Характерной особенностью данной сети является очень высокая скорость ее обучения.

И, наконец, для решения задачи прогнозирования была использована рекуррентная нейронная сеть Элмана.

Нейронная сеть Элмана — один из видов рекуррентной сети, которая получается из многослойного персептрона введением обратных связей, только связи идут не от выхода сети, а от выходов внутренних нейронов.

Это позволяет учесть предысторию наблюдаемых процессов и накопить информацию для выработки правильной стратегии управления.

На последнем этапе для решения задачи прогнозирования была использована иерархическая нейронная сеть [2]. Иерархическая нейронная сеть представляет собой совокупность двух и более совместно работающих нейронных сетей.

В данной работе была использована двухуровневая иерархическая нейронная сеть. Первый иерархический уровень предназначен для прогнозирования значений управляющего параметра для векторов входа второго иерархического уровня. Таким образом, сеть второго иерархического уровня осуществляет прогнозирование с учетом полученного на выходе сети первого уровня управляющего параметра. В общем случае, каждый иерархический уровень сети может иметь несколько скрытых слоев. Подробно методика построения иерархической нейронной сети представлена в [2].

Также было проведено дополнительное исследование задачи с целью проверки ряда на наличие – отсутствие тренда с помощью метода Фостера-Стьюарта [4].

Тренд – основная тенденция изменения временного ряда. Тренды могут быть описаны различными уравнениями – линейными, логарифмическими, степенными и т. д. Фактический тип тренда устанавливают на основе подбора его функциональной модели статистическими методами либо сглаживанием исходного временного ряда.

Для проверки гипотезы на наличие-отсутствие тренда в данном временном ряду был использован метод Фостера – Стьюарта.

Распространенным приемом при выявлении и анализе тенденции развития является сглаживание временного ряда. Суть различных приемов сглаживания сводится к замене фактических уровней временного ряда расчетными уровнями, которые в меньшей степени подвержены колебаниям. Это способствует более четкому проявлению тенденции развития.

Существует множество алгоритмов сглаживания временных рядов. В данной работе использовались алгоритмы сглаживания простых скользящих средних и взвешенных скользящих средних [3].

2. Экспериментальная часть

Для реализации прогнозирования временных рядов был использован пакет прикладных программ и одноименный язык программирования MATLAB. Он позволяет смоделировать и настроить нейронные сети для различных задач, в данном случае, для прогнозирования временных рядов.

Реализация сети многослойный персептрон началась с моделирования сети и определения числа нейронов в скрытых слоях. Оно определялось для каждого временного ряда индивидуально в соответствии с размерностью обучающего вектора. Результат работы сети многослойный персептрон представлен на рис.1, ошибка прогноза составила порядка 15.6%.

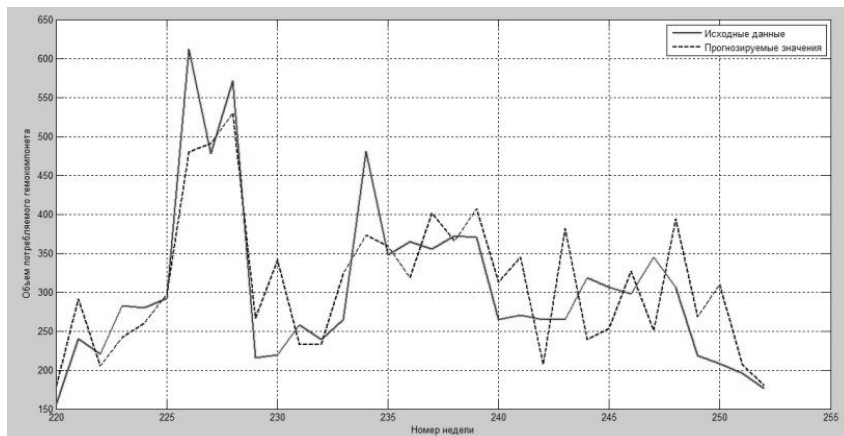


Рис. 1. Результат работы сети многослойный персептрон

Реализация обобщенно-регрессионной сети GRNN зависела от выбора функции Spread, которая отвечает за точность вычислений, то есть чем меньше значение функции spread, тем ближе значения аппроксимирующей кривой к заданным. Правда при этом получаемая кривая получается менее гладкой и если задать слишком маленькое значение spread, то получится просто график из соединенных между собой отрезков. Результат работы НС GRNN со значением Spread = 0.5 представлен на рис.2. Ошибка прогноза составила порядка 16.5%.

Реализация НС Элмана зависела от определения числа нейронов в скрытых слоях, также как и у сети МП. Так как она использует обратные связи, на ее обучение ушло большее время, чем на остальные. Результат работы НС Элмана представлена на рис. 3. Ошибка прогноза составила порядка 13.7%.

На последнем этапе прогнозирования временных рядов использовалась иерархическая нейронная сеть. Сложность реализации данной сети заключалась в подборе количества слоев для каждой нейронной сети. Результат работы иерархической сети представлен на рис.4. Ошибка прогноза составила порядка 9.5%.

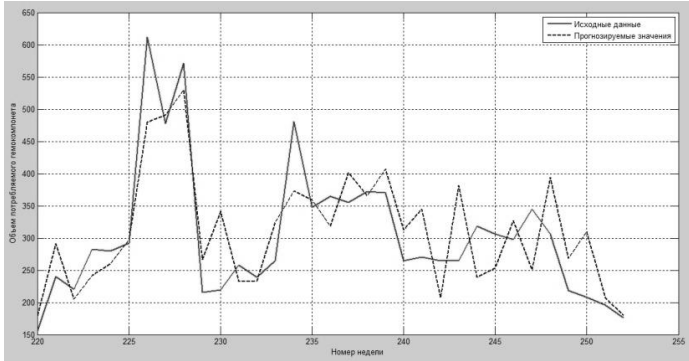


Рис. 2. Результат работы НС GRNN со значением spread = 0.5

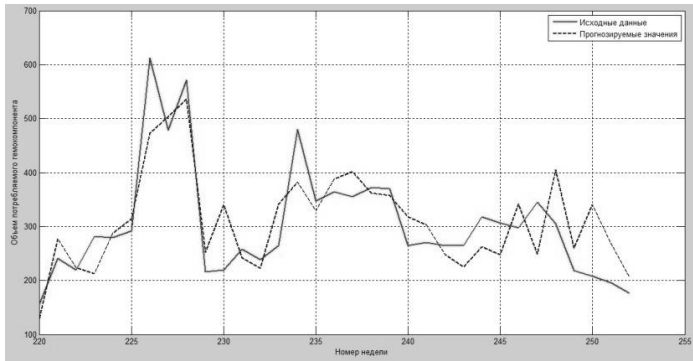


Рис. 3. Результат работы рекуррентной НС Элмана

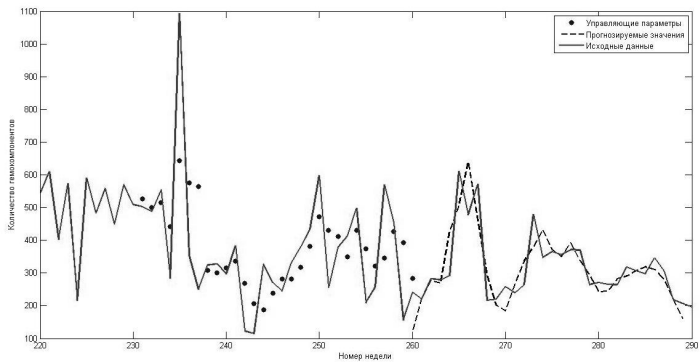


Рис. 4. Результат работы иерархической нейронной сети

Для наглядного сравнения качества прогнозирования различными искусственными нейронными сетями (ИНС), построим таблицу:

Таблица

Сравнение качества прогнозирования ИНС

| Тип нейронной сети | Ошибка прогноза |
|---------------------------|------------------------|
| Многослойный перцептрон | 15.6% |
| GRNN | 16.5%. |
| Реккурентная сеть Элмана | 13.7% |
| Иерархическая НС | 9.5% |

Далее в работе были использованы методы простых и взвешенных скользящих средних. Результаты их работы представлены на рис. 5 и рис. 6 соответственно.

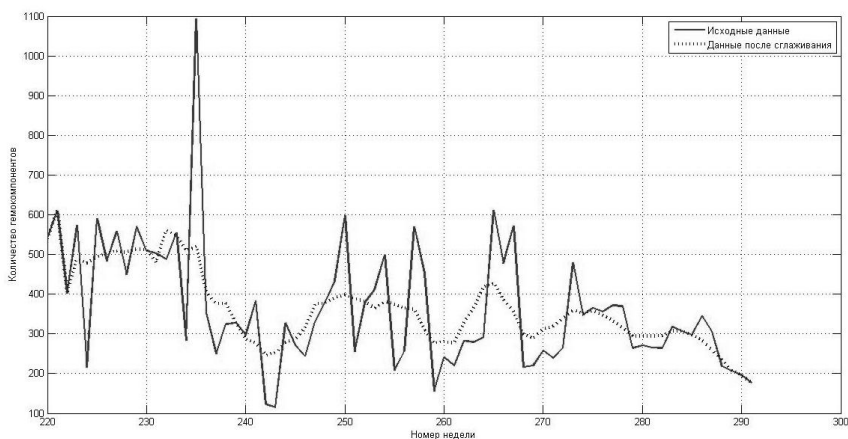


Рис. 5. Сглаживание методом простых скользящих средних

Заключение

Таким образом, в данной работе было реализовано прогнозирование временных рядов несколькими нейронными сетями. Было установлено, что для прогнозирования наиболее подходит иерархическая нейронная сеть, так как она имеет лучший показатель по точности прогнозирования, времени обучения и работы. Было реализовано сглаживание временных рядов с помощью методов простых и взвешенных скользящих средних и установлено, что в используемом ряде имеется тренд.

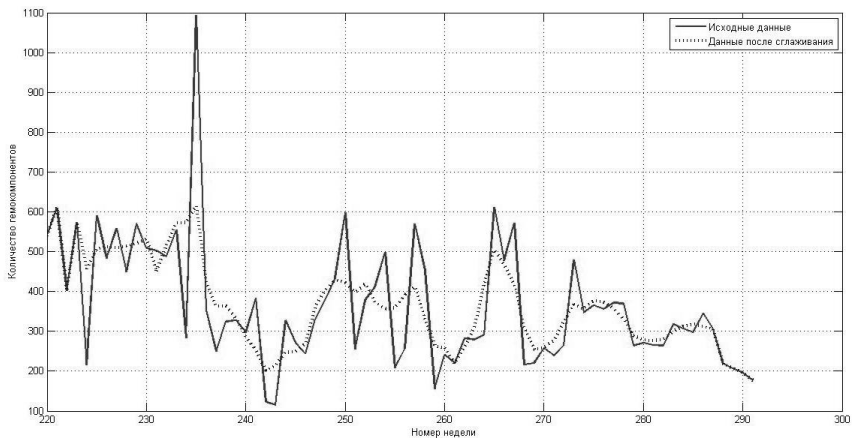


Рис. 6. Сглаживание методом взвешенных скользящих средних

Список литературы

1. Медведев, В. С. Нейронные сети. MATLAB 6 / В. С. Медведев; под ред. В. Г. Потемкина. – Москва : ДИАЛОГ-МИФИ, 2002. – 496 с.
2. Сирота, Е. А. Прогнозирование нестационарных временных последовательностей на основе иерархических многослойных нейронных сетей / Е. А. Сирота // Информационные технологии. – 2014. – №1. – С. 67-71.
3. Дуброва, Т. А. Статистические методы прогнозирования в экономике / Т. А. Дуброва, М. Ю. Архипова. – Москва : ДИАЛОГ-МИФИ, 2004. – 136 с.
4. Лавров, К. Н. Финансовая аналитика. MATLAB 6 / К. Н. Лавров, Т. П. Цыплякова; под ред. В. Г. Потемкина. – Москва : ДИАЛОГ-МИФИ, 2001. – 363 с.

КОМПЬЮТЕРНОЕ ТЕСТИРОВАНИЕ КАК МЕТОД ОЦЕНКИ КАЧЕСТВА ЖИЗНИ СТУДЕНТОВ

С. И. Трушков
Студент

А. О. Новикова
Студент

Е. А. Раскина
Студент

С. Д. Кургалин
Профессор

С. В. Борзунов,
Доцент

Л. П. Друганова,
Доцент

Т. Е. Фертикова
Доцент

Введение

Оценка состояния здоровья студентов – это актуальная задача, поскольку в жизни студенческой молодежи имеют место высокие психофизиологические и эмоциональные нагрузки, нарушения режимов труда, отдыха и питания, гиподинамия и т. п. Кроме того, студентам часто требуется мобилизация сил для адаптации к новым условиям проживания и обучения, формирования межличностных отношений и преодоления сложных жизненных ситуаций [1-2]. Важным при этом является изучение адаптации студентов к образовательному процессу на основе вырабатываемых показателей с применением информационных технологий [3-6].

