

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет компьютерных наук

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

В ы п у с к 12

*Под редакцией
кандидата технических наук Д. Н. Борисова*

Воронеж
Издательский дом ВГУ
2018

УДК 004.65 + 004.438.5

ББК Ч481(2)22

С23

Рекомендовано к печати

Ученым советом факультета компьютерных наук ВГУ

Сборник студенческих научных работ факультета компьютерных наук ВГУ / под ред. Д. Н. Борисова ; Воронежский государственный университет. – Выпуск 12. – Воронеж : Издательский дом ВГУ, 2018. – 382 с.

ISBN 978-5-9273-2621-1

В сборник включены научные работы студентов факультета компьютерных наук ВГУ, выполненные в 2017–2018 гг. под руководством преподавателей факультета, представленные в виде докладов и рекомендованные к опубликованию оргкомитетами студенческих научных конференций.

УДК 004.65 + 004.438.5

ББК Ч481(2)22

Научное издание

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

Под редакцией *Д. Н. Борисова*

Выпуск 12

Издано в авторской редакции

Подписано в печать 14.06.2018. Формат 60×84/16.

Усл. печ. л. 22,2. Тираж 500 экз. Заказ 353

Издательский дом ВГУ. 394018 Воронеж, пл. Ленина, 10

Отпечатано с готового оригинал-макета в типографии

Издательского дома ВГУ. 394018 Воронеж, ул. Пушкинская, 3

© Воронежский государственный университет,
2018

© Оформление. Издательский дом ВГУ, 2018

ISBN 978-5-9273-2621-1

РЕАЛИЗАЦИЯ АЛГОРИТМА ОБУЧЕНИЯ БАЙЕСОВСКИХ НЕЙРОННЫХ СЕТЕЙ НА ОСНОВЕ КОРРЕКЦИИ ОШИБКИ

М. Ю. Алексеева

Студент

С. В. Власов

Доцент

Введение

Современные проблемы искусственного интеллекта, машинного обучения, больших наборов данных, классификации, прогнозирования и другие, в которых присутствует неопределенность, с успехом решаются на основе теории Байеса, несмотря на критику [1]. Для решения задач автоматизации принятия решений в различных областях науки и практики широко используются компьютерные модели нейронных сетей [2-4]. Байесовская парадигма может быть использована для определения весов нейронной сети, используемой как классификатор, при обучении которой вычисляются апостериорные вероятности того, что объект принадлежит каждому классу. В частности, байесовские нейронные сети получили широкое применение для автоматизации постановки медицинских диагнозов [5]. Целью данной работы является анализ метода коррекции ошибки в байесовских нейронных сетях на основе корреляционного ранга гамма Гудмана-Крускала [6], использованного в стратегии обучения для повышения эффективности получения диагноза при большом количестве медицинских данных.

1. Однослойный персептрон

Рассмотрим однослойный персептрон. Обучающая выборка содержит в себе N объектов. Каждый из них кодируется как вектор $X_k = (x_1^k, \dots, x_i^k, \dots, x_p^k; y_j)$, представляющий собой i -ый признак k -го объекта, $k = 1, 2, \dots, N$. Здесь y_j – это метка класса принятия решения, к которому принадлежит объект X_k , $j = 1, 2, \dots, q$, w_{ij} – весовые коэффициенты нейронной сети. Наша задача заключается в применении корректировки ошибок при обучении для функции ошибки, задаваемой распределением апостериорной вероятности весов из теории Байеса, с использованием гамма корреляционного ранга Гудмана-Крускала, который является мерой силы ассоциации табулированных данных, ранжируемых от -1 (100% отрицательная ассоциация или точная инверсия) до $+1$ (100% положительная ассоциация или полное совпадение), нулевое значение указывает на отсутствие ассоциации.

Для каждого $k = 1, 2, \dots, N$, значения x_i^k , принадлежащие атрибуту A_i , $i = 1, 2, \dots, p$, определяются случайной величиной X_i . Пусть $F_i(x)$ – это распределение вероятностей X_i , $i = 1, 2, \dots, p$. Статистически, множество $\{x_i^1, x_i^2, \dots, x_i^N\}$ представляет собой случайную выборку длины N , соответствующую случайной величине X_i . Тогда метка класса принятия решения y_j , $j = 1, 2, \dots, q$, определяется случайной величиной Y с распределением вероятности $F(y)$, так что набор $\{y_j^1, y_j^2, \dots, y_j^N\}$ представляет собой случайный образец для категории Y .

Обучающая выборка TS содержит объекты, характеризуемые данными, которые подаются на вход – особенности атрибутов, (input = features) и теми, что должны быть на выходе – класс принятия решения (output = category). Поскольку эта информация доступна уже на этапе обучения, байесовский подход предлагает использовать ее для улучшения поиска приемлемого решения. При этом требуется оценить два отношения: (а) связь между атрибутами и классами принятия решений и (b) связь между атрибутами и ошибкой, возникающей при обучении нейронной сети

2. Теория Байеса

С точки зрения байесовского подхода, процесс принятия решений объединяет в себе предварительные (априорные) знания с информацией, полученной из наблюдений (апостериорной).

Если $P(h)$ – априорные вероятности, то есть вероятности гипотез до опыта, $P(h|D)$ – апостериорные вероятности, то есть вероятности гипотез после опыта, $P(D|h)$ – вероятность результата D , при условии гипотез H – правдоподобие (likelihood) можно вычислить по правилу Байеса:

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}, \quad (1)$$

где $P(D) = \sum_h P(D|h)P(h)$.

Учитывая, что для каждого $k = 1, 2, \dots, d$ признак x_i^k является значением атрибута A_i , $i = 1, 2, \dots, p$, относящегося к случайной величине X_i , формула Байеса будет выглядеть следующим образом:

$$P(A_i|B) = \frac{P(B|A_i)P(A_i)}{\sum_n P(B|A_i)P(A_i)}, \quad (2)$$

где $P\{B\} > 0, P\{A_i\} > 0, i = 1, 2, \dots, n$.

Байесовская классификация или принятие решения заключается в рассмотрении объектов с атрибутами $\{A_1, A_2, \dots, A_d\}$ и определении принадлежности объекта такому классу $C = C_k$, который максимизирует вероятность $P\{A_1, A_2, \dots, A_n|C_j\}$. При этом обычно предполагается независимость атрибутов и наивная байесовская классификация (Idiot's Bayes):

$$P\{A_1, A_2, \dots, A_n|C\} = P\{A_1|C\} \cdot P\{A_2|C\} \cdot \dots \cdot P\{A_n|C\}. \quad (3)$$

3. Алгоритм обучения

При обычном обучении нейронной сети коррекцией ошибки применяется метод минимизации стоимости, например, методом скорейшего спуска и обратного распространения ошибки, требующий вычисления частных производных и приводящий к так называемому дельта-правилу. В нашем случае мы используем условные вероятности вероятности $P(E(n)|w_{ij})$, где $E(n)$ – оценка ошибки на n -ой итерации, w_{ij} – весовые коэффициенты нейронной сети, вычисляемые как:

$$w_{ij}(n+1) = P(A_{ij}|E(n)). \quad (4)$$

Для начальной оценки вероятностей мы используем гамма корреляцию Гудмана-Крускала:

$$P(A_{ij}) = \Gamma(X_i, Y_j), \quad (5)$$

$$P(E(n)|A_{ij}) = \Gamma(X_i, E(n)), \quad i = 1, \dots, d; \quad j = 1, \dots, p. \quad (6)$$

Тогда семантические веса вычисляются по формуле:

$$w_{ij}(n+1) = \frac{\Gamma(X_i|E(n))\Gamma(X_i|Y_j)}{\sum_d \Gamma(X_i|E(n))\Gamma(X_i|Y_j)}, \quad (7)$$

где $i = 1, \dots, d; \quad j = 1, \dots, p$.

Блок-схема алгоритма представлена на рис. 1 и рис. 2.

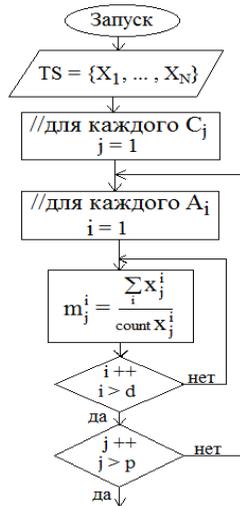


Рис. 1. Блок-схема алгоритма (начало)

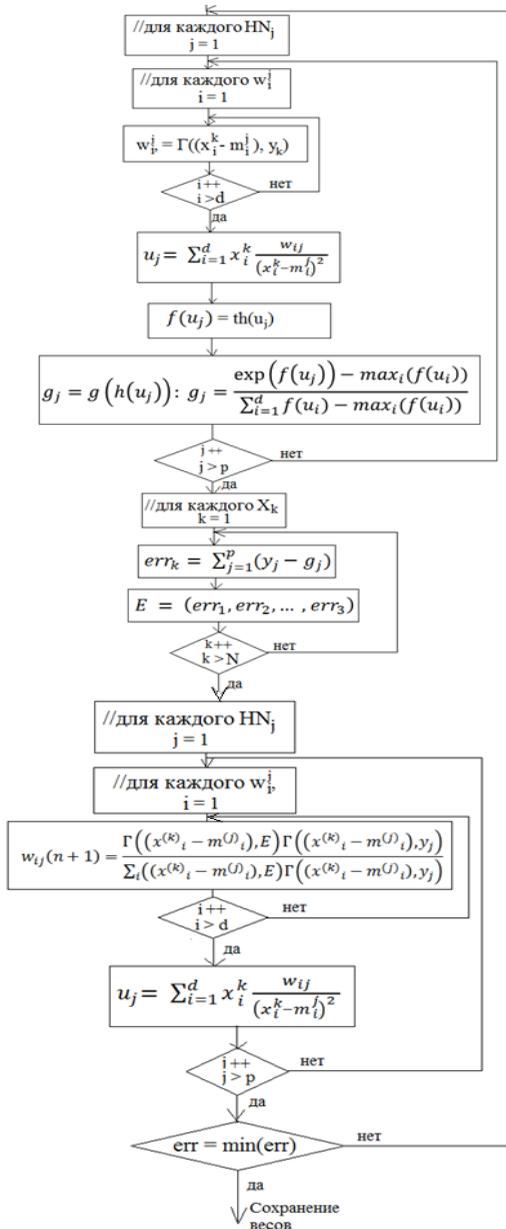


Рис. 2. Блок-схема алгоритма (продолжение)

4. Результаты тестирования

Поскольку рассматриваемый алгоритм имеет стохастический характер, он должен быть независимо запущен несколько раз, чтобы результат получился надежным и эффективным.

Анализ производительности модели был проведен с использованием метода скользящего контроля, который помогает предотвратить проблему переобучения модели. Средняя точность, рассчитанная как процент правильно выполненной классификации, рассматривается как результат принятия решения. Результаты эксперимента в процентном отношении показаны в таблице.