Имеется достаточное количество научных работ по прогнозированию и регулированию качества жизни студентов, однако компьютерное тестирование для оценки качества жизни и программная обработка его результатов для получения максимально возможной информации с оценкой ее достоверности пока ещё используется слабо. Создается противоречие между объективной потребностью повышения качества

© Трушков С. И. (ВГМУ), Новикова А. О. (ВГМУ), Раскина Е. А. (ВГМУ), Кургалин С. Д. (ВГУ), Борзунов С. В. (ВГУ), Друганова Л. П. (ВГМУ), Фертикова Т. Е. (ВГМУ), 2015

жизни студентов в процессе обучения и недостаточностью знаний по оценке качества жизни самими студентами.

Целью настоящей работы является создание программного средства для оценки качества жизни студентов, проведение исследований на его основе и анализ полученных результатов.

1. Цель и задачи исследования

Для оценки качества жизни студентов было создано специальное программное приложение. Один из интерфейсов программы представлен на рис. 1.

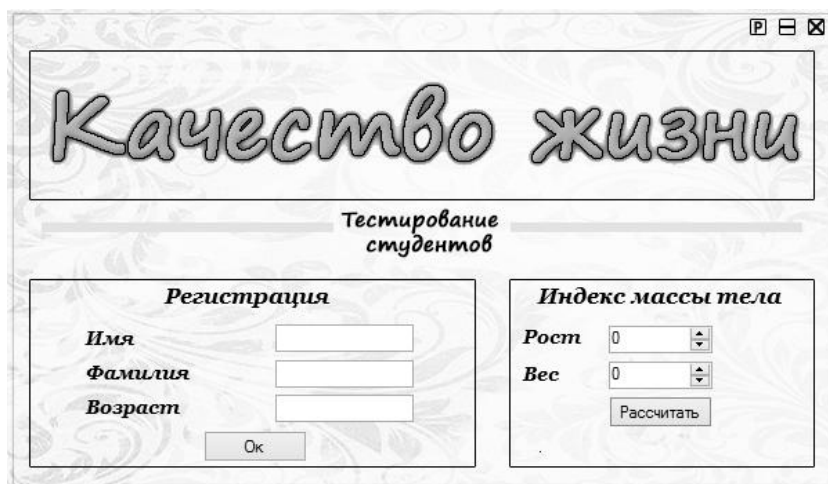


Рис. 1. Вид интерфейса программного приложения

Целью практической части исследования явилось получение новых знаний о качестве жизни студентов Воронежского государственного университета (ВГУ) и Воронежского государственного медицинского университета им. Н.Н.Бурденко (ВГМУ).

Задачи исследования: сравнить качество жизни студентов ВГУ и ВГМУ по различным параметрам; сформулировать рекомендации по необходимой корректировке образа жизни студентов.

2. Организация и методы исследования

В исследовании приняли участие 209 человек: студенты 2-го и 3-го курсов факультета компьютерных наук ВГУ и студенты 2-го и 5-го курсов лечебного факультета ВГМУ, из них 53 – юноши и 156 – девушки.

Методы исследования: анкетирование, компьютерное тестирование, статистический анализ. До исследования был проведен анализ существующей научно-методической литературы по данной тематике.

Для оценки качества жизни студентов был составлен опросник, позволяющий оценить интегральную характеристику физического, психологического, эмоционального и социального состояния студентов. Вопросы сгруппированы в пять блоков, по которым оцениваются: соблюдение режима питания, режимов сна и бодрствования, подверженность сезонным заболеваниям, стрессоустойчивость и материальное положение. Опросник является «закрытым», в нем заранее предусмотрены варианты ответов на поставленный вопрос и испытуемый должен выбрать один из вариантов.

Программное приложение позволяет достаточно просто оценить качество жизни на основе предложенных показателей и статистической обработки результатов. Максимальная сумма всех имеющихся показателей равна 119. Чем больше набранная сумма показателей при опросе, тем, соответственно, более высоким является качество жизни определенного студента.

3. Результаты исследования

Результаты проведенных исследований показали, что только 18% опрошенных второкурсников и 62% пятикурсников ВГМУ соблюдают правильный режим питания, в то время как 40% опрошенных второкурсников и 64% третьекурсников ФКН ВГУ придерживаются рационального режима питания.

40% студентов младших курсов и 20% старшекурсников ВГМУ тратят на ежедневные занятия 12 и более часов, в то же время 48% старшекурсников ежедневно занимаются 6 и менее часов. Что касается студентов ФКН ВГУ, то 50% студентов тратят на ежедневные занятия около 10 часов, а 57% старшекурсников занимаются 6 и менее часов.

Студенты-медики в меньшей степени соблюдают режим сна и бодрствования по сравнению со студентами ФКН ВГУ.

Старшекурсники обоих университетов оказались достаточно адаптированными к ежедневным стрессам и поэтому мало восприимчивы к ним. Студенты младших курсов только начинают адаптироваться к стрессовым факторам и являются более им подверженными.

Подавляющее большинство студентов лечебного факультета ВГМУ жалуются на сильное чувство усталости после учебы, а некоторые – на чувство усталости ещё до начала учебного дня. Студенты же факультета компьютерных наук отмечают усталость только после окончания учебы.

По результатам опроса выявлено, что девушки в целом оценивают свой уровень качества жизни ниже, чем юноши. Девушки более

подвержены ежедневным и еженедельным стрессам. В среднем, они уделяют 6 и менее часов на сон, мало занимаются в спортивных секциях, продолжительно и часто болеют простудными заболеваниями.

Студенты-старшекурсники оценивают качество своей жизни как более высокое по сравнению со студентами младших курсов по таким параметрам, как режим питания, труда и отдыха, время, затрачиваемое на ежедневную подготовку к занятиям, и устойчивость к стрессовым ситуациям.

Заключение

Созданное программное средство является полезным для выяснения важных характеристик жизни и учебы студентов вузов. Простота процедуры регистрации в программе и обработки данных позволяют легко проводить массовые обследования студентов.

На основе проведенных исследований можно сделать вывод, что качество жизни студентов не является достаточно высоким. Для его улучшения необходимо:

- формирование комплексных оздоровительных программ в вузах и реальное их осуществление;
- составление расписания занятий в соответствии с динамикой работоспособности студентов в течение учебного дня и недели, недопущение «перегрузок»;
- равномерное распределение учебной нагрузки в течение семестра;
- установление требований соблюдения студентами необходимых гигиенических норм и правил.

Список литературы

1. Концептуальные основы охраны здоровья и повышения качества жизни учащейся молодежи региона / И. Э. Есауленко [и др.]. – Воронеж : ИПЦ Научная книга, 2013. – 797 с.
2. Фертикова, Т. Е. Проблема здоровья и болезни студенческой молодежи / Т. Е. Фертикова, А. А. Рогачев // Научно-медицинский вестник ЦЧ : научно-практический журнал. – Воронеж, 2015. – № 60 (II квартал). – С. 35-38.
3. К проблеме изучения адаптивных резервов студентов с применением информационных технологий / С. Д. Кургалин, А. М. Попов, Л. В. Болдырева [и др.] // Бюллетень Северного государственного медицинского университета. – 2013. – № 1 (Выпуск XXX). – С. 134-135.
4. Оценка адаптационного потенциала студентов-медиков с использованием компьютерных программ / С. Д. Кургалин, Л. П. Друганова, Т. Е. Фертикова, Т. В. Шаева // Медико-биологические и

педагогические основы адаптации, спортивной деятельности и здорового образа жизни : сборник науч. статей II Всерос. заочной науч.-практической конф. с международным участием, Воронеж, 2013 г. / под ред. Г.В. Бугаева, И.Е. Поповой. – Воронеж : Издательско-полиграфический центр «Научная книга», 2013. – Т. 1. – С. 50-53.

5. Исследование адаптационных возможностей организма студентов / С. Д. Кургалин, А. Г. Панов, А. М. Попов [и др.] // Сборник студенческих научных работ факультета компьютерных наук ВГУ / под ред. Е. А. Сирота. – Воронеж : Издательско-полиграфический центр Воронеж. гос. ун-та, 2012. – Вып. 6. – С. 130-133.

6. Друганова, Л. П. Изучение адаптации студентов-медиков к образовательному процессу на основе показателей системы кровообращения с применением информационных технологий / Л. П. Друганова, Т. Е. Фертикова, С. Д. Кургалин // Личность, семья и общество : вопросы педагогики и психологии : сборник статей по материалам XXXIX Международ. науч.-практической конф. – Новосибирск : НП «СибАК», 2014 г. – № 4(39). – Ч. 1. – С. 8-13.

РАЗРАБОТКА ИНТЕРНЕТ-САЙТА «ЗДОРОВЫЙ ОБРАЗ ЖИЗНИ СТУДЕНТОВ»

С. И. Трушков
Студент

С. Э. Черенкова
Студент

К. А. Матвеева
Студент

А. Г. Правильников
Студент

С. Д. Кургалин,
Профессор

Ю. Г. Притулина
Профессор

© Трушков С. И. (ВГМУ), Черенкова С. Э. (ВГМУ), Матвеева К. А. (ВГМУ), Правильников А. Г. (ВГМУ), Кургалин С. Д. (ВГУ), Притулина Ю. Г. (ВГМУ), Борзунов С. В. (ВГУ), Фертикова Т. Е. (ВГМУ), Друганова Л. П. (ВГМУ), Целиковский А. В. (ВГМУ), 2015

С. В. Борзунов,
Доцент
Т. Е. Фертикова,
Доцент
Л. П. Друганова,
Доцент
А. В. Целиковский
Ассистент

Введение

Актуальность создания сайта о здоровом образе жизни не вызывает сомнений, в особенности, если он посвящается здоровому образу жизни студентов. Проводимые исследования показывают что, к сожалению, жизнь современных студентов во многих случаях характеризуется отсутствием режима дня, неправильным питанием, ухудшением сна, игнорированием занятий физкультурой и спортом, а также, иногда, и наличием вредных привычек. Поэтому важно донести до молодежи, используя возможности Интернет, путем создания Интернет-сайтов (см., например, [1]), основные принципы и методы ведения здорового, правильного образа жизни.

Целью настоящей работы является разработка Интернет-сайта, позволяющего оказать помощь в формировании необходимых представлений о здоровом образе жизни студентов, научить их планировать режим дня, донести информацию о принципах эффективной организации умственного труда, о допустимых физических нагрузках, способах борьбы с вредными привычками и методах их профилактики.

1. Задачи и особенности Интернет-сайта, его информационное наполнение

Основными задачами создаваемого сайта являются:

- привлечь внимание студентов к главным аспектам здорового образа жизни;
- сообщить о важности соблюдения режима дня, научить правильно строить его для выработки динамического стереотипа поведения;
- показать принципы рационального питания, донести правила построения рациональных физических и умственных нагрузок;
- рассказать о способах борьбы с вредными привычками и методах их предотвращения.

С целью реализации поставленных задач группой студентов факультета компьютерных наук Воронежского государственного университета и Воронежского государственного медицинского университета им. Н.Н. Бурденко под руководством преподавателей создается Интернет-сайт «Здоровый образ жизни студентов».

Вид главной страницы сайта представлен на рис. 1.



Рис. 1. Главная страница сайта

Особенности создаваемого сайта:

- сайт строится на базе информационных технологий в виде полноценного веб-проекта;
- в процессе создания сайта непрерывно анализируется его эффективность относительно привлечения внимания пользователей;
- обеспечивается возможность модернизации и развития сайта;
- организована обратная связь администратора сайта с его пользователями.

На сайте предусмотрена функция регистрации и авторизации, которая предоставляет администратору возможность контролировать его посещаемость. Авторизованные пользователи могут оставлять комментарии к статьям, а также вносить свои предложения по улучшению сайта.

Для создания сайта был использован бесплатный конструктор uCoz, имеющий в своем составе большое количество стандартизированных и специальных методов разработки сайтов. Выбор этого конструктора обусловлен также и тем, что он рассчитан на администрирование сайта неопытным пользователем, а его характерные особенности – простота и доступность.

Созданный сайт содержит множество практических советов и рекомендаций по различным вопросам. Наиболее важные из них, составляющие основу информационного наполнения сайта, кратко представлены ниже.

Первый момент – важность соблюдения режима дня. Организация рационального режима дня должна проводиться с учетом особенностей работы конкретного вуза (расписания занятий), оптимального использования имеющихся условий, учета индивидуальных качеств студентов. В распорядок дня, помимо, конечно, занятий, необходимо включить утреннюю гимнастику, прогулки на свежем воздухе, 3-4-разовый прием пищи. Рекомендуется заниматься каким-нибудь видом спорта. Время сна должно составлять не менее 7 часов. Лучшее время для умственных занятий – утро (до полудня) и вечер (с 16.00 до 18.00). Через каждый час необходимо делать перерыв. Желательно ежедневно по вечерам уделять 3-5 минут времени для планирования следующего дня. Правильно спланированный режим дня является ключом к достижению успехов и сохранению хорошего здоровья.

Вторым важным моментом является сон. Во время сна восстанавливаются системы, которые подвергались перегрузкам (переутомлению или болезненным изменениям), происходит накопление богатых энергией фосфорных соединений, повышаются защитные силы организма. Хронические недосыпания способствуют так называемой «сшибке» – появлению неврозов, ухудшению функциональных состояний и снижению защитных сил организма.

Третьим важным моментом являются физические упражнения. Не обязательным является непрерывное посещение тренажерного зала или спортивной секции, но необходимо уделить внимание гимнастическим упражнениям, утренней пробежке, прогулкам быстрым шагом.