Таблица

Вероятность модели

Набор данных	BCWD	BC	BCWP	LC	ECHO	PID
Качество обучения	84.07	77.53	83.03	66.48	81.34	78.10
Качество тестирования	81.14	76.99	81.31	62.88	79.04	76.17

Наборы данных для обучения были выбраны из открытой базы данных, представленной на сайте: <http://archive.ics.uci.edu/ml/datasets.html>. BCWD, BC, BCWP – различные данные диагностики рака молочной железы с девятью атрибутами и двумя классами принятия решений каждый, LC – данные диагностики рака легких с 56 атрибутами и тремя классами принятия решений, ECHO – данные диагностики эхокардиограммы сердечного приступа с 10 атрибутами и двумя классами принятия решений, PID – диагностики диабета с 7 атрибутами и двумя классами принятия решения.

В зависимости от каждого из наборов данных, точность обучения/тестирования, полученная как среднее из 100 независимых полных циклов скользящего контроля, колеблется от 66.48% до 84.07% и от 62.88% до 81.31% соответственно. Стоит отметить, что небольшая разница между обучением и результатами тестирования, независимо от набора данных, доказывает надежность данного алгоритма.

Заключение

Реализованный алгоритм с успехом может использоваться для поиска приемлемого решения при автоматизации установления медицинского диагноза. Надежность алгоритма доказана небольшой разницей между обучением и тестированием, независимо от набора данных. Стабильность алгоритма доказана его работой независимо от типов данных, поскольку предполагает использование только коэффициента ранговой корреляции. В результате тестирования алгоритма на нескольких наборах данных было выявлено, что точность классификации данного алгоритма напрямую зависит от наборов данных и соответствующих атрибутов.

Список литературы

1. Efron, B. Theorem in the 21st / B. Efron // Science. – 2013. – Т. 340. – С. 1177-1178.
2. A Brief Introduction to Neural Networks [Электронный ресурс]. – Режим доступа : http://www.dkriesel.com/en/science/neural_networks
3. Bishop, C. Neural Networks for Pattern Recognition / C. Bishop // Oxford. – 1995. – P. 385-429.
4. Applications of neural networks Networks [Электронный ресурс] – Режим доступа : <https://cs.stanford.edu/people/eroberts/courses/soco/projects/neural-networks/Applications/index.html>
5. Belciug, S. Error-correction learning for artificial neural networks using the Bayesian paradigm. Application to automated medical diagnosis. / S. Belciug, F. Gorunescu // Journal of Biomedical Informatics. – 2014. – vol. 52. – P. 329-337.
6. Goodman, L. Measures of Association for Cross Classifications // L. Goodman, W. Kruskal // Journal of the American Statistical Association. – 1954. – V. 49. – P. 732-764.

РАЗРАБОТКА МОДУЛЕЙ КОРПОРАТИВНОЙ ТЕЛЕФОНИИ НА БАЗЕ ОТКРЫТЫХ ПЛАТФОРМ FREESWITCH И KAZOO 2600 HZ

О. В. Бердников
Студент
А. И. Чекмарёв
Ассистент

Введение

Для большинства организаций потребность в услугах телефонной связи не ограничивается стандартным набором опций, необходимых физическому лицу. В связи с этим существует практика использования корпоративных АТС с расширенной функциональностью, обеспечивающих потребности таких организаций. Современные коммуникационные системы, продукты и услуги могут помочь малым и средним предприятиям вести бизнес, увеличить объем продаж, продвигать услуги, повысить эффективность бизнеса [1].

Существуют различные варианты организации корпоративной телефонии внутри организации:

- Классическая офисная АТС. Самостоятельный аналоговый или цифровой телефонный коммутатор;
- Телекоммуникационный аутсорсинг [2];
- IP-АТС. Программно-реализованные АТС.

Сегодня всё большую популярность приобретают именно IP-АТС. Это связано с простотой масштабируемости такой телефонной сети, относительной дешевизной, простотой внедрения и управления. IP-АТС также подразделяются на коммерческие и бесплатные. На сегодняшний день существует несколько различных вариантов бесплатных IP-АТС, тем не менее, функциональных возможностей большинства из них недостаточно для удовлетворения всех нужд организаций, основными из которых являются:

- Интеллектуальное распределение вызовов;
- Объединение филиалов в единую систему;
- Высокая надежность и безотказность;
- Групповой вызов
- Сопряжение с телефонной сетью общего пользования [3].

1. Цель работы

Основной проблемой всех бесплатных IP-АТС является отсутствие гибкости в интеллектуальном распределении вызовов. Из-за этого развернуть Call-центр, который подходил бы под потребности большинства организаций, на базе открытых платформ практически невозможно. Существует возможность реализовать функции распределения вызовов с помощью написания скрипта с соответствующей логикой для платформы FreeSWITCH, но такой подход имеет ряд недостатков. Так, например, невозможно было создать несколько точек подключения для пользователя (например, Web и SIP). В связи с этим было принято решение реализовать логику Call-центра для платформы 2600Hz. Выбор платформы обоснован тем, что в ней используется принципиально другой подход в создании пользователей. Отличительной чертой данного подхода является возможность присваивания пользователю различных устройств, доступных по одному номеру.

2. 2600Hz KAZOO

2600Hz Kazoo – масштабируемая программная платформа с открытым исходным кодом, предназначенная для обеспечения функций VoIP-коммутатора. Платформа предоставляет возможность реализовывать гибкие и надёжные телекоммуникационные решения [4].

Также 2600Hz Kazoo уже имеет графический интерфейс и реализованный API для взаимодействия с ядром программы. Структура платформы показана на рис. 1. Все компоненты могут быть развёрнуты на

разных серверах. Видно, что данная платформа использует FreeSWITCH в качестве мультимедиа-сервера и инструмента для транскодирования.

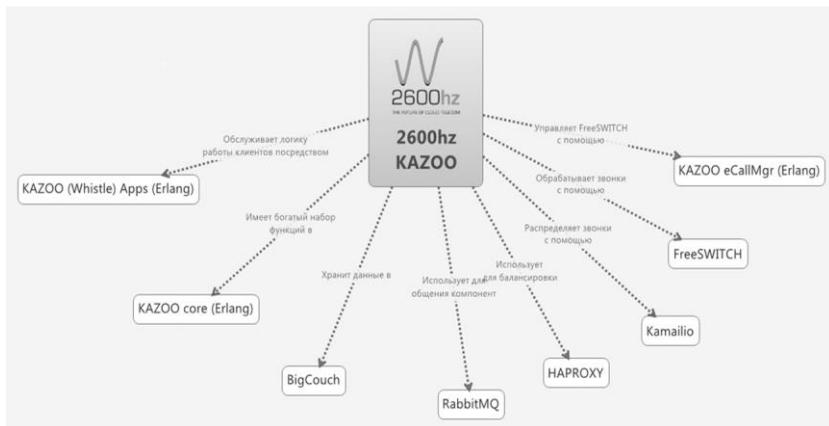


Рис. 1. Структура платформы 2600 Hz

В 2600Hz Kazoo присутствует логика работы Call-центра, но её недостаточно для выполнения сложного распределения вызовов. В связи с этим было решено дополнить существующее распределение вызовов собственной логикой.

3. Реализация

В данном случае реализация заключается в создании промежуточного сервера, который получает события, генерируемые ядром платформы, применяет к ним соответствующую логику и отправляет ответ обратно ядру. На промежуточном сервере будет развёрнуто приложение, которое будет осуществлять распределение вызова. Таким образом, написание дополнительного приложения не сказывается на функциональности платформы 2600Hz Kazoo, так как создание отдельного сервера позволяет говорить о распределённости и масштабируемости платформы в целом.

Конечная схема работы приложения приведена на рис. 2. Видно, что логика звонка задаётся именно промежуточным сервером.

Для оповещения сервера о событиях был выбран механизм Pivot, встроенный в платформу 2600Hz Kazoo. Принцип его работы похож на работу Webhook, однако его преимущество над стандартным Webhook состоит в том, что для его функционирования не требуется дополнительно удерживать вызов до применения необходимой логики и выстраивать очередь ожидания, так как все эти действия выполняются автоматически.

Для работы достаточно направить вызов на этапе создания в Pivot и сконфигурировать его таким образом, чтобы он отсылал запрос с событием на созданный нами сервер с приложением. То же самое справедливо и при сравнении Pivot с WebSocket.

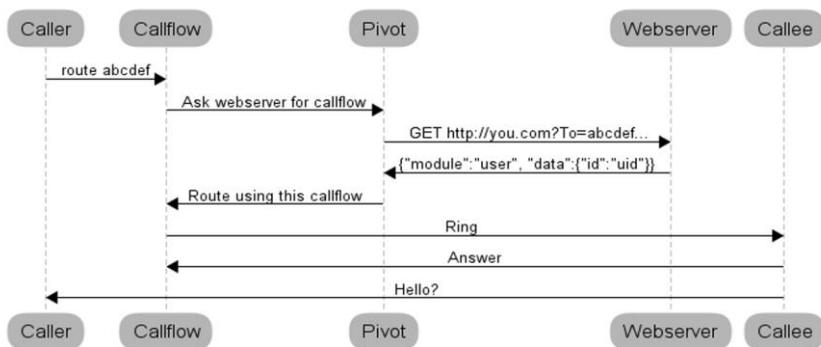


Рис. 2. Схема работы приложения с промежуточным сервером

Для взаимодействия между ядром и сервером был выбран JSON, хотя также существует возможность использовать XML.

В качестве целевого сервера был выбран стандартный Apache Web Server.

В качестве языка написания приложения был выбран Python.

Интерфейс Kazoo, который упоминался ранее, выступает в качестве панели администратора, на которой можно создавать и редактировать профили пользователей, а также сопоставлять их с группами и устройствами.

В ходе работы было разработано и реализовано три различных варианта распределения вызовов:

- Приоритет пользователя. Используется в случае, если пользователи в группе имеют разные приоритеты, и пользователям с большим приоритетом вызовы должны распределяться в последнюю очередь, лишь при занятости пользователей с меньшими приоритетами;

- Состояние пользователя. Используется для реализации различных состояний пользователей. Необходимо для того, чтобы не распределять звонки на пользователей в состоянии «Не доступен» и распределять звонки на пользователей в состоянии «Занят» в последнюю очередь;

- История звонков. Распределение осуществляется на основании истории предыдущих разговоров пользователя. Проще всего представить это на примере работы технического отдела. Если пользователь уже

общался с оператором, который решил его проблему, то логично направить пользователя вновь на того же оператора.

Данные варианты распределения не являются взаимоисключающими, и различные комбинации могут использоваться для коммутации внутри различных групп. Так, например, для технического отдела было предложено использовать цепочку распределений, представленную на рис. 3.



Рис. 3. Цепочка распределения вызовов на примере технического отдела

То есть, сначала выбираются только те пользователи, которые имеют состояние «Готов к разговору», после чего определяется их приоритет и очередь распределения. После этого производится проверка истории звонков пользователя. Если найден сотрудник, с которым уже связывался клиент, то данный сотрудник становится первым в очереди.