Четвертый момент – правильное питание. Оно должно включать основные источники белка – молочные продукты, яйца, рыбу, мясо. В пище также должны содержаться жиры, не менее одной трети из которых – растительные. Блюда из рыбы позволяют улучшить деятельность головного мозга и станут хорошей профилактикой атеросклероза. Большое значение имеют углеводы, которые выступают в роли «топлива» клеток мозга. Картофель, хлеб, сахар, каши, кондитерские изделия, шоколад – основные источники углеводов, однако избыток данных продуктов может стать причиной появления избыточного веса. Главными источниками витаминов являются фрукты и овощи, зелень, поэтому они должны обязательно присутствовать в рационе, желательно употреблять их не менее 200-300 г в день. Такая пища содержит минимальное количество

калорий, что дает возможность насытить организм питательными веществами и при этом не увеличить вес.

Пятым пунктом является иммунопрофилактика. Речь идет о предупреждении инфекционных заболеваний студентов. Стоит подчеркнуть, что в молодом возрасте (в этой возрастной группе онкологические заболевания и болезни сердечно-сосудистой системы являются достаточно редкими) инфекционные заболевания выходят на первое место среди других по частоте встречаемости. Способом предупреждения инфекционных болезней является иммунопрофилактика, основным компонентом которой является вакцинопрофилактика, о ней имеется информация на сайте.

Крайне важным при формировании здорового образа жизни является также оценка и учет адаптационных возможностей организма студентов [2-5].

Конечно, сайт рекомендует исключить из жизни студента курение и алкоголь. Продуктом окисления алкоголя является ацетальдегид. Он является основным повреждающим фактором, вызывает гибель клеток печени, на месте которых образуется соединительная и жировая ткань, а, следовательно, нарушается ее функционирование. Этанол, поступив в ткань головного мозга, вызывает гибель части его клеток. На их месте образуются рубцы, а при обширных поражениях – кисты. При курении из табачного дыма высвобождается более 200 вредных веществ. Токсичные вещества, оседая на всём протяжении дыхательной системы, вызывают ее воспаление, поэтому курильщики болеют ларингитом, трахеитом, бронхитом, кроме того, вещества табачного дыма оказывают и канцерогенное воздействие.

Повышение эффективности работы сайта в дальнейшем может быть достигнуто следующими способами:

- регулярным обновлением информации и добавлением новых материалов;
- введением на сайт интерактивных элементов, которые могли бы упростить связь между администраторами сайта и его пользователями;
- проведением рекламной кампании с целью привлечения внимания широких масс студентов и преподавателей вузов к данному веб-сайту.

Заключение

Разработанный интернет-сайт, предназначенный, в основном, для посетителей студенческого возраста, помогает сформировать правильное представление о здоровом образе жизни. Он является эффективным средством, позволяющим знакомить студентов с основными принципами и методами, которые обеспечат сохранение и укрепление здоровья.

Входящая в состав сайта система тестирования дает возможность выявить проблемы со здоровьем.

Список литературы

1. Интернет-сайт стоматологической поликлиники с акцентом на профилактику заболеваний / Д. И. Саранцев, А. А. Смолина, Е. В. Калмыкова [и др.]; под ред. Е. А. Сирота // Сборник студенческих научных работ факультета компьютерных наук ВГУ. – Воронеж : Издательско-полиграфический центр Воронеж. гос. ун-та, 2013. – С. 114-118.

2. К проблеме изучения адаптивных резервов студентов с применением информационных технологий / С. Д. Кургалин, А. М. Попов, Л. В. Болдырева [и др.] // Бюллетень Северного государственного медицинского университета. – 2013. – № 1 (Выпуск XXX). – С. 134-135.

3. Друганова, Л. П. Изучение адаптации студентов-медиков к образовательному процессу на основе показателей системы кровообращения с применением информационных технологий / Л. П. Друганова, Т. Е. Фертикова, С. Д. Кургалин // Личность, семья и общество : вопросы педагогики и психологии : сборник статей по материалам XXXIX Международ. науч.-практической конф. – Новосибирск : НП «СибАК», 2014 г. – № 4(39). – Ч. 1. – С. 8-13.

Оценка адаптационного потенциала студентов-медиков с использованием компьютерных программ / С. Д. Кургалин, Л. П. Друганова, Т. Е. Фертикова, Т. В. Шаева // Медико-биологические и педагогические основы адаптации, спортивной деятельности и здорового образа жизни : сборник науч. статей II Всерос. заочной науч.-практической конф. с международным участием, Воронеж, 2013 г. / под ред. Г. В. Бугаева, И. Е. Поповой. – Воронеж : Издательско-полиграфический центр «Научная книга», 2013. – Т. 1. – С. 50-53.

4. Исследование адаптационных возможностей организма студентов / С. Д. Кургалин, А. Г. Панов, А. М. Попов [и др.]; под ред. Е. А. Сирота // Сборник студенческих научных работ факультета компьютерных наук ВГУ. – Воронеж : Издательско-полиграфический центр Воронеж. гос. ун-та, 2012. – Вып. 6. – С. 130-133.

ИНФОРМАЦИОННАЯ СИСТЕМА ТАКТИЛЬНОЙ И ТЕМПЕРАТУРНОЙ ОБРАТНОЙ СВЯЗИ В МИОЭЛЕКТРИЧЕСКИХ ПРОТЕЗАХ

К. И. Фисенко

Студент

С. А. Пешков

Студент

Я. А. Туровский

Доцент

Введение

Реабилитация инвалидов – яркий пример одной из самых острых проблем здравоохранения в Российской Федерации. В частности, обеспечение необходимой помощи нуждающимся, оснащение их техническими средствами реабилитации и протезно-ортопедическими изделиями являются важными элементами деятельности системы лечебно-профилактических учреждений. Примерно 10% из 12 миллионов инвалидов в России нуждаются в протезировании конечностей [1]. На 2011 год потребность в них была удовлетворена лишь на 70-80%. А объем рынка на этот же год составил порядка 59 тыс. штук искусственных конечностей [2]. При этом ассортимент современных протезов в нашей стране на 90% состоял из импортных изделий, в то время как, доля отечественных производителей была весьма малой.

1. Система обратной связи в миоэлектрических протезах

За последние несколько лет достаточно обширное распространение получили бюджетные протезы рук, созданные с помощью 3D печати. В нашей стране появилось несколько команд, пытающихся решить острую нехватку протезов именно этим способом. Одним из главных недостатков практически всех проектов независимо от технологии изготовления является отсутствие системы обратной связи между протезом и пользователем. Следствием этой проблемы является неточная работа большинства протезов. Одним из примеров может служить избыточное или наоборот недостаточное давление механизмов искусственной кисти на предметы, с которыми возможно её взаимодействие. В настоящее время данная проблема решается так, что в электромеханические системы протезов вводятся уже готовые программно-аппаратные решения,

изменяющие силу сжатия кисти независимо от команд, подаваемых пользователем, и связанные с параметрами удерживаемого объекта.

2. Цель работы

Целью работы является создание доступного и простого в производстве комплекса тактильной и температурной обратной связи миоэлектрического протеза.

Эта система разрабатывается как дополнительный модуль, расширяющий его функциональные возможности, автономный или частично интегрируемый в искусственную конечность.

Для этого потребуется выполнение ряда задач:

- разработка подходов по установке и расположению тактильных и температурных датчиков в зависимости от анатомических особенностей руки и возраста пользователя;
- локализация поступающих сигналов с датчиков на кисти протеза и поиск оптимального количества датчиков;
- создание тактильного стимулятора передающего результат обработки сигнала на другие части тела;
- создание системы обучения человека работе с подобными миоэлектрическими протезами.

3. Макет кисти человека

В качестве прототипа для решения первой проблемы был изготовлен образец протеза кисти человека (рис. 1).

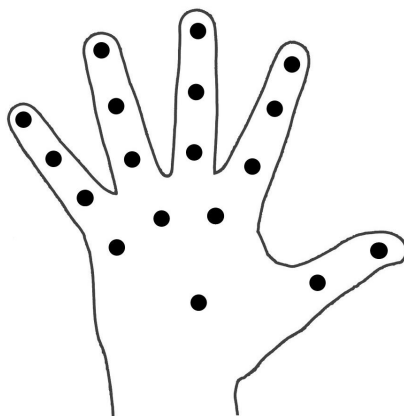


Рис. 1. Макет кисти человека

Создание этого макета было выполнено путем печати 3D модели из открытого аппаратного проекта <http://www.inmoov.fr/>. Одним из достоинств этого метода является возможность простого масштабирования и изменения деталей, что позволяет подобрать подходящую по размерам модель для любого инвалида. Так же это гарантирует максимальное приближение к продукции многих команд, разрабатывающих бюджетные протезы с помощью подобной технологии. Изготовление модели кисти руки позволило проведение дальнейших исследований по расположению датчиков разной направленности, а так же помогло в решении следующей задачи нашей работы.

4. Расположение датчиков, обработка сигнала, стимуляторы

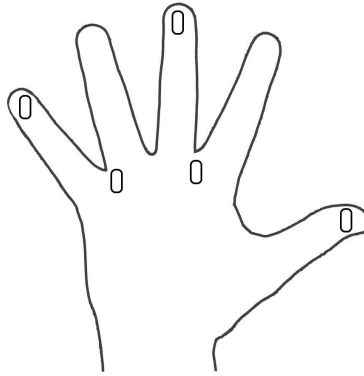
Вторая задача заключается в изучении локализации поступающих сигналов и в поиске минимально возможного количества датчиков для нормальной работы системы без искажений. В ходе работы опытным путем было определено оптимальное количество тактильных датчиков для кисти человека в количестве 18 штук (рис. 2).



● Тактильные датчики

Рис. 2. Схема расположения тактильных датчиков.

Число температурных цифровых датчиков DS18B20 было ограничено в размере пяти (рис. 3). Диапазон измерений от -55°C до $+125^{\circ}\text{C}$ и точность 0.5°C в диапазоне от -10°C до $+85^{\circ}\text{C}$. Другая особенность DS18B20 – способность работать без внешнего питания. При этом максимальная измеряемая температура $+100^{\circ}\text{C}$. Для расширения диапазона температура до $+125^{\circ}\text{C}$ необходимо использовать внешнее питание [3].



□ Температурные датчики

Рис. 3. Схема расположения температурных датчиков

В нашем комплексе сигнал с тактильных сенсоров идет через LPT разъем к USB модулю KernelChip Ke-USB24A. Затем он обрабатывается с помощью одноплатного микрокомпьютера Raspberry Pi. Программно-аппаратный комплекс системы позволяет локализовать источник сигнала, определить одновременную работу нескольких датчиков, визуализировать связь между областями кисти руки для отображения, взятого рукой объекта. Полученные данные передаются тактильному стимулятору, преобразующему их по алгоритмам на основе нейронных сетей нечеткой логики. Датчики DS18B20 посылают сигнал сразу на Raspberry Pi. Откуда уже обработанные данные отправляются на температурный стимулятор. В данный момент ведется разработка тактильного и температурного стимуляторов.

И на последнем этапе изготовления лабораторной установки системы потребуются разработка сферы обучения человека использованию этого комплекса обратной связи для протезов различных типов. Эта программа поможет инвалидам в определении температуры, твердости и форме предметов. В настоящее время разрабатывается прототип.

5. Аналогичные разработки за границей

Изучение данной системы в области протезирования в последнее время набирает обороты по популярности. Финансированием исследований подобного плана в США занимается агентство перспективных оборонных исследований (DAPRA).

Созданием миоэлектрических протезов с обратной связью занимаются:

- лаборатория гибких биоэлектронных интерфейсов Федеральной политехнической школы Лозанны (Швейцария);
- лаборатория передовых разработок робототехники и систем, Высшей нормальной школы, Пиза (Италия);
- университет Тель-Авива (Израиль).

К сожалению, стоимость таких разработок очень высока, и готовая продукция оценивается десятками тысяч долларов.

Заключение

Данная статья посвящена разработке системы обратной тактильной и температурной связи в миоэлектрическом протезе. Разработана и реализована схема размещения сенсоров на механической части миоэлектрического протеза руки человека.

Список литературы

1. Федеральная служба государственной статистики [Электронный ресурс] : база данных. – Режим доступа : http://www.gks.ru/wps/wcm/connect/rosstat_main/rosstat/ru/statistics/population/disabilities/#
2. Abercade – исследование промышленных рынков [Электронный ресурс] : Отраслевые новости. – Режим доступа : <http://abercade.ru/research/industrynews/9165.html>
3. MaximIntegrated. Sensors and Sensor interface [Электронный ресурс]. – Режим доступа : <http://www.maximintegrated.com/en/products/analog/sensors-and-sensor-interface/DS18B20.html>

ПРЕДВАРИТЕЛЬНАЯ ПОДГОТОВКА РЕЧЕВЫХ СИГНАЛОВ ДЛЯ РАСПОЗНАВАНИЯ

А. С. Четкин

Студент

С. А. Запрягаев

Профессор

Введение

Подготовка и преобразование звуковых данных для распознавания включает в себя задачу представления звукового сигнала в максимально компактной форме без потери значимой информации. В настоящее время существует ряд способов представления исходных данных в компактной форме, достаточных для хорошей идентификации звуковых сигналов. В

данной работе реализованы методы подготовки и обработки исходного сигнала для последующего его распознавания с использованием специализированной программной оболочки. Задачей программной оболочки является реализация адекватного интерфейса для анализа полезной информации в звуковом сигнале. В рамках данного программного средства реализована следующая схема подготовки сигнала:

- запись сигнала;
- нормирование;
- очистка сигнала от шума;
- сегментация речи;
- сокращение объема данных.