Заключение

Данная статья посвящена разработке модулей корпоративной телефонии. В ходе работы было разработано и реализовано приложение, решающее проблему отсутствия интеллектуального распределения вызовов на платформе 2600 Hz Kazoo. Указанное решение не является единственным. В тексте статьи также приводятся различные варианты, доступные для данной платформы. Указанное же решение является наиболее удобным, так как допускает возможность масштабирования и переносимости, а также гарантирует корректное распределение вызовов за счёт механизма Pivot.

Список литературы

1. Гольдштейн, Б.С. От сервисного обслуживания и технической поддержки телекоммуникационного оборудования к аутсорсингу эксплуатации операторских сетей / Б.С. Гольдштейн, А.Б. Гольдштейн // Журнал «Connect! Мир связи». – 2009. – № 9. – С. 2-6.
2. Орлов Сергей. Современная телефония для современного бизнеса [Электронный ресурс] / Сергей Орлов // Журнал сетевых решений/LAN. – Электрон. журн. – Режим доступа: <https://www.osp.ru/lan/2015/10/13047298>
3. Мухин, С. В. Какой должна быть УАТС [Электронный ресурс] / С. В. Мухин // Сети и системы связи. – Электрон. журн. – Режим доступа : http://www.ccc.ru/magazine/depot/04_05/read.html?0304.htm
4. 2600Hz Open Source VoIP [Электронный ресурс] / 2600hz.org. – Режим доступа : <https://www.2600hz.org>

РАЗРАБОТКА ОКРУЖЕНИЯ РАБОЧЕГО СТОЛА АДАПТИРОВАННОГО ДЛЯ ЛЮДЕЙ ПОЖИЛОГО ВОЗРАСТА

Р. И. Бобков

Студент

С. А. Зуев

Доцент

Введение

Мы живем в прекрасное время! Время – когда практически каждый гражданин России может позволить себе какую-либо компьютерную технику. Будь то смартфон, компьютер, смарт-часы или какое-либо другое устройство. Многие из молодых людей не могут себе представить жизнь без смартфона или ПК. Мы пользуемся компьютером для того, чтобы узнавать актуальные новости, прогноз погоды, пообщаться в Интернет со своими друзьями и просто знакомыми. Молодые люди делают это не задумываясь, как само собой разумеющееся. Для них не составляет ничего сложного освоить какой-либо новый интерфейс и воспользоваться основными функциями только что установленного приложения.

Однако, люди пожилого возраста испытывают сильные трудности при работе с различного рода устройствами, будь то смартфон, планшет или персональный компьютер. Они теряются при виде рабочего стола даже Windows XP, а Windows 8 или Windows 10 просто ставит их в ступор.

1. Требования к интерфейсу операционной системы

Для комфортной работы пользователя пожилого возраста необходимо обеспечить следующие условия:

- максимальная простота интерфейса, нет лишних функций;
- отсутствие всплывающих окон и меню, которые могут напугать пользователя;
- интерфейс должен быть выполнен строго, без каких-либо анимаций и в одном цветовом решении;
- должна быть возможность использования компьютерной мыши, так и сенсорного экрана;
- разграничение прав на пользователя и суперпользователя;
- доступ к суперпользователю по паролю, который имеют кураторы (родные пользователя/техподдержка);
- невозможность установки/удаления/настройки/скачивания приложений без прав суперпользователя;

- одно рабочее приложение на экране, без возможности свернуть его;
- невозможно открыть другое приложение, не закрыв запущенное;
- работа интерфейса на слабых компьютерах/компьютерах без видеокарты.

2. Концепция интерфейса

На рис. 1 можно видеть концепцию интерфейса, все построено в минималистичном стиле. Планируется использование модифицированного LXDE (Lightweight X11 Desktop Environment) [1].



Главный экран пользователя после запуска системы

Рис. 1. Интерфейс пользователя

Дополнительные ограничения были выявлены при консультациях на кафедре психологии ВГУ, путём анкетирования людей пожилого возраста, которые имели опыт освоения компьютеров в последнее время.

На главном экране расположены основные необходимые пользователю приложения (таблица).

3. Концепция системы

В целом должен быть создан огромный стек программного и аппаратного обеспечения. Будет реализован центральный сервер (серверы), через который (которые) и будет проходить интернет-трафик всех пользователей (рис 2).

Основные приложения дистрибутива

Интернет	Интернет-браузер, с функцией безопасного поиска и серфинга интернет-страниц. Включена фильтрация Adult-контента и мошеннических сайтов. Трафик идет через специальный сервер, который сжимает контент на сайте и передает его в сжатом виде пользователю. Тем самым мы экономим интернет-трафик пользователя и ускоряем загрузку страниц. Основан на WebKit (Chromium).
Файлы	Тривиальный менеджер файлов. Под режимом пользователя исключает загрузку и установку исполняющихся файлов. Запрещает загрузку файлов .exe. Не предоставляет доступ к системным директориям. Автоматически сортирует файлы по папкам: «Музыка», «Картинки», «Документы», «Прочее» основываясь на расширениях файлов. Основан на PCManFM (PCMan File Manager).
Блокнот	Обычный текстовый редактор, с минималистичным интерфейсом, основанный на abiWord.
Почта	Минималистичный почтовый клиент, который основан на Geary.
Видеочат	Текстовый и видео чаты, аналог Skype с собственной системой регистрации. Будет создан веб-клиент, для возможности общения без установки приложения. Также будет разработан клиент для ОС Windows. Используемый протокол – Jingle (дополнение XMPP [2]) – поддерживает передачу голоса, видео и файлов по сети. Клиент основан на Pidgin.
Настройки	Самое важное приложение в системе. Обычный пользователь будет иметь доступ к малозначительным настройкам системы: фоновое изображение рабочего стола и цветовой гаммы, настройка времени, переключение языка. Подключение к сети Wi-Fi. Запуск обновления системы. Выход и авторизация пользователя. Суперпользователь будет иметь расширенные права – установка/настройка/удаление приложений. Установка и настройка новых подключенных устройств. Изменение прочих системных настроек. Приложение «Настройки» планируется разработать с нуля.

Серверы должны выполнять следующие функции:

- Контроль удаленного подключения к компьютеру пользователя. Подключение будет происходить с помощью протокола SSH;
- Фильтрация и сжатие интернет-трафика пользователя (исключение загрузки сайтов мошенников и adult-сайтов);
- Хранение учетных записей пользователя;
- Ведение логов по каждому пользователю;
- Раздача обновлений с сервера;
- Установка приложений с сервера.

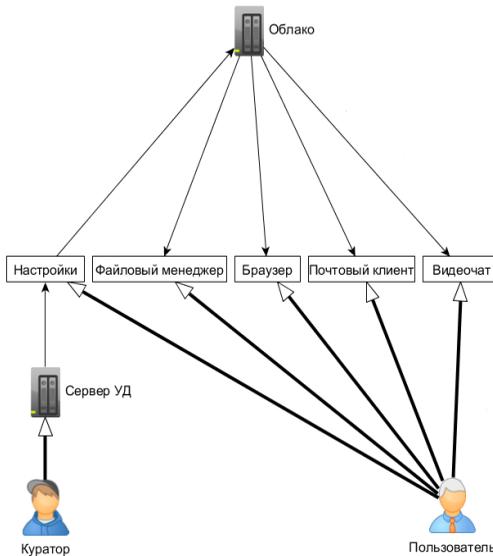


Рис. 2. Общий вид системы

Все это позволит контролировать систему и выявлять ошибки пользователя/операционной системы/приложение в оперативном режиме. Так же это позволит безопасно подключаться к компьютеру пользователя и производить манипуляции с настройками, доступ же к файлам пользователя будет ограничен.

4. Безопасность системы

Безопасность пользователей будет осуществляться с помощью различных утилит, которые настроены и готовы к работе “из коробки” [3].

Для пожилого пользователя, как уже было ранее сказано, необходимо обеспечить невозможность удаления многих системных файлов. Более

того, необходимо запретить полностью модификацию всех файлов, кроме тех, которые пользователь сам загрузил в процессе работы за ПК. С этим вопросом поможет стандартная утилита `chmod` [3]. `chmod` - программа, которая настраивает параметры доступа для файлов и папок. Используя `chmod`, мы можем закрыть доступ ко многим необходимым системным и прикладным файлам и программам, разрешить читать, изменять и запускать их только из под `root`-пользователя. Это позволит оградить пользователя от случайного удаления/перемещения/изменения важного файла и соответственно повысит отказоустойчивость системы и софта. Во многих современных дистрибутивах Linux этот инструмент очень активно используется для обеспечения безопасности файлов.

Необходимо ограничить возможность скачивания пользователем файлов всех разрешений, кроме файлов картинок, музыки, видео. То есть, делаем невозможным скачивание файлов, которые могут потенциально нанести вред системе. Скачивание же необходимых приложений можно осуществить в специальном “Центре приложений”, аналог Google Play и App Store.

Также необходимо настроить DNS-сервера. Компания Яндекс предоставляет услуги DNS-сервиса, который имеет такие полезные вещи как: фильтрацию контента для взрослых и имеет базу данных вредоносных сайтов на которые доступ пользователю будет закрыт. Пожилые пользователи при таком подходе не попадутся на уловки мошенников только лишь потому, что доступа к сайтам мошенников, у бабушки и дедушки, не будет. Соответственно и фильтрация `adult`-контента, этаким детский контроль, только для наших пожилых пользователей.

И, пожалуй самое главное, это разграничение на простого пользователя и суперпользователя [3]. Под простым пользователем невозможно будет даже минимально настроить дистрибутив - изменить время и дату, поменять цветовую гамму интерфейса, установить какие-либо новые приложения или удалить старые. За все изменения дистрибутива будет отвечать суперпользователь, то есть куратор коим будет являться близкий родственник или техподдержка. Это позволит максимально повысить отказоустойчивость системы.

Заключение

В статье рассмотрен подход, основанный на существенном упрощении вида рабочего стола для пользователей пожилого возраста и «упрятывании» функций управления системой от доступа самого пользователя.

Естественно, такая операционная система и ее интерфейс в частности, не могут гарантировать быстрого и беспроблемного освоения любым пожилым пользователем компьютера. Однако, возможность удаленного

доступа для исправления ошибок и помощи пользователю существенно экономит время и денежные средства куратору. Кроме того, данная совокупность программного обеспечения будет распространяться под лицензией GNU GPL, то есть open source, а значит, за нее не нужно платить ни рубля! Учитывая финансовое состояние большинства людей пожилого возраста — этот факт является неоспоримым преимуществом.

Список литературы

1. Russel J. LXDE / Jesse Russel, Ronald Cohn. – New York : Wiley, 2012. – 712 p.
2. XMPP [Электронный ресурс]. – Режим доступа : <https://xmpp.org/>
3. Andrew Tanenbaum S. Modern Operating System / Tanenbaum S. Andrew // Boston : Pearson Education Inc. – 2009. – Vol. 3. – 552 p.

GARBAGE COLLECTION – ОБЛЕГЧЕНИЕ РАБОТЫ ПРОГРАММИСТА?