Данная последовательность обработки данных выделяет «полезную» часть звукового сигнала, в которой сосредоточено максимум необходимой информации. В качестве полезной информации рассматривается звуковой сигнал с минимальными шумовыми составляющими и без «тихий» областей. Для нормировки сигнала используется линейное нормирование от -1 до 1 по формуле:

$$\tilde{X}_k = 2 \frac{X_k - \min(X)}{\max(X) - \min(X)} - 1, \quad (1)$$

где $\max(X)$ – максимум сигнала X , $\min(X)$ – минимум сигнала X .

1. Очистка сигнала от шума

Если в качестве исходных данных используются файлы с высоким уровнем шума, то необходимо произвести фильтрацию сигнала, т. к. высокие значения шумовых составляющих могут сильно повлиять на качество распознавания. Кроме того, фильтрация позволяет получить точные результаты сегментации речи. В качестве фильтра для уменьшения зашумленности сигнала был выбран фильтр на основе вейвлет преобразования. В качестве вейвлетов использовались вейвлеты Добеши от 1 до 4 порядка [1]. Используемый алгоритм основан на применении технологии мягкого порога (soft thresholding) для детализирующих коэффициентов преобразования. Схема алгоритма шумоподавления на основе вейвлет преобразования включает в себя:

- прямое дискретное вейвлет преобразование звукового сигнала;
- определение порога;
- применение soft thresholding для детализирующих коэффициентов;
- обратное дискретное вейвлет преобразование.

Для определения порога шума используется универсальный метод, предложенный Донохо и Джонстоном [2]. В рамках данного метода на первом этапе вычисляется уровень шума по формуле:

$$S = \text{mediana}(X) \cdot wc, \quad (2)$$

где $\text{mediana}(X)$ – значение медианы на всей длине сигнала, wc – весовой коэффициент.

Соответственно, на втором этапе определяется значение порога Δ по формуле:

$$\Delta = S\sqrt{2\ln N}, \quad (3)$$

где Δ – значение порога, N – длина сигнала, S – уровень шума, вычисленный на предыдущем шаге.

В качестве весового коэффициента использовалось значение $wc=1.48$, которое является отношением между среднеквадратическим и медианным абсолютным отклонением. Полученное значение порога используется для soft thresholding на детализирующих коэффициентах дискретного вейвлет преобразования. Преобразование детализирующих коэффициентов осуществляется по формуле:

$$\tilde{C}_k = \text{sign}(C_k) \max(0, |C_k| - \Delta), \quad (4)$$

где C_k – k -ый коэффициент, $\text{sign}(\cdot)$ – функция определения знака, Δ – значение порога.

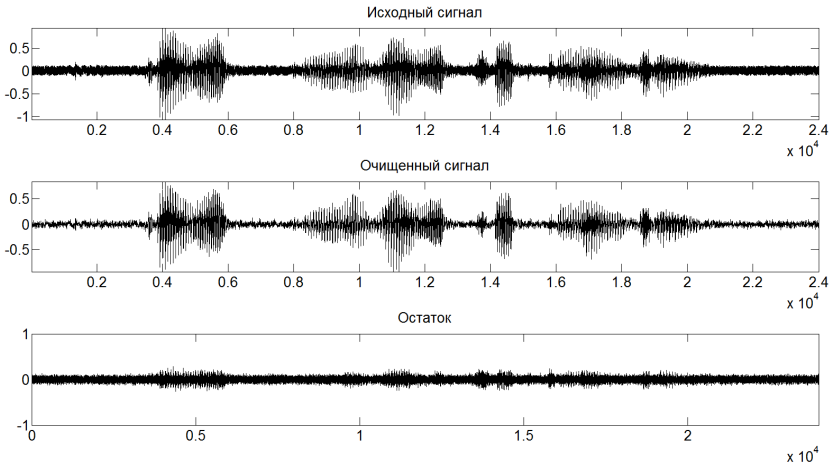


Рис. 1. Очистка от шума фразы “Привет, меня зовут Четкин Андрей”

На рис. 1 представлен реальный пример очистки от шума речевого сигнала по указанной выше методике. Верхний сигнал – исходный. Ниже представлен вид очищенного сигнала при сохранении масштаба. Нижний сигнал – разница между исходным и очищенными сигналами.

2. Сегментация речи

На втором этапе подготовки входных данных выделяются только те участки сигнала, в котором содержится полезная информация. При записи речи возникают промежутки, в которых диктор не говорит. При распознавании необходимо удалить такие участки и выделить из сигнала полезные промежутки, в которых представлен значащий речевой сигнал. Для выделения полезных участков используется алгоритм сегментации речи. Данный алгоритм состоит из 2 этапов: а) поиск начала слова, б) поиск длины слова. На каждом этапе сравниваются значение мощности звука со значением порога на участке в 20 миллисекунд. Средняя мощность сигнала вычисляется по формуле [3]:

$$P = \frac{1}{N} \sum_{k=0}^{N-1} x_k^2. \quad (5)$$

Если значение порога больше значения мощности $\Delta > P$, то данный участок является тишиной. Сравнение происходит до тех пор, пока значение мощности не станет больше значения порога. Затем происходит вычисление длины полезного сигнала аналогично, только при условии, что значение мощности сигнала больше порога $P > \Delta$. Переход от одного участка к другому производится с перекрытием, т. е. переход к следующему участку производится каждые 10 мс сигнала. Кроме того, в конце алгоритма происходит проверка длины слова. Если длина слова оказалась меньше 100 мс, то данный участок рассматривается как тишина. Данная проверка проводится для того, чтобы не выделить случайно возникшие всплески. Значение порога вычисляется путем поиска максимума мощности в записи «пустого» сигнала.

3. Сжатие звука

После очистки сигнала и сегментации осуществляется преобразование сигнала к компактной форме. Звуковая информация является избыточной, поэтому данное свойство звука может влиять на результат и скорость распознавания. Для уменьшения избыточности в данной работе использовалось сжатие данных, основанное на вейвлет преобразовании и частотном представлении сигнала. Большое количество данных влечет за собой не только более длительное время распознавания, но и увеличивающуюся со временем ошибку, которая может негативно влиять на результат распознавания. Поэтому необходимо представить данные в максимально «сжатой форме» с минимальной потерей информации в сигнале.

Для сжатия звуковой информации был использован метод на основе вейвлет преобразовании. Данный метод удаляет высокочастотные составляющие из коэффициентов вейвлет преобразования. Для получения

спектра вейвлет коэффициентов использовалось дискретное косинус преобразование.

Сжатие происходит за счет обнуления коэффициентов косинус преобразования. Для изучения влияния сжатия исходных сигналов использовались записи слов длиной в одну секунду с частотой дискретизации в 8000 Гц. То есть каждый сигнал содержал 8000 отсчетов. В рамках эксперимента из 12 слов (представленных в табл. 1 и табл. 2) было выявлено, что обнуление в среднем 1000 значений в начале детализирующих коэффициентов и 1000 значений в конце аппроксимирующих дает результат сокращения данных на 25% при использовании вейвлета Добеши 4 порядка. Ниже (для сравнения) приведена табл. 2 сжатий различных слов при использовании различных вэйвлетов Добеши (db4 и db1) при которых искажения исходного сигнала практически не проявляется. В качестве порога для обнуления использовалась 1/10 от максимального значения в спектрах вейвлет коэффициентов.

Таблица 1

Результаты сжатия для 6 команд

| Слово \ Сжатие | Вверх | Вниз | Влево | Вправо | Старт | Стоп |
|----------------|-------|------|-------|--------|-------|------|
| db4 | 24% | 25% | 20% | 19% | 19% | 30% |
| db1 | 10% | 6% | 6% | 7% | 19% | 25% |

Таблица 2

Результаты сжатия для 6 цифр

| Слово \ Сжатие | Один | Два | Три | Четыре | Пять | Шесть |
|----------------|------|-----|-----|--------|------|-------|
| db4 | 20% | 30% | 19% | 18% | 20% | 18% |
| db1 | 14% | 28% | 12% | 7% | 19% | 10% |

4. Построение векторов признаков

Для исследования звуковых сигналов и поиска отличительных признаков был использован набор из 18 слов-команд: «вверх», «вниз», «влево», «вправо», «старт», «стоп», «вперед», «назад», «разворот», «один», «два», «три», «четыре», «пять», «шесть», «семь», «восемь», «девять». Каждое из этих слов было записано в файл длиной в 1 секунду с частотой дискретизации 8000 Гц и амплитудным квантованием в 16 бит. Для построения векторов признаков было использовано мел-кепстральное преобразование с 80 треугольными окнами.

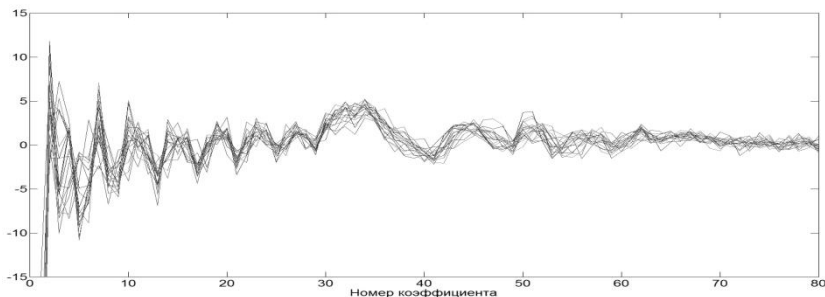


Рис. 2. Мел-кепстральные коэффициенты 18 слов, сказанных молодым человеком 23 лет

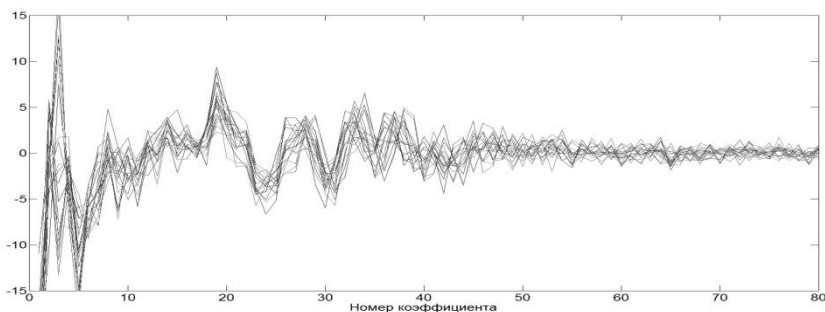


Рис. 3. Мел-кепстральные коэффициенты 18 слов, сказанных девушкой 22 года

Заключение

Результаты анализа мел-кепстральных коэффициентов показали, что данное преобразование может быть успешно использовано для идентификации дикторов. Для исследования особенностей мел-кепстрального преобразования 4 человека: 2 молодых человека и 2 девушки сказали 18 слов. В результате полученных данных было выявлено, что в выходных векторах присутствует общий тренд у всех слов каждого диктора. Данный тренд может свидетельствовать о том, что все слова имеют общий признак, а именно, общий для всех слов голос человека.

Список литературы

1. Добеши, И. Десять лекций по вейвлетам / И. Добеши ; пер. с англ., под ред. А. П. Петухова. – Ижевск : НИЦ «Регулярная и хаотическая динамика», 2001. – 464 с.

2. Donoho, D. L. Ideal spatial adaptation by wavelet shrinkage / D. L. Donoho, I. M. Johnstone // *Biometrika*. – 1994. – No. 81. – P. 425-455.

3. Макс, Ж. Методы и техника обработки сигналов при физических измерениях : в 2-х т., Т. 1 / Ж. Макс; пер. с франц. А. Ф. Горюнова и А. В. Крянева; под ред. Н. Г. Волкова. – Москва : Мир, 1983. – 312 с.

О МАТРИЧНЫХ ЭЛЕМЕНТАХ ОДНОГО СЕМЕЙСТВА МАТРИЧНЫХ АЛГЕБР ЛИ

А. Г. Чурсин

Студент

Е. А. Павлова

Студент

А. В. Лобода

Профессор

Постановка задачи

В работе рассматривается вопрос о существовании 6-мерных и 5-мерных вещественных матричных алгебр Ли, имеющих некоторые ограничения на элементы, содержащихся в их матрицах. Такой вопрос связан с задачей описания свойства аффинной однородности для вещественных гиперповерхностей в \mathbb{C}^3 .

Рассмотрим, например, алгебру Ли, имеющую базисом 6 комплекснозначных квадратных матриц 4-го порядка следующего специального вида $(\mu_1 = 2, \nu_1 = 0, \mu_2 = (1 + 2\varepsilon), \nu_2 = (1 - 2\varepsilon), m_1, m_2, m_3, m_4, m_5, m_6 = 2, \lambda, \varepsilon \in \mathbb{R})$:

$$\begin{aligned}
 E_1 &= \begin{pmatrix} A1_1 & A2_1 & A3_1 & 1 \\ B1_1 & B2_1 & B3_1 & 0 \\ 2i\mu_1 & 0 & 2m_1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, & E_2 &= \begin{pmatrix} A1_2 & A2_2 & A3_2 & i \\ B1_2 & B2_2 & B3_2 & 0 \\ 2\nu_1 & 0 & 2(m_2 - i\mu_0) & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \\
 E_3 &= \begin{pmatrix} A1_3 & A2_3 & A3_3 & 0 \\ B1_3 & B2_3 & B3_3 & 1 \\ 0 & 2i\mu_2 & 2m_3 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, & E_4 &= \begin{pmatrix} A1_4 & A2_4 & A3_4 & 0 \\ B1_4 & B2_4 & B3_4 & i \\ 0 & 2\nu_2 & 2m_4 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, & (1)
 \end{aligned}$$

$$E_5 = \begin{pmatrix} A1_5 & A2_5 & A3_5 & 0 \\ B1_5 & B2_5 & B3_5 & 0 \\ -\mu_0 & 0 & 2(m_5 + i\lambda) & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \quad E_6 = \begin{pmatrix} A1_6 & A2_6 & A3_6 & 0 \\ B1_6 & B2_6 & B3_6 & 0 \\ 0 & 0 & 2m_6 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

В работе изучается вопрос о возможности равенства $\mu_0 = 1$ для таких алгебр. Этот же вопрос представляет интерес и для 5-мерных алгебр, базис которых свободен от матрицы E_6 .

Дело в том, что алгебры Ли, отвечающие аффинно-однородным поверхностям типов $(1/2, \varepsilon)$ (см. [1]) в пространстве \mathbb{C}^3 , могут быть только 5-мерными или 6-мерными, а их базисы можно задать в виде (1) (полном или «усеченном» за счет матрицы E_6). При этом в большинстве изученных случаев однородности (см. [1], [2], [3]) параметр μ_0 , совпадающий с одним из коэффициентов аффинного канонического уравнения поверхности (см. [1]), может принимать лишь нулевое значение. Вместе с тем, в [1] имеется пример однородной поверхности, имеющей значение $\mu_0 = 1$.

Кроме того, согласно [1], значение параметра μ_0 всегда можно считать равным либо нулю, либо единице.

Формулировка первого основного результата

Основным термином, описывающим алгебру Ли, является скобка (коммутатор) произвольных двух элементов этой алгебры. Для матричных алгебр Ли эта скобка имеет вид:

$$[M_1, M_2] = M_1 \cdot M_2 - M_2 \cdot M_1. \quad (2)$$

Условие принадлежности такой скобки исходной алгебре является квадратичным соотношением относительно M_1, M_2 .

Нас будет интересовать, когда линейная оболочка шести матриц вида (1) является алгеброй Ли. Для полного решения такой задачи необходимо и достаточно изучить совокупность всех скобок вида (2) для произвольных пар базисных матриц.

Требование замкнутости линейной оболочки матриц $E_1 - E_6$ относительно коммутатора (2) позволяет получить большую систему уравнений относительно совокупности всех матричных элементов составляющих, которая сохраняет силу и в 5-мерном случае.

Исследованию таких систем при разных значениях параметров, входящих в базисные матрицы изучаемых алгебр посвящены, например, работы [1], [2], [3], [4].

Общая схема исследования из этих статей подразумевает выделение из большой системы более простых подсистем, замкнутых относительно уменьшенных наборов матричных элементов. После изучения простейшей, линейной подсистемы главным фрагментом схемы является исследование квадратичной подсистемы относительно небольшого количества матричных элементов.

Для решения нашей задачи оказывается достаточно изучить лишь линейную систему уравнений. Итогом таких рассмотрений является следующее утверждение.

Теорема 1. Для 6-мерных матричных алгебр Ли с базисами вида (1) при выполнении условия $0 < \varepsilon \neq 1/2$ параметр μ_0 не может равняться единице.

Доказательство теоремы 1

Для доказательства теоремы выделим у базисных матриц (1) их левые верхние 2×2 -блоки, а у каждого элемента из таких блоков выделим вещественную и мнимую части.

Тогда любой обсуждаемый базис примет вид ($t_k \in R$):

$$\begin{aligned}
 E_1 &= \begin{pmatrix} t_1 + it_2 & t_3 + it_4 & A3_1 & 1 \\ t_5 + it_6 & t_7 + it_8 & B3_1 & 0 \\ 4i & 0 & 2m_1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \\
 E_2 &= \begin{pmatrix} t_9 + it_{10} & t_{11} + it_{12} & A3_2 & i \\ t_{13} + it_{14} & t_{15} + it_{16} & B3_2 & 0 \\ 0 & 0 & 2(m_2 - i) & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \\
 E_3 &= \begin{pmatrix} t_{17} + it_{18} & t_{19} + it_{20} & A3_3 & 0 \\ t_{21} + it_{22} & t_{23} + it_{24} & B3_3 & 1 \\ 0 & 2i(1 + 2\varepsilon) & 2m_3 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \\
 E_4 &= \begin{pmatrix} t_{25} + it_{26} & t_{27} + it_{28} & A3_4 & 0 \\ t_{29} + it_{30} & t_{31} + it_{32} & B3_4 & i \\ 0 & 2(1 - 2\varepsilon) & 2m_4 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix},
 \end{aligned} \tag{3}$$

$$E_5 = \begin{pmatrix} t_{33} + it_{34} & t_{35} + it_{36} & A3_5 & 0 \\ t_{37} + it_{38} & t_{39} + it_{40} & B3_5 & 0 \\ -1 & 0 & 2(m_5 + i\lambda) & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix},$$

$$E_6 = \begin{pmatrix} t_{41} + it_{42} & t_{43} + it_{44} & A3_6 & 0 \\ t_{45} + it_{46} & t_{47} + it_{48} & B3_6 & 0 \\ 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

Линейные соотношения на неизвестные элементы t_k, m_j возникают при рассмотрении (3,1)- и (3,2)-элементов скобок вида (2). Первоначально с помощью пакета Maple были вычислены все такие элементы.

Более внимательное их рассмотрение позволяет ограничиться лишь тремя комплексными уравнениями. Нам понадобятся для их получения элементы (3,2) скобок $W_{23} = [E_2, E_3]$, $W_{24} = [E_2, E_4]$ и элемент (3,1) скобки $W_{34} = [E_3, E_4]$. Кроме того, учтем, что в разложении

$$M = r_1 E_1 + r_2 E_2 + r_3 E_3 + r_4 E_4 + r_5 E_5 + r_6 E_6$$

любой матрицы M из обсуждаемой 6-мерной алгебры по ее базису первые пять из шести коэффициентов определяются 4-м столбцом M .

Для интересующих нас матриц-скобок четвертые столбцы легко вычисляются и имеют вид:

$$W_{23} = \begin{pmatrix} t_{11} + it_{12} - it_{17} + t_{18} \\ t_{15} + it_{16} - it_{21} + t_{22} \\ 0 \\ 0 \end{pmatrix},$$

$$W_{24} = \begin{pmatrix} it_{11} - t_{12} - it_{25} + t_{26} \\ it_{15} - t_{16} - it_{29} + t_{30} \\ 0 \\ 0 \end{pmatrix},$$

$$W_{34} = \begin{pmatrix} it_{19} - t_{20} - t_{27} - it_{28} \\ it_{23} - t_{24} - t_{31} - it_{32} \\ -4 \\ 0 \end{pmatrix}.$$

Тогда для трех матричных элементов мы получим следующую систему уравнений:

$$\begin{cases} 2i(2m_2 - 2i)(1 + 2\varepsilon) - 2i(1 + 2\varepsilon)(t_{15} + it_{16}) - 2i(t_{15} + t_{22})(1 + 2\varepsilon) - (t_{16} - t_{21})(2 - 4\varepsilon) = 0, \\ (2m_2 - 2i)(2 - 4\varepsilon) - (2 - 4\varepsilon)(t_{15} + it_{16}) + 2i(t_{16} - t_{30})(1 + 2\varepsilon) - (t_{15} - t_{29})(2 - 4\varepsilon) = 0, \\ 2it_{29} - 2t_{30} + 4i\varepsilon t_{29} - 4\varepsilon t_{30} - 2t_{21} - 2it_{22} + 4\varepsilon t_{21} + 4i\varepsilon t_{22} + 4it_{20} + 4it_{27} - 4\mu_0 = 0. \end{cases} \quad (4)$$

Отделяя вещественные и мнимые части в каждом из трех комплексных уравнений (4), получим шесть вещественных уравнений:

$$\begin{cases} 4\mu_0 + 8\varepsilon + 8\varepsilon t_{16} + 2t_{21} - 4\varepsilon t_{21} = 0, \\ 4m_2 + 8\varepsilon m_2 - 4t_{15} - 8\varepsilon t_{15} - 2t_{22} - 4\varepsilon t_{22} = 0, \\ 4m_2 - 8\varepsilon m_2 - 4t_{15} + 8\varepsilon t_{15} + 2t_{29} - 4\varepsilon t_{29} = 0, \\ -4\mu_0 + 8\varepsilon + 8\varepsilon t_{16} - 2t_{30} - 4\varepsilon t_{30} = 0, \\ -2t_{30} - 4\varepsilon t_{30} - 2t_{21} + 4\varepsilon t_{21} - 4\mu_0 = 0, \\ 2t_{29} + 4\varepsilon t_{29} - 2t_{22} + 4\varepsilon t_{22} + 4t_{20} + 4t_{27} = 0. \end{cases} \quad (5)$$

Рассмотрим систему алгебраических уравнений, состоящую из трех уравнений системы (5), а именно элементы исправленных скобок $\text{Re}(W_{23}(3,2))$, $\text{Im}(W_{24}(3,2))$, $\text{Re}(W_{34}(3,1))$ при условии, что $\mu_0 = 1$:

$$\begin{cases} 4 + 8\varepsilon + 8\varepsilon t_{16} + 2t_{21} - 4\varepsilon t_{21} = 0, \\ -4 + 8\varepsilon + 8\varepsilon t_{16} - 2t_{30} - 4\varepsilon t_{30} = 0, \\ -2t_{30} - 4\varepsilon t_{30} - 2t_{21} + 4\varepsilon t_{21} - 4 = 0. \end{cases} \quad (6)$$

Обозначим матрицу системы (6), составленную из коэффициентов при переменных, через A . Она имеет вид:

$$A = \begin{pmatrix} 8\varepsilon & 2 - 4\varepsilon & 0 \\ 8\varepsilon & 0 & -2 - 4\varepsilon \\ 0 & -2 + 4\varepsilon & -2 - 4\varepsilon \end{pmatrix}.$$

Матрица A системы (6) имеет неполный ранг (т. к. ее определитель равен нулю), а ранг расширенной матрицы системы равен 3. Замена первого столбца матрицы A столбцом свободных коэффициентов приводит к матрице с ненулевым определителем $16 \cdot (2\varepsilon - 1)(2\varepsilon + 1)$.

По известной теореме Кронекера-Капелли система (6) несовместна. Тем самым, при условиях $0 < \varepsilon \neq 1/2$ и $\mu_0 = 1$ набор матриц (1) не может быть базисом алгебры Ли. Теорема 1 доказана.

Случай 5-мерных алгебр

Вопрос о параметре μ_0 в случае 5-мерных алгебр Ли необходимо рассматривать для «усеченного» базиса E_1, E_2, E_3, E_4, E_5 из (1). Поскольку в приведенном доказательстве теоремы 1 нигде не использовалась шестая

матрица этого базиса, оно оказывается справедливым и для 5-мерного случая.

Выводы

Объединение теоремы 1 с комментарием к 5-мерному случаю приводит к следующему основному результату нашей статьи.

Теорема 2. Для матричных алгебр Ли, отвечающих аффинно-однородным поверхностям типов $(1/2, \varepsilon)$ в пространстве \mathbb{C}^3 , параметр μ_0 может быть только нулевым.

Список литературы

1. Лобода, А. В. Об аффинной однородности поверхностей трубчатого типа в \mathbb{C}^3 / А. В. Лобода, Т. Т. З Нгуен // Труды МИАН. – 2012. – Т. 279. – С. 102-119.
2. Atanov, A. V. Affine homogeneous strictly pseudoconvex hypersurfaces of the type $(1/2, 0)$ in \mathbb{C}^3 [Электронный ресурс] / A. V. Atanov, A. V. Loboda, A. V. Shipovskaya. – Режим доступа : <http://arxiv.org/abs/1401.2252>
3. Лобода, А. В. Об одном семействе аффинно-однородных вещественных гиперповерхностей 3-мерного комплексного пространства / А. В. Лобода, А. С. Ходарев // Известия ВУЗов. Сер. Математика. – 2003. – № 10. – С. 38-50.
4. Лобода, А. В. О полном списке аффинно-однородных поверхностей $(\varepsilon, 0)$ -типов в пространстве \mathbb{C}^3 / А. В. Лобода, А. В. Шиповская // Известия ВУЗов. Сер. Математика. – 2015. – № 6. – С. 75-82.
5. Нгуен Т. Т. З. Построение 5-мерных матричных алгебр Ли с помощью пакета MAPLE / Т. Т. З. Нгуен // Вестник ВГУ Сер. Физика. Математика. – 2012. – № 1. – С. 162-170.