Р. И. Бобков

Студент

С. А. Зуев

Доцент

Введение

В таких языках, как С или С++ нет встроенного сборщика мусора, поэтому задача освобождения и управления памятью лежит полностью на программисте. Грамотная работа с памятью играет одну из важнейших ролей (если вообще не является самой главной) в производительности приложения, и, соответственно, системы, в которой выполняется приложение, в целом. Эта работа включает в себя такие процессы как: Распознавание, когда созданные объекты больше не нужны и их можно спокойно удалить. И, непосредственно, само удаление более неиспользуемых объектов для освобождения памяти, которую в дальнейшем мы сможем в дальнейшем использовать. Работа с памятью даже для опытного программиста — задача сложная и уж точно нетривиальная, так как один неосторожный шаг может вызвать непредвиденное и неконтролируемое поведение программы или даже сломать ее полностью [1].

Классические языки программирования, например, С, позволяют освобождать используемую память вручную. В С для этого существует функция free().

Явное управление памятью неизбежно ведет к возникновению проблем.

1. Проблемы явного управления памятью

Итак, у вас есть программа, в которой вы пытаетесь освободить память, которая выделена под связный список. Допустим, имеется ошибка в коде, которая обрабатывает освобождение памяти головного элемента списка (освобождает память выделенную под головной элемент списка). Тогда оставшиеся элементы списка становятся потерянными - на них просто не существует ссылки. В связи с этим – мы никогда не сможем ни восстановить, не удалить их окончательно – они утеряны до перезапуска программы, а соответственно и утеряно полезное пространство памяти. Такие небольшие потери и называются утечкой памяти.

Висячие ссылки же появляются в, например, таком случае: программист освобождает память, используемую объектом, к которому у какого-либо другого объекта все еще есть ссылка. Тогда попытка доступа к этому другому объекту в будущем приведет к непредсказуемому поведению. Более наглядное объяснение выглядит следующим образом: предположим, есть два объекта, у которых есть ссылки на одно и тоже выделенное место в памяти, как показано на рис. 1.

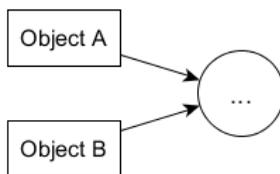


Рис. 1. Два объекта с одной ссылкой

Но после того, как разработчик освобождает память, удаляя Object A – Object B начинает ссылаться на пустое пространство памяти, как показано на рис. 2, что соответственно приведет к непредсказуемым последствиям.

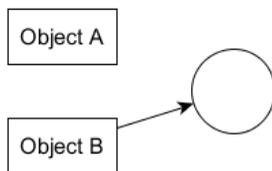


Рис. 2. Потеря ссылки

2. Garbage Collection

Во многих современных языках программирования высокого уровня, например, таких как Java или C# имеется встроенный сборщик мусора, который освобождает программиста от ручной рутинной работы с памятью. Например, в Java не нужно явным образом управлять жизненным циклом объектов [2]. Объекты создаются по мере необходимости, и в случае, когда объект больше не используется, JVM автоматически удаляет объект и освобождает занятую им память. Объект удаляется в том случае, если на него больше нет ссылок, и никто не может получить к нему доступ. Таким образом, сборщик мусора на достаточно хорошем уровне помогает работать с памятью.

Базовый процесс работы сборщика мусора представлен на рис. 3.

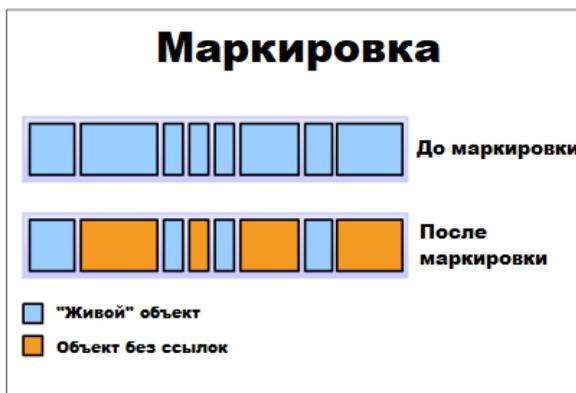


Рис. 3. Маркировка

На этом этапе сборщик мусора идентифицирует, какие из существующих объектов используются, а какие – нет [3].

На рис. 4 представлен этап «нормальное» удаление.

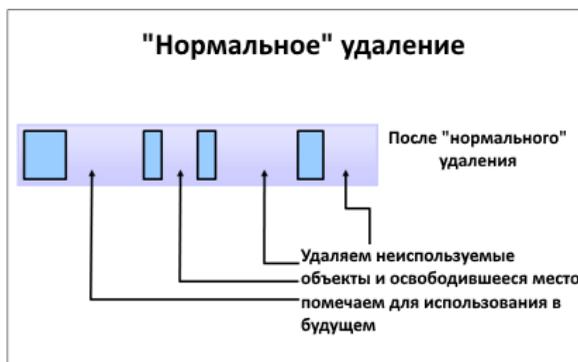


Рис. 4. «Нормальное» удаление

На данном этапе сборщик мусора удаляет неиспользуемые объекты и освобождает пространство памяти, которое можно будет использовать в дальнейшем для размещения новых объектов [3].

И, наконец, последний этап «уплотнение» представлен на рис. 5.

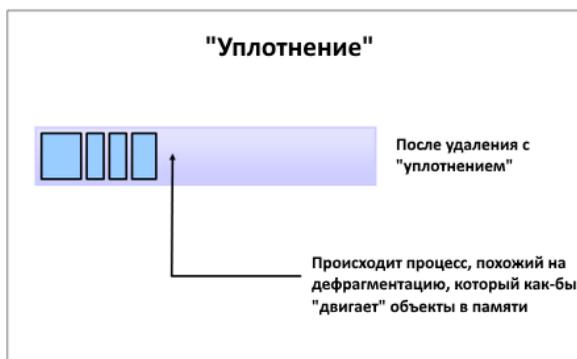


Рис. 5. «Уплотнение»

Так как все существующие объекты “разбросаны” в памяти, то необходимо их скомпоновать вначале кучи. Это необходимо для предотвращения ситуации, когда созданный в будущем объект не сможет поместиться ни в одну из освобожденных областей в памяти ввиду своего большого размера. Именно для этого и происходит уплотнение - для того чтобы предотвратить такой сценарий.

3. Типичные ситуации, которые обрабатывает GC

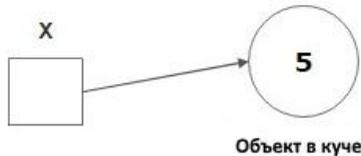
Немного примеров, для более полного понимания как работает Garbage collector.

1. Присвоение переменной null. Если мы присвоим переменной значение null, то сборщик мусора удалит такую переменную.

Объект, которому присвоено действительное значение не подходит для удаления соответственно (листинг 1, рис 6.)

Листинг 1

```
Integer x = new Integer(5); // рис. 6
```



Integer x = new Integer(5)

Рис. 6. Состояние объекта

Но, после присвоения `x` значения `null` (рис 7), сборщик мусора сделает свое дело в процессе "чистки".



x = null

Рис. 7. Состояние объекта

Переназначение ссылки, реализуется на основе кода, указанном в листинге 2.

Листинг 2

```
StringBuffer s1 = new StringBuffer("Garbage");  
StringBuffer s2 = new StringBuffer("collection");
```

В данном случае у нас есть два объекта и на оба объекта существуют ссылки (рис. 8).

Теперь объект, на который ранее ссылался `s1` остался без ссылки, а значит при следующем проходе сборщика мусора он будет удален.

2. Анонимные объекты. Поскольку ссылка на анонимный объект нигде не хранится, то такой объект должен быть удален сборщиком мусора. Наглядный пример, показан в листинге 3.

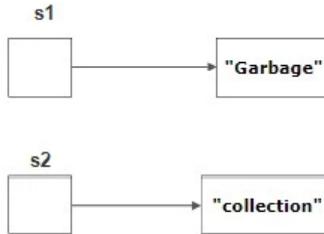


Рис. 8. Первоначальное состояние

Но после присвоения s1 значения s2 произойдет следующее (рис. 9).

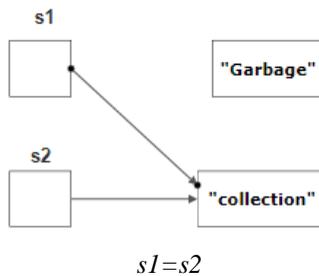


Рис. 9. Присваивание переменной

Листинг 3

```
class Employee {
    String empName;
    public Employee(String name) {
        this.empName = name;
    }
    public static void main(String[] args) {
        new Employee("Alla"); // анонимный
        объект без ссылки
    }
}
```

3. Island of Isolation. Island of Isolation – или по-другому “остров изоляции” возникает в таких случаях, как, например: класс имеет переменную, которая является ссылочной переменной другого экземпляра того же класса [4]. Представим, что существует три объекта, и они относятся к друг другу, как показано на рис. 10. Если исходные ссылки на объекты удаляются, то даже если они имеют действительную ссылку, они все равно недоступны. Это состояние и называется “остров изоляции”.

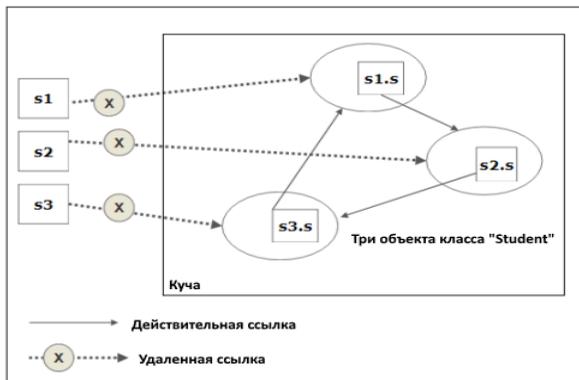


Рис. 10. Демонстрация «острова изоляции»

Заключение

В данной статье кратко рассмотрены основные причины утечки памяти и метод борьбы сборщика мусора с данными причинами.

Сборщик мусора является отличным решением, которое действительно берет на себя 99% работы с памятью.

Однако, при всех плюсах есть и некоторые существенные минусы. Например, реализация сборщика мусора зачастую возможна только если программа выполняется в виртуальной машине, например JVM. Из-за этого производительность приложения, написанного на Java, будет много хуже чем производительность аналогичного приложения, написанного на классическом языке программирования, таком как С или С++.

Однако, учитывая производительность существующих компьютеров – для рядового пользователя конечного приложения разница в скорости работы будет незаметна.