ПРИМЕНЕНИЕ АЛГОРИТМОВ КОМПЬЮТЕРНОГО ЗРЕНИЯ ПРИ РАЗРАБОТКЕ ЧЕЛОВЕКО-МАШИННОГО ИНТЕРФЕЙСА

В. В. Шишкалов

Студент

А. А. Крыловецкий

Доцент

Введение

В настоящее время компьютерное зрение активно применяется для решения целого ряда прикладных задач, среди которых: автоматизация управления производственными процессами [1], моделирование объектов окружающей среды [2, 3], распознавание образов [4-6], человеко-машинное взаимодействие [7] и др. Внедрение решений, основанных на применении компьютерного зрения в сфере человеко-машинного взаимодействия позволяет разрабатывать новые человеко-машинные интерфейсы. Такие интерфейсы отслеживают положение пользователя в пространстве, его движения, перемещение [7, 8] либо положение (перемещение) управляющих устройств [9, 10], и интерпретируют их в управляющие команды. При этом наибольшую сложность в оценке положения объектов в пространстве представляет определение отдаления объектов от камеры («глубины» сцены). Для решения данной задачи предложено большое количество алгоритмов, основанных на анализе изображений, получаемых с двух параллельно работающих камер [2], однако, взаимное положение и угол визирования этих камер должны быть фиксированными, что приводит к необходимости создания специальных платформ (с постоянными либо измеряемыми параметрами камер) [10, 11], поэтому зачастую разработчики прибегают к альтернативным аппаратным решениям (ИК-датчики глубины, светочувствительные линзы со светоизлучающим управляющим устройством и др.) [8, 9].

В данной работе рассматривается разрабатываемая в настоящий момент система, способная отслеживать положение (лица) пользователя в пространстве с целью оптимизации представления информации на мониторе компьютера. На данный момент существует ряд программно-аппаратных систем, используемых для решения подобных задач, и применяемых в качестве: (игровых) контроллеров (Kinect [8], PlayStation Move [9]), систем разработки и дизайна (NettleDesc [10]), систем помощи людям с ограниченными возможностями (Nouse, SmartNav и др. [7]). По отношению к перечисленным системам разрабатываемое приложение

имеет ряд особенностей, к которым относятся: определение удаления пользователя на основе анализа данных с нескольких камер (в отличие от [7, 8, 9]); расположение камер на произвольном расстоянии друг от друга и под произвольным углом (в отличие от [10]); отсутствие какого-либо управляющего оборудования вследствие слежения непосредственно за пользователем (в отличие от [9, 10]); возможность работы с любой конфигурацией пары камер (различающихся по min/max разрешению, производителю, возможности автофокуса и пр.) подключенных посредством USB (в отличие от [8, 9, 10]); возможность использования встроенной камеры (при работе с ноутбуком). Из перечисленных особенностей последние три определяют высокую ценовую доступность данной системы.

1. Обзор применяемых средств и алгоритмов компьютерного зрения

Для решения задачи распознавания образов на изображении использовался алгоритм Виолы-Джонса, реализованный в библиотеке OpenCV, работа с которой осуществлялась при помощи API-интерфейса OpenCvSharp на языке C#.

Алгоритм Виолы-Джонса (Viola-Jones) предложен П. Виола (Paul Viola) и М. Джонсом (Michael Jones) на конференции IEEE посвященной компьютерному зрению и распознаванию образов (Computer Vision and Pattern Recognition) в 2001 году [4].

Работа алгоритма Виолы-Джонса построена на следующих принципах:

1. *Интегральное представление изображений.* Изображение преобразуется в матрицу, соответствующую размеру изображения в пикселях. В каждом элементе матрицы сохраняется сумма интенсивностей всех пикселей, находящихся левее и выше данного элемента, рассчитываемая по формуле 1.

$$L_{(x,y)} = \sum_{x=0}^{i \leq x} \sum_{j=0}^{j \leq y} I_{(i,j)}, \quad (1)$$

где $L_{(x,y)}$ – текущий элемент матрицы; x, y – координаты текущего элемента матрицы; $I_{(i,j)}$ – яркость пикселя исходного изображения; i, j – координаты пикселя исходного изображения;

Интегральное представление изображений позволяет быстро вычислять сумму яркости пикселей произвольного прямоугольника (время вычисления возрастает линейно по отношению к площади прямоугольника), что позволяет в дальнейшем применять признаки Хаара, основанные на разнице этих сумм [12, 13].

2. *Признаки (примитивы) Хаара.* На изображении позиционируются наборы из смежных прямоугольных областей, продемонстрированных на рисунке 1. Вычисляется разность между суммами интенсивности пикселей темных и светлых областей (на рис 1. черные и белые области соответственно). Эта разность, а также размер и позиция признака Хаара сохраняются в виде его значения.

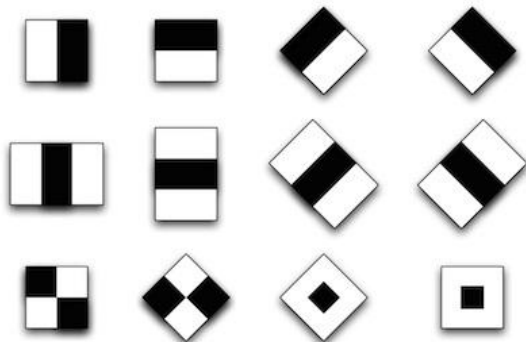


Рис. 1. Примитивы Хаара

Выбор признаков и их расположения производится в процессе обучения классификаторов. Определение наиболее значимых признаков (значения которых проявляются наиболее часто) позволяет отсеивать большую часть областей изображения, полученных сканирующим окном, не прибегая к их детальной обработке.

3. *Сканирующее окно (scanning window).* С заданным шагом (размер ячейки изображения) по изображению проходит сканирующее окно, после полного прохода размер окна поиска увеличивается, (начальный/текущий размер окна поиска + шаг увеличения размера окна поиска). Каждое положение окна передает выделенную область изображения на вход каскада классификаторов, который выдает результат «верно» либо «ложь», соответствующий наличию, либо отсутствию искомого объекта в изображении-кандидате [13].

4. *Каскад классификаторов.* Процесс обучения каскада разделяется на несколько этапов, каждый из которых соответствует обучению каждого последующего классификатора на определенном наборе данных. Первый классификатор обучается на всем наборе данных, второй – на выборке, в которой первым классификатором верный результат получен в 50% случаях, а третий – на данных, при которых результаты первых двух классификаторов различны. Подобный подход позволяет объединить “слабые” классификаторы (с процентом получения верного результата

близким к случайному) в каскад с достаточно низким процентом ошибок классификации.

OpenCV – библиотека алгоритмов компьютерного зрения с открытым исходным кодом, распространяемая по условиям лицензии BSD. Помимо алгоритмов компьютерного зрения (обнаружение объектов на изображении, анализ движения и отслеживание объектов, распознавание и описание плоских примитивов) в библиотеке реализованы алгоритмы обработки изображений, машинного обучения, численных алгоритмов общего назначения (генераторы случайных чисел, DFT, DCT), а также поддержка Nvidia CUDA и OpenCL. OpenCV доступна на C/C++, Java, Python [14]. Для работы с библиотекой посредством C# используются такие API как: Emgu CV, SharperCV и OpenCvSharp. Выбор сделан в пользу последнего по причине полной совместимости с Visual Studio 2013, и поддержки менеджером расширений NuGet [15].

2. Описание разрабатываемого программного решения

Как было описано ранее, алгоритм Виолы-Джонса используется для поиска объектов в статичном изображении, но благодаря высокой скорости работы он так же применим для поиска объектов в видеопотоке. Для этих целей из видеопотока в бесконечном цикле с определенным шагом (в нашем случае с шагом в 33 мс, что позволяет получить 30 изображений в секунду) выбираются кадры, которые конвертируются в массивы (интегральное представление) и передаются функции, которая сканирует изображение окном поиска и применяет каскад к полученным областям. В случае обнаружения объекта каскадом, функция возвращает координаты и размер области, в которой он был обнаружен. Эти данные в дальнейшем используются программой, и как показано на рис 2, могут быть использованы для выделения найденной области непосредственно в видеопотоке.

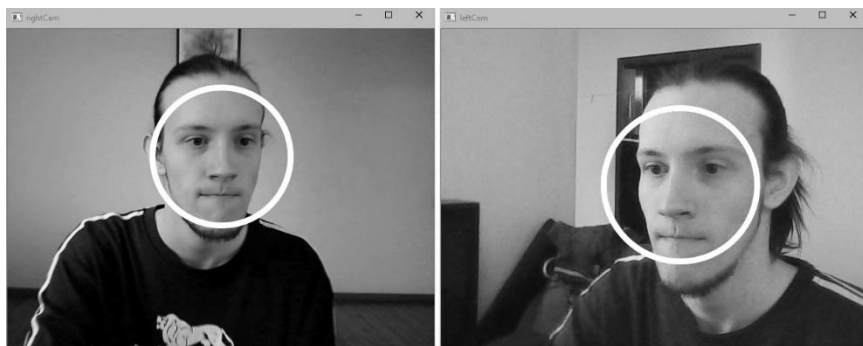


Рис. 2. Один из кадров видеопотока после процесса распознавания

Как видно из рис. 2, доступна возможность одновременной обработки данных с двух (и более) веб-камер, что позволяет определять расстояние до объектов на основе стереоскопии.

В процессе работы алгоритма Виолы-Джонса неизбежно возникают ошибки, которые могут быть следующих типов:

- имеющийся на изображении объект не распознан;
- распознан объект, отсутствующий на изображении (ложное срабатывание);
- один и тот же объект распознан более одного раза.

Примеры подобных ошибок представлены на рис 3.



Рис. 3. Ошибки при распознавании (окружностью обозначены – верные срабатывания, прямоугольником – ложные)

Для исключения подобных ошибок выполняется проверка данных текущего кадра видеопотока данными из предыдущих по алгоритму, изображенному на рис. 4.

Посредством алгоритма, описанного на рис. 4, для каждого из обнаруженных объектов мы имеем массив, хранящий его последние координаты и текущие, что позволяет постоянно отслеживать состояние всех обнаруженных объектов, а также регулярность их обнаружения в видеопотоке.

Таким образом, все ложные срабатывания (а также распознавание одного объекта дважды) игнорируются приложением, так как вероятность их появления в трех из пяти кадрах крайне мала, как и вероятность ошибки нераспознавания (негативное влияние которой частично нивелируется расчетом новой координаты на основе предыдущих). Исключением является ситуация, при которой вероятность такой ошибки резко возрастает при повороте лица более чем на 30 градусов.

Также, исходя из задачи приложения, ошибкой считается обнаружение двух и более лиц, даже если они действительно присутствуют на изображении. В данном случае приложение игнорирует лицо, появившееся позднее. Если лицо неподвижно оно также игнорируется.

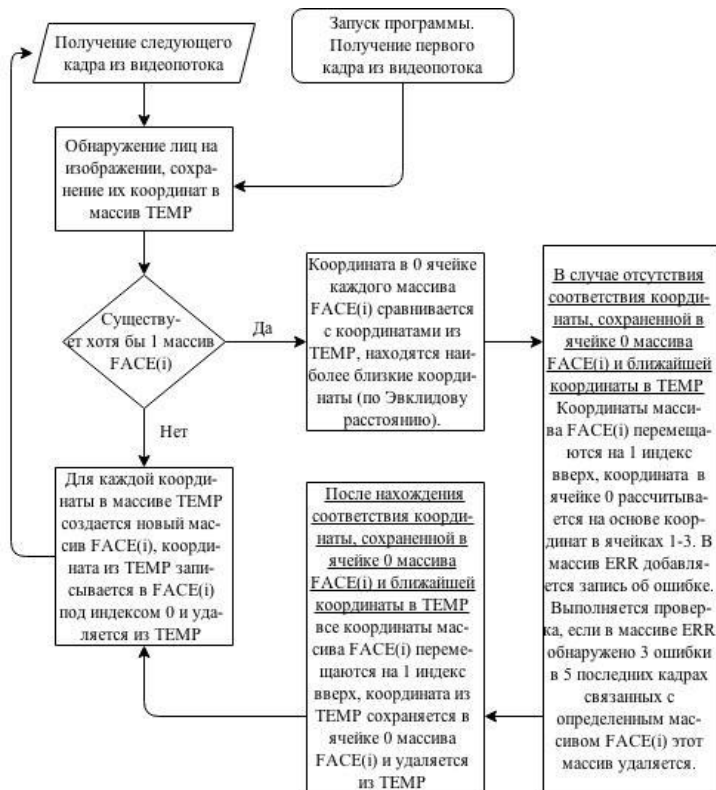


Рис. 4. Алгоритм проверки данных в текущем кадре

При первом запуске и пошаговой настройке приложения пользователю предлагается поочередно занять крайние положения (максимальное и минимальное удаление от монитора), стараясь находиться строго напротив его центра. Данные условия необходимы по причине свободного крепления камер на мониторе, главными условиями которого являются:

- расположение камер на максимально возможном расстоянии;
- установка камер под углом, позволяющим пользователю находиться в зоне их видимости при занятии крайних положений.