Список литературы

1. Robert Martin C. Clean Code: A Handbook of Agile Software Craftsmanship / Martin C. Robert. – Boston : Pearson Education Inc, 2008. – 464 p.
2. Herbert Schildt. Java: A Beginner`s Guide 5th Edition / Schildt Herbert. – New York : The McGraw-Hill Companies Inc, 2012. – 729 p.
3. Habr. Garbage Collection наглядно [Электронный ресурс] – Режим доступа : <https://habr.com/post/112676/>
4. Oracle Tutorials [Электронный ресурс] – Режим доступа : <http://www.oracle.com/webfolder/technetwork/tutorials/obe/java/gc01/index.html>

СЛОЖНОСТЬ РАЗРАБОТКИ ASIC ПОД АЛГОРИТМЫ CRYPTONIGHT И ETHASH И ЧЕМ ГРОЗИТ РЕАЛИЗАЦИЯ ПОДОБНЫХ УСТРОЙСТВ

Р. И. Бобков

Студент

С. А. Зуев

Доцент

Введение

CryptoNight – алгоритм proof-of-work (доказательство через “работу”). Изначально разрабатывался как алгоритм, ориентированный на процессоры персональных компьютеров. Сделано это было с целью полной децентрализации сетей на данном алгоритме. Однако со временем были созданы более совершенные устройства – видеокарты и с недавнего времени ASIC-устройства известных компаний о которых речь пойдет далее. Первой монетой на этом алгоритме стал Bytecoin. Самые популярные монеты на данный момент – Monero (~1.01 GH/s) и Electroneum (~300 MH/s) [1]. Все транзакции в сети полностью анонимны. На данный момент нет сведений о компрометации алгоритма.

Ethash – алгоритм proof-of-work (доказательство через “работу”) [2]. Изначально разработан для Ethereum 1.0. Также называется Dagger Hashimoto (на самом деле Ethash и Dagger Hashimoto это не одно и то же, но это неважно в рамках этой статьи). Алгоритм разрабатывался для сетей, поддержка которых осуществляется с помощью видеокарт с большим объемом, так как необходимо размещать DAG-файл [3] и быстрой памятью. Так как подходящие видеокарты приобрести может практически каждый человек на Земле, то как результат – сильная децентрализация сети. Первой монетой на этом алгоритме стал Ethereum. Самые популярные монеты на этом алгоритме на данный момент – Ethereum (~ 230.5 TH/s) и Ethereum Classic (9.5 TH/s). Также существует большое количество форков (монет на данном алгоритме, похожих своим “ядром” на Ethereum, однако имеющих более лучший функционал или другое предназначение). Главная особенность – создание смарт-контрактов (своего рода микропрограмм, которые, если упрощенно, могут сами распоряжаться финансами на кошельке). Кошельки в сети Ethereum бывают двух видов – те, что управляются людьми, и те, которые управляются программно.

Данного минимального введения или набора знаний достаточно для перехода к обсуждению самих алгоритмов.

1. CryptoNight и почему устойчив к ASIC-устройствам

Так уж сложилось, что на конец марта 2018 уже не все версии алгоритма устойчивы к ASIC-устройствам, но обо всем по порядку.

Алгоритм требует L3 (и более быстрый) кэш современных процессоров. В этом и заключается основная трудность создания ASIC-устройства под данный алгоритм — невозможно создать устройство с большим количеством быстрой памяти дешево. То есть это просто-напросто экономически невыгодно. Однако, некоторые видеокарты позволяют участвовать в поддержке сети, и это зачастую даже более выгодно (соотношение количества решений в секунду к цене устройства) чем использование процессора персонального компьютера. Здесь стоит сказать о том, что современная память видеокарт — GDDR5 медленнее, и имеет плохую скорость произвольного доступа (в отличие от L3 процессора), однако позволяет добиваться хорошей производительности на данном алгоритме благодаря хорошей пропускной способности.

Несмотря на это данный алгоритм все-равно считается децентрализованным, ведь, как уже говорилось выше, приобрести видеокарту и присоединиться к сети может любой желающий.

Что же произошло за последнее время? Несколько компаний практически одновременно выпустили анонсы своих новых устройств:

- Baikal Giant N (20 kH/s, 60W, 3600 USD);
- Bitmain X3 (220kH/s, 550W, 7599-11999 USD);
- PinIdea RR-200 (27kH/s, 350W, N/A USD).

Чем же грозит выпуск таких устройств? Для примера возьмем первое устройство из списка Baikal Giant N и на его примере попробуем произвести некоторые расчеты.

Итак, возьмем вторую по популярности монету на алгоритме CryptoNight – Electroneum.

Мощность сети данной сети, как уже было сказано выше, ~300 MH/s. Таким образом один экземпляр данного ASIC-устройства составляет $20/300020 = \sim 6,67 \cdot 10^{-5}\%$ от мощности сети. Казалось бы - немного, но что если кто-то купит 1000 таких устройств? $20000/320000 = 6,26\%$ сети. Это уже весомый процент, но не сильно пугающий, что ж, давайте возьмем менее популярную монету на данном алгоритме — Sumokoin (16,45 MH/s). 1000 подобных ASIC-устройств одного владельца составит $20000/36450 \sim 54,9\%$, а это значит что сеть централизована. Соответственно такая сеть обречена. И это только третья монета по популярности! Думаю, что исходя из расчетов выше можно сделать вывод, что для таких монет как Karbowanec (3,89 MH/s), Bytecoin (8,02 MH/s), DigitalNote (5,04 MH/s), DinastyCoin (683 kH/s!) и прочих, более мелких

сетей присутствие даже нескольких подобных асиков одного владельца в сети грозит централизацией!

Что же было предпринято?

Команды разработчиков Monero и Electroneum, в срочном порядке, уже объявили о переходе в ближайшее время на алгоритм CryptoNight v7, который имеет резист к вышеназванным ASIC-устройствам. При этом процессоры персональных компьютеров, а так же видеокарты “не пострадают”, то есть продолжают работать в штатном режиме, чего не скажешь об ASIC-устройствах. Хорошая ли это новость? Безусловно! Децентрализация и анонимность – два главных плюса алгоритма CryptoNight. Надолго ли это? Пожалуй, да. Разработчики сети и разработчики ASIC-устройств будут вести своеобразную войну, которая будет выглядеть следующим образом: создается ASIC, меняется алгоритм монетки, ASIC становится бесполезным либо патчится каким-либо образом. И так это может продолжаться очень долго.

В конечном счёте кто-то будет должен сдать, но кто это уже вопрос времени и экономической целесообразности – в первую очередь, разработчиков ASIC-устройств.

2. Ethash и почему устойчив к ASIC-устройствам

Как уже было сказано во введении выше, Ethash это proof-of-work алгоритм, основанный на сети Ethereum. Ethash использует совокупность SHA-3 хеширование, которое известно также как Keccak. Ethash имеет ASIC-резист благодаря memory-hardness (которое сложно реализовать на ASIC чипах).

Общий маршрут, который выполняет алгоритм, выглядит следующим образом:

1. Существует начальный блок, который может быть вычислен для каждого блока путем сканирования через заголовки блоков до этой точки.

2. Из начального блока можно вычислить псевдослучайный кеш на 16 МБ. Клиенты хранят кеш.

3. Из кеша мы можем создать набор данных объемом 1 ГБ, с тем свойством, что каждый элемент в наборе данных зависит от небольшого количества элементов из кеша. Клиенты и майнеры хранят набор данных. Набор данных растет линейно со временем.

4. Добыча включает в себя захват случайных фрагментов набора данных и объединение их вместе. Проверка может выполняться с низкой памятью, используя кеш для регенерации определенных частей набора данных, которые нам нужны, поэтому нужно хранить только кеш.

Большой набор данных обновляется один раз каждые 30000 блоков, поэтому подавляющее большинство майнеров будет читать набор данных, а не вносить в него изменения [4].

Некоторое время назад компания Bitmain анонсировала новое ASIC-устройство под кодовым названием Bitmain F3. Согласно информации с сайта technews.cn, новое устройство будет иметь следующие характеристики:

- Микросхемы фирмы TSMC, клиенты которой, кстати, являются такие компании-производители видеокарт, как AMD и NVIDIA.
- Три платы на борту.
- Каждая плата содержит 6 процессоров.
- Каждый процессор содержит 32 чипа памяти по 1 GB DDR3.
- Но также Technews сообщает о общей памяти устройства равной 72 GB DDR3 (как так? – не совсем понятно, ведь $6 \times 32 \times 1 = 192$ GB, а не 72 GB. В общем и целом – вопрос остается открытым, прототипа еще нет, а значит нас могут ждать изменения).

3. Может ли данное устройство угрожать «домашним» и промышленным майнерам, использующим GPU?

Теоретически – да, практически – не совсем, и вот пара фактов для размышления.

Можно ли считать DDR3 память быстрой и современной? Определенно - нет и вот почему: современные видеокарты имеют на своем борту память GDDR5 или GDDR5X. А это значит, что вполне справедливо будет сравнить DDR3 и GDDR5, чтобы понять производительность устройств с данными типами памяти. На данный момент на рынке есть два типа карт Nvidia GT 1030 – с памятью DDR3 и GDDR5. Ниже, в табл. 1, приведено краткое сравнение по основным характеристикам [5].

Таблица 1

Сравнение производительности разных типов памяти

Карта и тип памяти	Частота, МГц	Пропускная способность	Разница в производительности
GT 730 GDDR5	5000	80Gb/s	~2.5 раза быстрее
GT 730 DDR3	1800	32Gb/s	1

Но это все цифры, а чтобы было более наглядно – проще всего протестировать карты, используя их по прямому назначению, то есть прогнать в игровых бенчмарках.

Результаты тестирования приведены в табл. 2 [5].

Таблица 2

Сравнение производительности разных типов памяти в игровых бенчмарках

Игра	GT 730 DDR3 AVG FPS	GT730 GDDR5 AVG FPS
Battlefield 4	26,3	43
Dota 2	40	68,3
GTA 5	28,7	53
3DMark 11	477 Score points	684 Score points
3DMark Sky Driver	4151 Score points	5782 Score points
3DMark Fire Strike	1005 Score points	1584 Score points

Исходя из результатов в табл. 2 можно сделать вывод, что производительность памяти действительно имеет большую роль при вычислительных операциях. Как мы можем видеть DDR3 память сильно отстает от GDDR5. Не удивительно – DDR3 существует на рынке уже более 10 лет.

Майнинг Ethereum завязан на быстрых вычислительных операциях, сможет ли DDR3 обеспечить конкуренцию GDDR5? Нет, только если Bitmain не придумает что-то крутое в плане оптимизации.

Ну и самое главное — как на это смотрит Ethereum-сообщество и сам Виталик Бутерин? 20.07.2016 был проведен DAO-hardfork, разделивший сеть эфириума на две цепочки [6]. Это была вынужденная мера, так как из-за ошибки в смарт-контракте DAO потеряла огромное количество Ethereum, на тот момент это был эквивалент 100 млн долларов. То есть из-за случившегося конфуза целую сеть “откатили”. Сомневаюсь, что Ethereum-сообщество не предпримет меры по устранению ASIC-устройств из сети путем softfork или hardfork. С другой стороны — все ближе и ближе переход сети Ethereum на proof-of-stake, а это значит что “майнить” в привычном для нас виде будет невозможно. Понимает ли это компания Bitmain? Думаем, что понимает, так что возможно анонс ASIC ни что иное, как простая манипуляция — курсом, волнением майнеров и трейдеров. В заключении обсуждения ASIC под Ethash хочется так же сказать, что выход такого ASIC-устройства если и не убьет майнинг на видеокартах AMD, то точно сильно повлияет на него, а это значит что значительная

часть видеокарт выльется на вторичный рынок: карт в избытке, цена падает и компания AMD несет убытки. Допустит ли это AMD? Конечно же нет, от этого зависит её благополучие.