Рекомендованное положение камер – на верхних углах монитора (для стационарных компьютеров) или одна камера (встроенная) в центре, одна в любом углу (для ноутбуков).

В дальнейшем положение пользователя в пространстве рассчитывается относительно заданных крайних положений. Данный процесс подробно представлен на рис. 5 и рис. 6.



*A0, B0 – базовые оси; A1, B1 – максимальные оси;
A2, B2 – оси текущего изображения*

Рис. 5. Определение расстояния (ось z) до лица

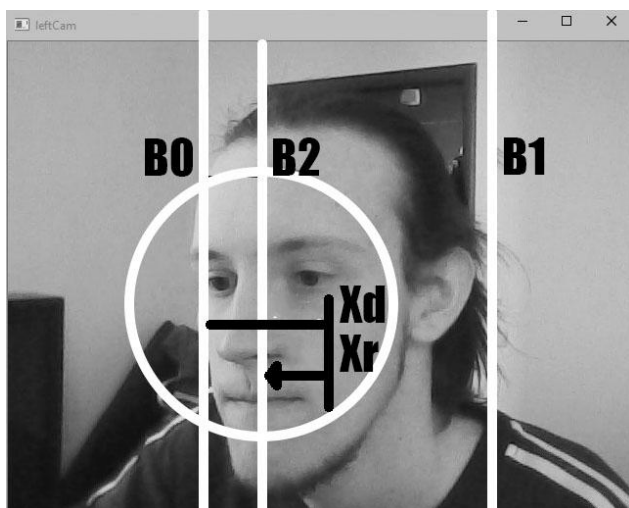
На рис. 5 верхние изображения – крайнее ближайшее положение, средние – крайнее дальнее, нижние – текущее. Для определения расстояния до лица измеряется положение текущих осей относительно базовых, результат получается в виде процентов от максимального

расстояния (между максимальными и базовыми осями). В общем виде решение представлено в виде:

$$Z = \frac{(A_2 - A_0) + (B_2 - B_0)}{(A_1 - A_0) + (B_1 - B_0)}, \quad (2)$$

где Z – расстояние до монитора, в процентах; $A_0 \dots B_2$ – расстояния осей (0 – базовой оси, 1 – максимальной оси, 2 – текущей оси) от левого края изображений с камер (A – изображения с левой камеры, B – изображения с правой камеры), в пикселях.

После получения данных о расстоянии до монитора (ось z), дальнейшим шагом является определение смещения по горизонтали (ось x), которое складывается из реального (непосредственно смещение лица в сторону) и дистанционного (смещение лица в сторону на видео при отдалении от монитора). В продолжение обработки изображения, продемонстрированной на рис. 5, на рис. 6 отображено определение смещения по горизонтали.



B_0 – базовая ось; B_1 – максимальная ось; B_2 – ось текущего положения, X_d – дистанционное горизонтальное смещение, X_r – реальное

Рис. 6. Определение горизонтального смещения (ось x) лица

В общем виде нахождение реального горизонтального смещения представлено в виде:

$$X = Z \times (B_1 - B_0) - B_2, \quad (3)$$

где X – реальное горизонтальное смещение в пикселах;

Данные вычисления выполняются на основе изображения правой камеры (камера для вычисления горизонтальных отклонений выбирается в начале работы программы и в дальнейшем не изменяется, выбор основывается на величине разницы крайних значений осей, в сторону большей). В случае, если «горизонтальной» камерой выбирается левая – значения осей B в формуле, необходимо заменить значениями осей A .

По аналогии с горизонтальным отклонением рассчитывается вертикальное.

На последнем этапе настройки приложение калибруется. Для этих целей используется тестовая фигура и набор следующих изменяемых параметров (индексов чувствительности:

- чувствительность по оси Z (расстояние);
- чувствительность по оси X (горизонталь);
- чувствительность по оси Y (вертикаль);
- зависимость горизонтального (X_d) смещения от расстояния (Z);
- начальное вертикальное положение фигуры.

Настройка происходит итеративно. Каждое изменение параметра сопровождается оценкой реакции тестовой фигуры на движения пользователя. Ниже приведен порядок действий пользователя.

Пользователь, находясь напротив центра монитора, приближается и отдаляется от него, наблюдая за тестовой фигурой. В случае, если видимый размер фигуры уменьшается при приближении (увеличивается при отдалении) следует понизить значение *индекса чувствительности по оси Z (расстояние)*, в противном случае – увеличить. Повторять действия до тех пор, пока видимый размер тестовой фигуры не станет постоянным.

Пользователь, находясь на фиксированном расстоянии от монитора движется (покачивается) из стороны в сторону, наблюдая за тестовой фигурой. В случае если фигура «обгоняет» пользователя следует понизить значение *индекса чувствительности по оси X (горизонталь)*, в противном случае – увеличить. Повторять действия до тех пор, пока тестовая фигура во время движения не будет строго напротив пользователя. В случае, если при изменении расстояния до монитора тестовая фигура начинает неправильно реагировать на движение пользователя, эта проблема решается изменением значения *индекса зависимости горизонтального (X_d) смещения от расстояния (Z)*.

Настройка *индекса чувствительности по оси Y (вертикаль)* происходит аналогично предыдущему, но с движением пользователя вверх и вниз.

Настройка *начального вертикального положения фигуры* позволяет вертикально отцентрировать тестовую фигуру на мониторах с различным наклоном.

После правильной калибровки системы фигура будет «следовать» за пользователем. Какое бы положение он не принял, до тех пор пока он находится в области видимости камер, и программа работает в штатном режиме, фигура будет всегда находится напротив глаз пользователя, и иметь фиксированный видимый размер (при этом она может частично или полностью выходить за пределы монитора). Для этого фигура значительно будет менять свой реальный размер и положение на мониторе.

Заключение

В данной статье приведено описание разрабатываемой в настоящий момент программного решения, базирующегося на алгоритмах компьютерного зрения, и способного обнаруживать положение пользователя в пространстве с целью оптимизации представления информации на мониторе компьютера.

Проведено описание применяемых методов и средств процесса работы программы, возможные ошибки и пути их устранения. Результатом работы программы является демонстрация тестовой фигуры, реагирующей на движения пользователя.

Не смотря на полученные результаты возможности данной технологии выходят далеко за рамки тестовой фигуры, так, например, настроенное приложение может использоваться в паре со средствами трехмерного моделирования для отображения разрабатываемого объекта под различными углами без применения стандартных средств (вращение мышью, изменение углов обзора «горячими» клавишами на клавиатуре). При работе приложения с графическим движком (игр, приложений) можно добиться 3D-эффекта без использования дополнительного оборудования (стерео-очков) и обработки изображения под требования данного оборудования.

Список литературы

1. Batchelor, B. G. Intelligent Vision Systems for Industry [Электронный ресурс] : Электронная книга. – Режим доступа : <http://www.eeng.dcu.ie/~whelanp/ivsi>
2. Scharstein, D. A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms / D. Scharstein, R. Szeliski // International Journal of Computer Vision. – 2002. – Vol. 47. – №1 – С. 7-42.
3. Пономарев, С. В. Методика сравнения алгоритмов стереозрения при восстановлении трехмерной модели лица человека / С. В. Пономарев //

Научно-технический вестник информационных технологий, механики и оптики – 2013. – № 6 (88). – С. 40-45.

4. Viola, P. Rapid Object Detection using a Boosted Cascade of Simple Features / P. Viola, M. J. Jones // IEEE Conf. on Computer Vision and Pattern Recognition (CVPR 2001) – Kauai : HI, 2001 – 9 с.

5. Viola, P. Robust real-time face detection / P. Viola, M. J. Jones // International Journal of Computer Vision. – 2004. – Vol. 57. – №2. – P. 137-154.

6. Кривцов, О.А. Отслеживание положения головы человека в пространстве на основе анализа видеок кадров / О. А. Кривцов, А. М. Корилов // Доклады ТУСУРа. – 2008. – № 2 (18), часть 2. – С. 32-39.

7. Вежневцев, В. Система управления курсором «мышь» при помощи движений головы пользователя / В. Вежневцев, Р. Шоргин, А. Вежневцев // Труды конференции по компьютерной графике и машинному зрению Graphicon. – 2006. – С. 455-458.

8. Content distribution regulation by viewing user / K. S. Perez, A. A. Kipman, A. J. Fuller; заявитель и патентообладатель Microsoft corporation. – US20120278904; заявл. 26.04.2011; опубл. 1.11.2012. – 19 р.

9. Flat joystick controller / E. Huang, A. Mikhailov; заявитель и патентообладатель Sony Computer Entertainment – US20100105480 A1; заявл. 22.09.2009; опубл. 29.04.2010 – 22 р.

10. Nettle, Virtual Holograms Technology [Электронный ресурс] : Официальный сайт Nettle – Режим доступа : <http://nttl.ru/technology>

11. Нгуен, Х. В. Изучение погрешности измерения стереоскопической системы в зависимости от угла визирования, величины стереобазы и расстояния от объекта до камер [Электронный ресурс] : Научно-технический интернет-журнал. – Режим доступа : http://www.telephototech.ru/kat_podr.php?stid=29&st_gr_id=5

12. Гонсалес, Р. Цифровая обработка изображений / Р. Гонсалес, Р. Вудс. – Москва : Технофера, 2005. – 1072 с.

13. Форсайт, Д. Компьютерное зрение. Современный подход / Д. Форсайт, Ж. Понс. – 2-е изд. – Москва : Вильямс, 2004. – 928 с.

14. Bradsky, G. Learning OpenCV / G. Bradsky, A. Kaehler. – 1-st edition. – Sebastopol, CA : O'Reilly, 2008. – 556 р.

15. GitHub, shimat, opencvsharp, OpenCV wrapper for .NET Framework. Хостинг и совместная разработка IT-проектов [Электронный ресурс]. – Режим доступа : <http://github.com/shimat/opencvsharp>

СОДЕРЖАНИЕ

| | |
|--|----|
| 1. Е. В. Акимова, А. В. Лобода. КОМПЬЮТЕРНЫЕ АЛГОРИТМЫ ИНТЕГРИРОВАНИЯ МАТРИЧНЫХ АЛГЕБР ЛИ..... | 3 |
| 2. А. В. Алексеев, Я. А. Туровский. РАЗРАБОТКА КОМПЬЮТЕРНОЙ СИСТЕМЫ МОНИТОРИНГА ДВИЖЕНИЯ ГЛАЗ..... | 8 |
| 3. С. С. Батищева, Д. И. Соломатин. ОБУЧАЮЩЕЕ ПРИЛОЖЕНИЕ «ПОШАГОВОЕ ВЫПОЛНЕНИЕ SQL-ЗАПРОСОВ»..... | 14 |
| 4. С. В. Беленов, Е. А. Сирота. ИССЛЕДОВАНИЕ И ПРИМЕНЕНИЕ МЕТОДОВ ШУМОПОДАВЛЕНИЯ В ЗАДАЧЕ АНАЛИЗА МОРФОЛОГИИ ПОВЕРХНОСТИ ИОНООБМЕННЫХ МЕМБРАН..... | 20 |
| 5. Б. Бендо, М. А. Дрюченко. ПРОГРАММНЫЙ МОДУЛЬ ДЛЯ РАСПОЗНАВАНИЯ СЛАБОКОНТРАСТНЫХ ОБЪЕКТОВ НА ТЕМАТИЧЕСКИХ РАСТРОВЫХ ИЗОБРАЖЕНИЯХ..... | 27 |
| 6. Н. А. Болдинова, А. С. Беляев. АДАПТИВНОЕ СВЕТОФОРНОЕ РЕГУЛИРОВАНИЕ НА МОДЕЛИ ИЗОЛИРОВАННОГО ПЕРЕКРЕСТКА..... | 34 |
| 7. Д. Ю. Большаков, Н. Н. Винокурова. ОЦЕНКА ВЕРОЯТНОСТНО-ВРЕМЕННЫХ ХАРАКТЕРИСТИК БЕСПРОВОДНЫХ СЕТЕЙ СВЯЗИ НА ОСНОВЕ МОДЕЛИ В СИСТЕМЕ NS2..... | 39 |
| 8. В. С. Бондаренко, А. П. Бубнов, А. В. Лобода. ОПИСАНИЕ ОДНОГО КЛАССА 6-МЕРНЫХ МАТРИЧНЫХ АЛГЕБР ЛИ..... | 44 |
| 9. А. В. Васильев, А. П. Толстобров. ИНТЕГРАЦИЯ ОБРАЗОВАТЕЛЬНОГО ПОРТАЛА MOODLE С ИНФОРМАЦИОННОЙ СИСТЕМОЙ УНИВЕРСИТЕТА..... | 50 |
| 10. Г. Н. Вознесенская, Д. И. Соломатин. РАЗРАБОТКА SMART TV ПРИЛОЖЕНИЯ ДЛЯ ПРОСМОТРА IPTV-ТЕЛЕВИДЕНИЯ..... | 58 |
| 11. Д. С. Дедов, И. В. Илларионов. ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ ФУНКЦИОНАЛЬНОГО МОДУЛЯ ДЛЯ РЕШЕНИЯ ОПТИМИЗАЦИОННЫХ ЗАДАЧ В SAP ERP..... | 65 |
| 12. Э. Джумари, А. А. Сирота. ПРОГНОЗИРОВАНИЕ СТОИМОСТИ ОБЪЕКТОВ НЕДВИЖИМОСТИ С ПРИМЕНЕНИЕМ НЕЙРОННЫХ СЕТЕЙ..... | 70 |
| 13. А. В. Ещенко, М. Е. Семенов. МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ УПРУГОПЛАСТИЧЕСКОГО ГИСТЕРЕЗИСА В НАНОРАЗМЕРНЫХ ФУЛЛЕРЕНОВЫХ ПЛЕНКАХ..... | 75 |