Заключение

Хорошо ли это – создание ASIC-устройств? И хорошо и плохо: Хорошо, потому что добиваемся большей вычислительной мощности, при меньшем энергопотреблении. Плохо потому что ASIC-устройств производится столько, что они просто вытесняют видеокарты из сети, и на самом-то деле экономии электроэнергии никакой нет, так еще и сеть становится централизованнее.

Правильно ли разработчики алгоритмов делают, внося изменения в алгоритм, чтобы сделать его более защищенным от ASIC-устройств. По нашему мнению, да! Как уже было сказано выше — сохраняем таким образом “домашний майнинг”, а значит и сохраняем сильную децентрализованность сети. Ведь в большинстве случаев на ASIC-устройствах зарабатывают только разработчики данных ASIC-устройств, а не их владельцы, ведь как мы все помним — в бум золотой лихорадки больше всех заработали именно продавцы инструмента, а не сами добытчики.

Конечно, производители ASIC устройств так просто свою нишу не отдадут, и мы будем ещё свидетелями этих битв между создателями алгоритмов и программ с одной стороны и производителями ASIC устройств с другой. Кто победит? На наш взгляд победят правительства промышленно развитых стран, договорившись между собой. Которые пойдут путем введения регулировки рынка криптовалют и тем самым положат конец такой борьбе.

Список литературы

1. Whattomine [Электронный ресурс]. – Режим доступа : <http://whattomine.com>
2. BitcoinWiki.org. Proof Of Work [Электронный ресурс]. – Режим доступа : <https://ru.bitcoinwiki.org/wiki/Proof-Of-Work>
3. Thulasiraman, K. Swamy Acyclic Directed Graphs / K. Thulasiraman // London : Pearson Education Inc., 1998. – 118 p.
4. 2bitcoins.ru 2 Биткоина [Электронный ресурс]. – Режим доступа : <https://2bitcoins.ru>
5. VideoCardBenchmark.net [Электронный ресурс]. – Режим доступа : <https://www.videocardbenchmark.net>
6. Wikipedia.org. The DAO [Электронный ресурс]. – Режим доступа : [https://en.wikipedia.org/wiki/The_DAO_\(organization\)](https://en.wikipedia.org/wiki/The_DAO_(organization))

АНАЛИЗ КОРПОРАТИВНЫХ ДАННЫХ НА ОСНОВЕ ТЕХНОЛОГИЙ DATA MINING В СРЕДЕ RSTUDIO

Н. А. Бойкова

Студент

В. В. Гаршина

Доцент

Введение

Современное информационное общество привело к пониманию важности задач, связанных с анализом накопленной информации для извлечения новых знаний. Появилась технология по обнаружению и извлечению новых знаний — Data Mining. Data Mining (также называемая Knowledge Discovery In Data) — исследование и обнаружение «машиной» (алгоритмами, средствами искусственного интеллекта) в сырых данных скрытых знаний, которые ранее не были известны, нетривиальны, практически полезны, доступны для интерпретации человеком [1].

Пристальное внимание к технологиям Data Mining, в первую очередь, стали уделять коммерческие организации, имея потребность разобраться в накопленных данных, чтобы максимально увеличить свою прибыль. Создаются аналитические отделы для выполнения данной функции, которым необходимо проводить анализ при помощи различных статистических пакетов. В данной работе будет рассмотрена методика проведения анализа данных на основе технологий Data Mining в среде RStudio.

1. Анализ корпоративных данных на основе стандарта CRISP

Приступая к анализу данных, необходимо учесть стандарты, которые затрагивают основные аспекты Data Mining. Они предъявляют требования не только к унификации интерфейсов, посредством которых любое приложение может получить доступ к функциональности Data Mining, а также выработке рекомендаций по организации процесса Data Mining в целом.

Существующий стандарт CRISP, призван удовлетворить потребности в методологии организации проведения процесса Data Mining. CRISP-DM (CRoss-Industry Standard Process for Data Mining) — непатентованная, документированная и свободно доступная модель, описывающая основные фазы, выполнение которых позволяет организациям получать максимальную выгоду от использования методов Data Mining. В настоящее время ведется работа над версией 2.0 данного стандарта [2].

На представленном рис. 1 описывается стандарт CRISP-DM в терминах иерархической модели процесса. Модель состоит из набора задач, описанных на четырех уровнях абстракции (от более общего к более конкретному): фазы, общие задачи, специализированные задачи и примеры процессов. Каждая фаза верхнего уровня включает в себя несколько общих задач, относящихся ко второму уровню иерархии.

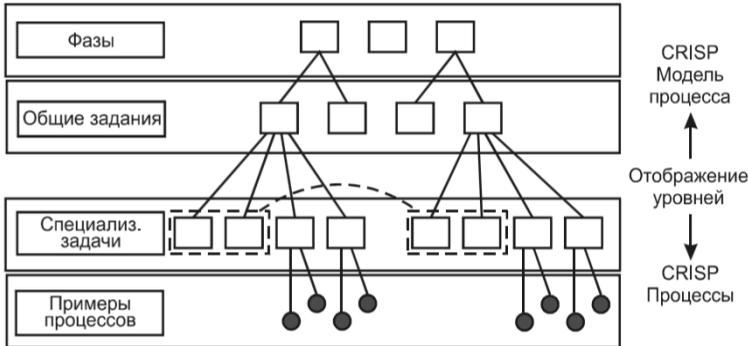


Рис. 1. Четыре уровня методологии CRISP

Рассмотрим более подробно два верхних уровня модели: фазы проекта Data Mining и общие задачи каждой из фаз. CRISP-DM делит жизненный цикл проекта Data Mining на следующие шесть фаз: понимание бизнес-процессов (business understanding); понимание данных (data understanding); подготовка данных (data preparation); моделирование (modeling); оценка (evaluation); размещение (deployment).

Перечисленные фазы и взаимоотношения между ними представлены на рис. 2. Стрелками изображены более важные и частые зависимости между фазами. Внешние стрелки, имеющие циклическую природу, иллюстрируют спиралеобразный процесс разработки проектов Data Mining. Другими словами, после фазы размещения может возникнуть необходимость в новом переосмыслении бизнес-процессов и повторения всех шести фаз сначала.

Фаза понимания бизнес-процессов — возможно, наиболее важная фаза в проекте Data Mining. Чтобы понять, какие данные и как в дальнейшем они должны быть проанализированы, важным является полностью понять бизнес, для которого происходит поиск решения. На данной фазе решаются следующие задачи: определение бизнес-целей; определение ситуации; определение целей Data Mining; создание плана проекта.



Рис. 2. Жизненный цикл процесса Data Mining согласно методологии CRISP

Фаза понимания данных — начинается с первоначального сбора данных. Затем происходит более близкое знакомство с ними с целью идентифицировать проблемы качества данных, исследовать их суть и выявить интересные поднаборы для формирования гипотез о скрытых знаниях. В этой фазе выполняются четыре общие задачи: первичный сбор данных; описание данных; изучение данных; проверка качества данных.

Фаза подготовки данных — третья фаза, включающая в себя все действия, связанные с окончательным формированием набора данных для анализа. При этом выполняются пять задач: выбор данных; очистка данных; конструирование данных; интеграция данных; форматирование данных.

Фаза моделирования — предназначена для выбора оптимального метода построения моделей и настройки его параметров для получения оптимальных решений: выбор метода моделирования; генерация тестового проекта; создание моделей; оценка моделей.

Фаза оценки — призвана более основательно оценить модель до процесса ее окончательного размещения, чтобы убедиться в достижимости поставленных бизнес-целей. Эта фаза включает следующие задачи: оценка результатов; пересмотр процесса; определение дальнейших действий [1, 2].

2. Описание данных для анализа методами Data Mining

Следуя стандарту CRISP, после фазы понимания бизнес-процессов компании, следует понять и подготовить для анализа набор данных. Набор данных — это, как правило, прямоугольный массив данных, в котором ряды соответствуют наблюдениям, а столбцы — признакам.

Таблица данных (data frame) – это более широко используемый по сравнению с матрицей объект, поскольку разные столбцы могут содержать разные типы данных (числовой, текстовый и т.д.). Набор данных про поставщиков состоит из числовых и текстовых данных. Эти данные нужно представить в виде таблицы данных, поскольку здесь есть данные разных типов [3].

Проанализировав имеющийся отчет, который содержит 211 признаков и 9 063 наблюдений было решено не использовать при анализе все 211 признаков, которые получились при выгрузке данных. Так как большее число признаков, вряд ли позволит нам сделать на их основе какой-либо анализ. Итак, целесообразнее проводить анализ данных в среде R по признакам, представленным в таблице ниже.

Таблица

Используемые для анализа признаки

№ п/п	Признаки	Поле Excel	Тип данных	Класс данных
1	Первичная потребность	Заявка подр.	char	Категориальные
2	Закупщик	Гр.зак.	char	Категориальные
3	Заказчик	Клиент: имя	char	Категориальные
4	Предмет закупки	Материал Имя	char	Категориальные
5	Объем закупки в БЕИ	Кол-во в Заказах, БЕИ	numeric	Интервальные
6	Плановая стоимость предмета закупки	План.цена без НДС	numeric	Интервальные
8	Заявка на закупку	Заявка на закупку	char	Категориальные
9	Позиция заявки на закупку	Позиция заявки на закупку	char	Категориальные
10	Договор закупки	№ договора при регистрации	char	Категориальные
11	Поставщик	Поставщик имя	char	Категориальные
12	Стоимость закупки	Сумма заказов с НДС	numeric	Интервальные
13	Срок поставки в Первичной потребности	Срок поставки	date	Дата

№ п/п	Признаки	Поле Excel	Тип данных	Класс данных
15	Назначение поставки	СПП имя	char	Категориальные
16	Подразделение Заказчика	СПП отв. имя	char	Категориальные
17	Фактическая дата поставки	Дата последней поставки (21)	date	Дата
18	Класс (группа) предмета закупки	Код класса МТР	char	Категориальные
19	Соблюдение срока поставки	Расчетное	numeric	Интервальные
20	Соблюдение срока поставки	Расчетное2	char	Категориальные

3. Подготовка и анализ данных в среде RStudio

После описания и проверки данных переходим к фазе подготовки данных. При выполнении различных команд основная цель отформатировать набор данных, привести его к окончательному виду, чтобы в дальнейшем построить качественный анализ.

Для начала загружаем базу данных в R командой View и просматриваем содержимое файла командой str (листинг 1). Далее определяем количество пропущенных значений, выполнив команду sum(is.na(sas11_12_05)). Возникает необходимость удалить строки с пропущенными значениями и сохраняем результат в файл (sas11_12_05.nopain <- na.omit(sas11_12_05)).

Листинг 1

Очистка данных

```
View(sas11_12_05)
str(sas11_12_05)
sum(is.na(sas11_12_05))
sas11_12_05.nopain <- na.omit(sas11_12_05)
sum(is.na(sas11_12_05.nopain))
str(sas11_12_05.nopain)
```

Так как мы работаем в среде R с таблицей данных, далее создаем объект типа data.frame (листинг 2). Имея дело с разными типами данных в таблице, следует определить следующие поля как номинальные категориальные типа factor.

Определение класса данных

```

mydata <- data.frame(sas11_12_05.norain)
str(mydata)
mydata$Заявка.подр. <- as.factor(mydata$Заявка.подр.)
mydata$Гр.зак. <- as.factor(mydata$Гр.зак.)
mydata$Клиент..имя <- as.factor(mydata$Клиент..имя)
mydata$Материал.Имя <- as.factor(mydata$Материал.Имя)
mydata$X..договора.при.регистрации <-
as.factor(mydata$X..договора.при.регистрации)
mydata$Приложение <- as.factor(mydata$Приложение)
mydata$Поставщик.имя <- as.factor(mydata$Поставщик.имя)
mydata$СПП.имя <- as.factor(mydata$СПП.имя)
mydata$Материал <- as.factor(mydata$Материал)
mydata$Заявка.на.закупку <-
as.factor(mydata$Заявка.на.закупку)
mydata$Позиц. <- as.factor(mydata$Позиц.)
mydata$Поставщик <- as.factor(mydata$Поставщик)
mydata$СПП.отв..имя <- as.factor(mydata$СПП.отв..имя)
mydata$Код.службы.ГС <- as.factor(mydata$Код.службы.ГС)
mydata$Код.класса.МТР <- as.factor(mydata$Код.класса.МТР)
mydata$Наим.Код.кл. <- as.factor(mydata$Наим.Код.кл.)
mydata$Расчетное2 <- as.factor(mydata$Расчетное2)

```

После выполнения всех этих команд необходимо снова просмотреть результаты командой `str(mydata)` и вывести описательную статистику командой `summary(mydata)`.

Полученный набор данных (`sas11_12_05.norain`), является исходным для дальнейшего анализа. По сравнению с исходным набором — удалены строки, содержащие отсутствующие (пропущенные) значения (признаки). Текстовые данные (`string`) переопределены как факторные (`factor`).

Один из важных вопросов при заключении договоров на поставку товара, является соблюдение сроков поставки.

Согласно графику на рис. 3 было обнаружено, что в основном средняя просрочка поставки товара составляет от 10 до 20 дней, в тоже время обнаружены «выбросы» нарушений поставки до 200 дней. Общее количество нарушений по поставщикам составляет до 7 случаев. Помимо «Дней просрочено» и дат, по которым определяется просрочка, наибольшую значимость составляют «Класс материала» и «Поставщик». Далее в основном будем работать с независимыми переменными Дата поставки, Класс материала, Поставщик и зависимой переменной Дней просрочено.

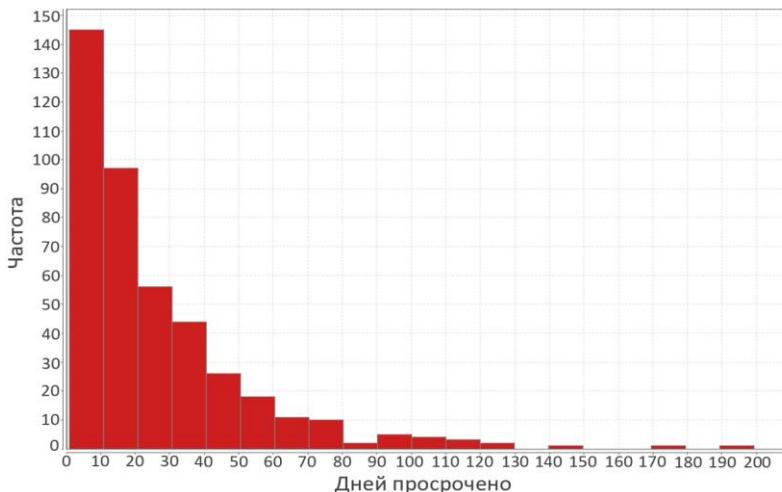


Рис. 3. Нарушения поставок

Заключение

В данной статье были рассмотрены фазы жизненного цикла проекта при анализе набора данных методами Data Mining, на основе стандарта CRISP. Также была рассмотрена и описана база данных, на основе которой будет выполнен анализ в среде R Studio. Показан один из вариантов подготовки таблицы данных при помощи команд в среде R. В итоге проделанной работы мы имеем очищенную от пропущенных значений базу данных, на основе которой был получен предварительный анализ поставщиков, несоблюдающих сроки поставок.

Список литературы

1. Барсегян, А. А. Анализ данных и процессов : учеб. пособие / А.А. Барсегян, М.С. Куприянов, С.И. Елизаров. – 3-е изд., перераб и доп. – СПб. : БХВ-Петербург, 2009. – 512 с.
2. Data Mining: Информация [Электронный ресурс] : статья – Режим доступа : <https://www.intuit.ru/studies/courses/6/6/lecture/198?page=3/>
3. Кабаков, Р.И. R в действии. Анализ и визуализация данных в программе R / Р.И. Кабаков; пер. с англ. П.А. Волковой. – М. : ДМК Пресс, 2014. – 588 с.

РАЗРАБОТКА ПРИЛОЖЕНИЯ ДЛЯ РАЗМЕТКИ КОНТУРОВ ЛИЦ

А. Д. Борисов

Студент

Д. И. Соломатин

Старший преподаватель

Введение

В задачах, решаемых с помощью машинного обучения, на конечный результат, помимо выбранной модели обучения, огромное влияние оказывает качество обучающей выборки. Довольно типичной является ситуация, когда для каких-то задач или классов задач успешные модели для обучения уже известны и основная трудоемкость состоит именно в формировании обучающей выборки приемлемого размера и качества.

Обучающая выборка для тренировки моделей автоматического распознавания контуров лиц на изображениях представляет собой набор размеченных фотографий, на которых требуемые контуры выделены (обведены). В данной работе рассматривается задача создания удобного и эффективного инструмента для осуществления такой разметки.

1. Существующие решения

Необходимость создания собственного размеченного набора данных для обучения моделей и других задач, так называемого датасета (от англ. data set), возникает потому, что существующие в открытом доступе датасеты имеют ряд недостатков:

- недостаточное количество точек для аппроксимации естественного контура искусственным, например, датасет Helen [1] с разметкой iBug [2] (Рис. 1а);

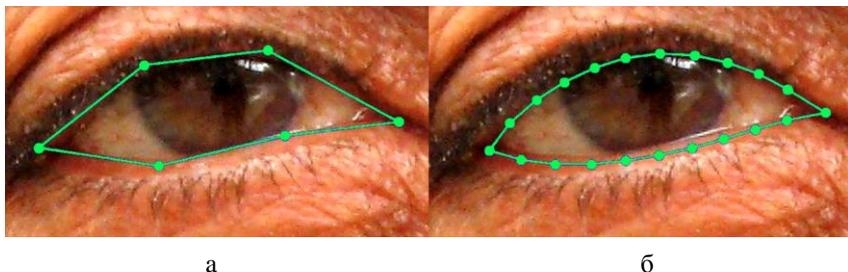
- плохое качество разметки, как в том же датасете Helen [1] (рис. 1б).

Даже если удастся найти датасет с необходимым качеством разметки и количеством точек, то все равно необходимо иметь инструмент для возможности добавления примеров в такой датасет.

Для разметки изображений можно использовать существующие инструменты. Стоит упомянуть, что процесс создания собственного размеченного набора изображений является крайне трудозатратным, поэтому приложения для разметки должны предоставлять не только возможность максимально точно осуществлять разметку, но и делать это максимально быстро и удобно. Найденные приложения для разметки изображений, в частности наиболее популярные программы: web-версия

imgLab [3] и FLAT [4], такими характеристиками не обладают. Среди их недостатков можно выделить:

- недостаточная точность по причине аппроксимации ломаными линиями;
- невозможность отображения контура в момент разметки;
- отсутствие быстрого управления разметкой;
- сложности при разметке изображений с плохим качеством.



*Рис. 1. Недостатки существующих датасетов:
а – нехватка точек, б – плохое качество разметки*

2. Требования к разрабатываемому приложению

Целью данной работы является разработка приложения для разметки контуров лиц, которое будет удовлетворять следующим функциональным требованиям:

- возможность осуществлять разметку контура высокого качества (относительно существующих средств);
- удобство редактирования уже существующей разметки, т.е. добавление, удаление и перемещение ключевых точек контуров;
- поддержка цветокоррекции изображений при разметке любительских фотографий, которые бывают осветленными, затемненными и так далее;
- наличие средств управления процессом разметки: идентификация размеченных и неразмеченных объектов, поиск по изображениям, возможность быстрой проверки существующей разметки и т.п.

3. Реализация

Существует несколько вариантов разметки контура: ломаные линии (рис. 2а) и кубические сплайны (рис. 2б). На рис. 2а можно заметить, что ломаные линии достаточно неплохо описывают контур, однако, для достижения большей точности необходимо использовать большое число точек. На рис. 2б осуществлена разметка того же глаза кубическими сплайнами, которые описывают контур лучше ломаных линий, при этом

используется в среднем в 2.5 раза меньше точек. Этот фактор является очень важным с учетом трудозатрат, необходимых для создания датасета.

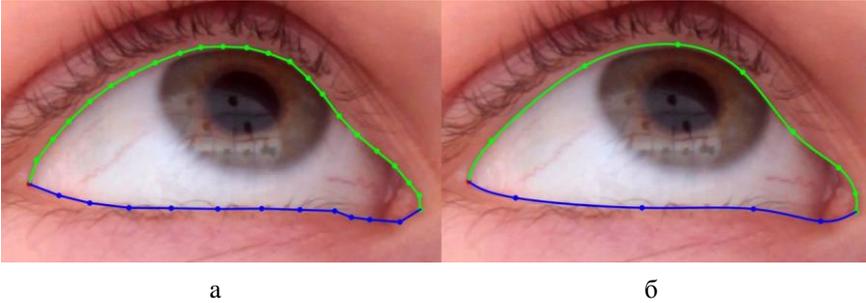


Рис. 2. Варианты разметки контура: а – ломаные линии, б – кубические сплайны

На основании данного сравнения было принято решение использовать кубические сплайны для разметки контуров лиц.

Большую часть времени человек, осуществляющей разметку, будет расставлять точки контура, поэтому этот процесс должен быть максимально понятным и быстрым. Самый простой способ добавить точку к контуру – это указать курсором, где точка должна быть расположена, а контур должен сам подстроиться под новую точку. Этого можно достичь, вычислив ближайшую линию к данной точке. Рассмотрим данную задачу в векторной форме (Рис. 3).