| | |
|---|-----|
| 14. О. В. Желтухина, Д. И. Соломатин. ОПРЕДЕЛЕНИЕ АВТОРСТВА ТЕКСТА НА ОСНОВЕ АНАЛИЗА ПАРНОЙ ВСТРЕЧАЕМОСТИ БУКВ..... | 83 |
| 15. М. В. Землянухин, Д. И. Соломатин. РАСПОЗНАВАНИЕ МИШЕНИ, ИЗОБРАЖЕННОЙ НА ЧАСТИЧНО ЗАКРЫТОМ АЛЬБОМНОМ ЛИСТЕ..... | 87 |
| 16. Д. А. Зяблов, В. В. Фертиков. РАСПОЗНАВАНИЕ ЛИЦ | 94 |
| 17. Ю. Ю. Иванников, Е. Ю. Митрофанова. МЕТОДЫ АНАЛИЗА ОТКРЫТЫХ ДАННЫХ ПОЛЬЗОВАТЕЛЕЙ СОЦИАЛЬНЫХ СЕТЕЙ..... | 99 |
| 18. М. С. Ильин, П. С. Лысачев. РЕАЛИЗАЦИЯ ФОНОВОГО СЕРВИСА С ВОЗМОЖНОСТЬЮ ОБРАБОТКИ СИСТЕМНЫХ СОБЫТИЙ ДЛЯ МОБИЛЬНЫХ УСТРОЙСТВ НА ПЛАТФОРМЕ ANDROID | 104 |
| 19. Г. И. Казун, А. А. Сирота. РАСПОЗНАВАНИЕ ОБРАЗОВ МЕТОДОМ ВИОЛЫ-ДЖОНСА С ИСПОЛЬЗОВАНИЕМ ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛЕНИЙ НА ОСНОВЕ ТЕХНОЛОГИИ CUDA | 109 |
| 20. Е. А. Копытина, А. И. Другалев. РАЗРАБОТКА МОБИЛЬНОГО ПРИЛОЖЕНИЯ НА ПЛАТФОРМЕ 1С:ПРЕДПРИЯТИЕ 8.3 | 115 |
| 21. В. В. Коротков, А. А. Вахтин. РАЗРАБОТКА КОМПЬЮТЕРНОГО НЕЙРОИНТЕРФЕЙСА НА ОСНОВЕ РЕГИСТРАЦИИ УСТОЙЧИВЫХ ЗРИТЕЛЬНЫХ ВЫЗВАННЫХ ПОТЕНЦИАЛОВ | 120 |
| 22. С. Г. Косачев, А. В. Атанов. АЛГОРИТМИЧЕСКИЙ ПОДХОД К ИЗУЧЕНИЮ ПОЛИНОМИАЛЬНЫХ ОТОБРАЖЕНИЙ..... | 126 |
| 23. Д. С. Лебедько, Н. А. Тюкачев. МЕТОДЫ ЧИСЛЕННОЙ ОЦЕНКИ КАЧЕСТВА ВЫДЕЛЕНИЯ ГРАНИЦ НА ИЗОБРАЖЕНИЯХ | 131 |
| 24. В. С. Левин, Ю. Б. Нечаев. РАЗРАБОТКА ПРОГРАММНО-АППАРАТНОГО КОМПЛЕКСА ДЛЯ КОНТРОЛЯ И АНАЛИЗА РАДИОЧАСТОТНОГО СПЕКТРА С ИСПОЛЬЗОВАНИЕМ SDR ТЕХНОЛОГИЙ | 138 |
| 25. Д. А. Мезин, Д. Н. Борисов. РЕАЛИЗАЦИЯ GPS-ТРЕКЕРА НА ОСНОВЕ МИКРОКОНТРОЛЛЕРА | 142 |
| 26. А. А. Назарова, В. С. Тарасов. ОПРЕДЕЛЕНИЕ ПОЛОЖЕНИЯ УСТРОЙСТВА В ПРОСТРАНСТВЕ ПО ЕГО ДАТЧИКАМ | 148 |

| | |
|--|-----|
| 27. В. А. Назарьева, М. Г. Матвеев. МЕТОДИКА КОМПЬЮТЕРНОГО АНАЛИЗА ПРОИЗВОДСТВЕННЫХ БИЗНЕС-ПРОЦЕССОВ НА ПРИМЕРЕ СКЛАДА ПРОИЗВОДСТВЕННЫХ РЕСУРСОВ | 155 |
| 28. М. Е. Ольшанская, М. А. Дрюченко. ЗАЩИТА ЦИФРОВЫХ ИЗОБРАЖЕНИЙ ОТ НЕСАНКЦИОНИРОВАННОГО ИСПОЛЬЗОВАНИЯ С ИСПОЛЬЗОВАНИЕМ «САМОРЕКОНСТРУИРУЮЩЕЙ» СТЕГАНОГРАФИИ | 161 |
| 29. Н. Н. Осолодков, С. А. Запрягаев. ПРОГРАММНАЯ ОБОЛОЧКА ДЛЯ РАСПОЗНАВАНИЯ ЛИЦ ПО 2D-СНИМКУ | 167 |
| 30. Е. А. Павлова, А. В. Лобода. КОМПЬЮТЕРНЫЕ АЛГОРИТМЫ ОПИСАНИЯ АФФИННО-ОДНОРОДНЫХ ПОВЕРХНОСТЕЙ $(1/2, \epsilon)$ -ТИПОВ В ПРОСТРАНСТВЕ S^3 | 174 |
| 31. Р. И. Петриев, Д. И. Соломатин. ОБФУСКАЦИЯ JAVA БАЙТ-КОДА | 179 |
| 32. А. А. Пешков, Ю. Б. Нечаев. ОЦЕНОЧНАЯ ХАРАКТЕРИСТИКА СЕТЕВОГО ТРАФИКА ЯЧЕЙСТОЙ СЕТИ В СИСТЕМЕ ОРNET | 186 |
| 33. А. Б. Пилюгин, А. Ю. Лукаш, Н. С. Троицкая, Ю. А. Пахомова, О. Н. Попова, С. В. Борзунов, С. Д. Кургалин, Л. П. Друганова, Т. Е. Фертикова. КОМПЬЮТЕРНАЯ ОЦЕНКА БИОЛОГИЧЕСКИХ РИТМОВ И АДАПТАЦИОННЫХ ВОЗМОЖНОСТЕЙ СТУДЕНТОВ ВОРОНЕЖСКОГО МЕДИЦИНСКОГО УНИВЕРСИТЕТА | 191 |
| 34. М. В. Полев, Д. И. Соломатин. СХЕМА РАБОТЫ ИНТЕРПРЕТАТОРА ЯЗЫКА ПРОГРАММИРОВАНИЯ С ДИНАМИЧЕСКИ РАСШИРЯЕМЫМ СИНТАКСИСОМ | 197 |
| 35. А. Г. Полтаева, П. С. Лысачёв. АЛГОРИТМ ВЫДЕЛЕНИЯ ГРАНИЦ РЕЧЕВЫХ ФРАГМЕНТОВ В АУДИОСИГНАЛАХ | 203 |
| 36. Р. И. Полянских, А. В. Распопов. МУЛЬТИПЛАТФОРМЕННОЕ НЕПРЕРЫВНОЕ ТЕСТИРОВАНИЕ ПРОГРАММНЫХ ПРОДУКТОВ ФИРМЫ UCS | 211 |
| 37. В. А. Потапова, М. Е. Семенов. ПРОГНОЗИРОВАНИЕ ФИНАНСОВЫХ РЫНКОВ С ПОМОЩЬЮ МОДЕЛИ ВЕГЕ-ИЗИНГА | 216 |
| 38. Е. В. Резниченко, М. Г. Матвеев. ПРИКЛАДНОЕ ПРОГРАММНОЕ СРЕДСТВО МОДЕЛИРОВАНИЯ МЕТОДА ВЫРАВНИВАНИЯ РЕСУРСОВ | 222 |
| 39. Е. В. Ремизова, А. В. Лобода. ПРИМЕРЫ АФФИННО-ОДНОРОДНЫХ ПОВЕРХНОСТЕЙ (ϵ, ϵ) -ТИПОВ В ПРОСТРАНСТВЕ S^3 | 228 |

| | |
|--|-----|
| 40. О. О. Решетова, М. Е. Семенов. ПРОГНОЗИРОВАНИЕ ВРЕМЕННЫХ РЯДОВ РЕГРЕССИОННЫМИ МОДЕЛЯМИ С НАСТРАИВЫМИ КОЭФФИЦИЕНТАМИ..... | 234 |
| 41. А. В. Руденко, А. В. Атанов. АФФИННО-ОДНОРОДНЫЕ ГИПЕРПОВЕРХНОСТИ ПАРАБОЛИЧЕСКОГО ТИПА С ИНДЕФИНИТНОЙ ФОРМОЙ ЛЕВИ | 240 |
| 42. С. В. Саввин, А. С. Беляев. ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ УЛИЧНО-ДОРОЖНОЙ СЕТИ..... | 244 |
| 43. И. Ю. Савченко, М. Г. Матвеев. РАЗРАБОТКА И РЕАЛИЗАЦИЯ МЕТОДА ПРОГНОЗИРОВАНИЯ РАСПРЕДЕЛЕННОЙ ПО ГЛУБИНЕ ТВЕРДОСТИ СТАЛИ..... | 250 |
| 44. Ю. И. Сафронова, А. П. Толстобров. АВТОМАТИЗАЦИЯ ПРОЦЕССА «ОТПУСК МАТЕРИАЛА» В СИСТЕМЕ SAP ERP С ТОЧКИ ЗРЕНИЯ ПРОЦЕССА ЗАКУПКИ..... | 254 |
| 45. А. Б. Тимофеев, Е. А. Сирота. АНАЛИЗ И ИССЛЕДОВАНИЕ ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ ДЛЯ РЕШЕНИЯ ЗАДАЧИ ПРОГНОЗИРОВАНИЯ ВРЕМЕННЫХ РЯДОВ | 259 |
| 46. С. И. Трушков, А. О. Новикова, Е. А. Раскина, С. Д. Курга- лин, С. В. Борзунов, Л. П. Друганова, Т. Е. Фертикова. КОМПЬЮТЕРНОЕ ТЕСТИРОВАНИЕ КАК МЕТОД ОЦЕНКИ КАЧЕСТВА ЖИЗНИ СТУДЕНТОВ | 266 |
| 47. С. И. Трушков, С. Э. Черенкова, К. А. Матвеева, А. Г. Пра- вильников, С. Д. Кургалин, Ю. Г. Притулина, С. В. Борзунов, Т. Е. Фертикова, Л. П. Друганова, А. В. Целиковский. РАЗРАБОТКА ИНТЕРНЕТ-САЙТА «ЗДОРОВЫЙ ОБРАЗ ЖИЗНИ СТУДЕНТОВ»..... | 270 |
| 48. К. И. Фисенко, С. А. Пешков, Я. А. Туровский. ИНФОРМАЦИОННАЯ СИСТЕМА ТАКТИЛЬНОЙ И ТЕМПЕРАТУРНОЙ ОБРАТНОЙ СВЯЗИ В МИОЭЛЕКТРИЧЕСКИХ ПРОТЕЗАХ | 276 |
| 49. А. С. Четкин, С. А. Запрягаев. ПРЕДВАРИТЕЛЬНАЯ ПОДГОТОВКА РЕЧЕВЫХ СИГНАЛОВ ДЛЯ РАСПОЗНАВАНИЯ | 280 |
| 50. А. Г. Чурсин, Е. А. Павлова, А. В. Лобода. О МАТРИЧНЫХ ЭЛЕМЕНТАХ ОДНОГО СЕМЕЙСТВА МАТРИЧНЫХ АЛГЕБР ЛИ | 286 |
| 51. В. В. Шишкалов, А. А. Крыловецкий. ПРИМЕНЕНИЕ АЛГОРИТМОВ КОМПЬЮТЕРНОГО ЗРЕНИЯ ПРИ РАЗРАБОТКЕ ЧЕЛОВЕКО-МАШИННОГО ИНТЕРФЕЙСА..... | 292 |

Научное издание

**СБОРНИК СТУДЕНЧЕСКИХ НАУЧНЫХ РАБОТ
ФАКУЛЬТЕТА КОМПЬЮТЕРНЫХ НАУК ВГУ**

Под редакцией Д. Н. Борисова

Выпуск 9

Подп. в печ. 24.06.2015. Формат 60x84/16. Усл. печ. л. 17,9. Тираж 250 экз. Заказ 449.

Издательский дом ВГУ

394000, г. Воронеж, пл. им. Ленина, 10

Отпечатано с готового оригинал-макета в типографии

Издательского дома ВГУ.

394000, г. Воронеж, ул. Пушкинская, 3