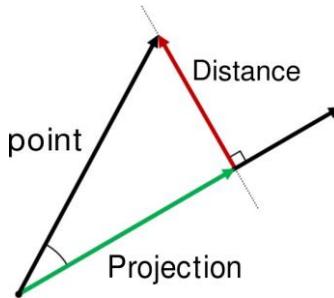


Рис. 3. Нахождение расстояния Евклида

Используя формулу евклидова расстояния, найдем длину отрезка между точкой и линией.

$$proj\ddot{e}ction = \frac{(point * line) * line}{(line * line)}; \tag{1}$$

$$distance = \sqrt{(point.x - projection.x)^2 + (point.y - projection.y)^2}. \quad (2)$$

Данные вычисления (1) и (2) необходимо провести для каждой линии контура, а затем выбрать ближайшую и разбить ее новой точкой.

При осуществлении разметки изображений с различными дефектами, из-за которых истинный контур может быть недостаточно хорошо виден, необходимо иметь возможность редактировать контрастность и яркость изображений. На рис. 4а приведен пример такого изображения, на котором видно, что истинный контур губ довольно нечеткий и для того, чтобы правильно разметить его, необходимо выполнить цветокоррекцию (рис. 4б), что позволяет однозначно идентифицировать естественный контур.



Рис. 4. Пример цветокоррекции: *а* – исходное изображение, *б* – отредактированное изображение

В качестве одной из функций управления процессом разметки реализована функция `save scale`, которая сохраняет фокус и масштаб над сегментом при переключении изображений, что значительно повышает скорость проверки и работы с уже выполненной разметкой (рис. 5).

Итоговая разметка сохраняется в виде набора точек в файл JSON формата. Данный формат выбран по причине его популярности и возможности считывания большим количеством фреймворков.

Приложение реализовано на языке C++ с использованием кроссплатформенного фреймворка Qt. На рис. 5 показан интерфейс созданного приложения.

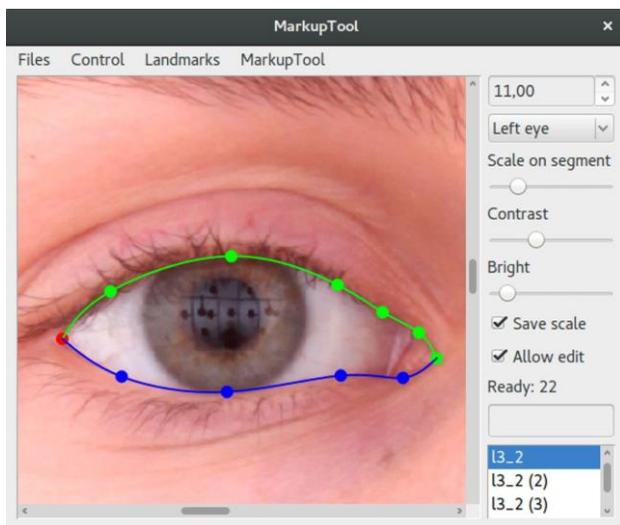


Рис. 5. Интерфейс реализованного приложения

Заключение

В отличие от существующих аналогов в разработанном приложении построение контуров осуществляется с помощью кубических сплайнов, что обеспечивает скорость разметки при сохранении высокой точности. При этом приложение имеет прозрачный интерфейс и необходимый набор функций для удобства работы.

В данный момент приложение активно используется при подготовке данных для обучения нейронной сети в рамках задачи детектирования контуров лиц в Воронежской студии Russian3DScanner.

Список литературы

1. Helen dataset [Электронный ресурс] : документация. – Режим доступа : <http://www.ifp.illinois.edu/~vuongle2/helen/>
2. Facial point annotations [Электронный ресурс]: научный журнал. – Режим доступа : <https://ibug.doc.ic.ac.uk/resources/facial-point-annotations/>
3. ImgLab [Электронный ресурс] : онлайн сервис. – Режим доступа : <https://naturalintelligence.github.io/imglab/>
4. Flat [Электронный ресурс] : онлайн сервис. – Режим доступа : <https://github.com/luigivieira/Facial-Landmarks-Annotation-Tool>

ПРИМЕНЕНИЕ СВЕРТОЧНЫХ НЕЙРОННЫХ СЕТЕЙ ДЛЯ РЕШЕНИЯ ЗАДАЧИ РАСПОЗНАВАНИЯ ИЗОБРАЖЕНИЙ

А. В. Бородин

Студент

А. А. Крыловецкий

Доцент

Введение

С давних времен люди мечтали создать думающую машину. В наше время искусственный интеллект (ИИ) – бурно развивающаяся отрасль науки, применяемая для решения самых разнообразных задач. В их число входят автоматизация рутинного труда, интеллектуальный анализ данных, распознавание речи, текста, изображений и так далее.

На заре становления науки об искусственном интеллекте были быстро исследованы и решены некоторые задачи, относительно простые для компьютеров – описываемые с помощью списка формальных математических правил. Настоящей проблемой для искусственного интеллекта стали задачи, которые легко решаются человеком, но с трудом поддаются формализации, – задачи, которые мы решаем интуитивно, как бы автоматически: распознавание устной речи или лиц на картинке [1].

Решение таких интуитивных задач требует особого подхода. По сути, требуется дать компьютеру возможность получать знания опытным путем. Другими словами, компьютер должен учиться сложным понятиям на основании более простых, то есть иметь иерархическую организацию понятий. Очевидно, что граф, описывающий такую иерархическую структуру, будет довольно глубоким (многоуровневым). Именно поэтому данный подход к ИИ был назван глубоким обучением.

В данной работе рассматривается задача распознавания фруктов на изображениях. Исследование включает в себя разработку архитектуры нейронной сети, а также нахождение её оптимальных параметров, позволяющих максимизировать точность классификации.

1. Сверточные нейронные сети

В области распознавания и классификации образов наиболее успешные результаты были получены с помощью искусственных нейронных сетей. Это послужило важной причиной для выбора глубокой нейронной сети в качестве решения задачи идентификации фруктов на изображениях. Глубокие нейронные сети сумели превзойти другие алгоритмы машинного обучения. Они первыми достигли превосходства

над человеком в определенных областях распознавания образов [2]. Это еще раз подтверждает тот факт, что глубокое обучение рассматривается как важный шаг к получению сильного ИИ.

Существует множество видов глубоких нейронных сетей со своими уникальными подходами к обучению. Так, в рекурсивных сетях нейроны с одинаковыми весами активируются рекурсивно в соответствии со структурой сети. Рекуррентные нейронные сети имеют возможность использовать свою внутреннюю память для обработки последовательностей произвольной длины, что сделало их успешными в сфере построения языковых моделей. Глубокая сеть доверия может использоваться в качестве предварительной подготовки для глубокой нейронной сети (происходит улучшение начальных значений весов). В свою очередь, сверточные нейронные сети достигли наилучших результатов в области распознавания изображений.

Действительно, классические полносвязные нейронные сети имеют ряд недостатков по сравнению со сверточными сетями. Во-первых, полносвязные сети требуют большое количество весов для обучения. Во-вторых, изображения в них представляются в виде плоского массива – теряется информация о топологии.

Сверточная нейронная сеть имеет несколько основополагающих принципов: локального восприятия, разделяемых весов и уменьшения размерности. Далее рассмотрим их немного подробнее.

Локальное восприятие подразумевает, что на вход одного нейрона подается не все изображение (или выходы предыдущего слоя), а лишь некоторая его область. Данный подход решает ранее упомянутую проблему сохранения информации о топологии изображения.

В основе большинства систем распознавания изображений лежит множество разных фильтров. В сверточной нейронной сети слой свёртки умножает значения фильтра на исходные значения пикселей изображения (поэлементное умножение), после чего все эти умножения суммируются. Каждая уникальная позиция введенного изображения производит число.

Концепция разделяемых весов предполагает, что для большого количества связей используется небольшой набор весов. Т.е. если у нас имеется на входе некоторое изображение, то каждый из нейронов следующего слоя примет на вход только небольшой участок этого изображения, причем каждый из фрагментов будет обработан одним и тем же набором. Важно понимать, что самих наборов весов может быть много, но каждый из них будет применен ко всему изображению. Такие наборы часто называют ядрами (kernels). В сверточных нейронных сетях ядра свертки определяются автоматически в процессе обучения.

Суть принципа уменьшения размерности, как легко догадаться, состоит в уменьшении изображения в заданное количество раз. Это позволяет распознавать объекты вне зависимости от масштаба изображения. Также данная парадигма делает факт наличия признака важнее места его точного положения на изображении.

2. Набор данных и используемые библиотеки

Известно, что качество обучающей выборки серьезно влияет на результаты работы нейронных сетей. Другими словами, наличие объёмной базы данных не гарантирует высокой точности распознавания.



Слева: исходное изображение, справа: после удаления фона и уменьшения размера

Рис. 1. Изображение фрукта

В качестве набора данных был выбран [3]. Приведенный набор данных, используемый для обучения, состоит из 38409 изображений, распределенных по 60 наименованиям фруктов. Особенностью данного набора является то, что фон на изображениях с фруктами был удален

(пример на рисунке 1). Это решает проблему зашумленности изображений, и, соответственно, положительно влияет на точность распознавания [4].

Программная реализация выполнена на языке Python. Для работы с сетями глубокого обучения была выбрана открытая нейросетевая библиотека Keras. В качестве вычислительного бэкенда использовалась библиотека Tensorflow.

3. Результаты

Чтобы лучше понимать принцип работы свёрточных сетей, важно изучить их общую архитектуру. При проектировании свёрточных сетей в архитектуру могут добавляться разнообразные слои для повышения точности распознавания (dropout layer, batch normalization layer и т.д.).

В процессе работы проводились эксперименты по обучению и тестированию сетей с различной архитектурой. В первую очередь в сеть загружается изображение. Далее начинается итерация обучения – изображение проходит через все слои по порядку, а получившийся на выходе прогноз сравнивается с истинным значением с помощью функции потерь (loss function). С каждой следующей итерацией веса сети в процессе обучения постепенно подгоняются к оптимальному значению, необходимому для точного решения поставленной задачи.

После завершения всего цикла обучения сети, начинается тестирование для оценки точности распознавания. Тестирование нужно проводить на данных, которые не были в тренировочной выборке.

Отметим, что алгоритм работы нейронной сети является итеративным, а его шаги называют эпохами или циклами. Эпоха – это одна итерация в процессе обучения, включающая предъявление всех примеров из обучающего множества и проверку качества обучения на контрольном множестве.

В эксперименте 1 и эксперименте 2 использовались полносвязные нейронные сети, состоящие из 3 и 6 полносвязных слоев соответственно. В эксперименте 3 тестировалась сверточная нейронная сеть, включающая 2 слоя свертки, слой подвыборки MaxPooling (выбор максимального значения) и 2 полносвязных слоя. В эксперименте 4 исследовалась сверточная нейронная сеть, включающая чередующиеся 4 слоя свертки и 4 слоя подвыборки MaxPooling, а также 2 полносвязных слоя в конце. Подробную реализацию модели сети из эксперимента 4 можно изучить в листинге. Точность распознавания фруктов для всех экспериментов представлена на рис. 2.

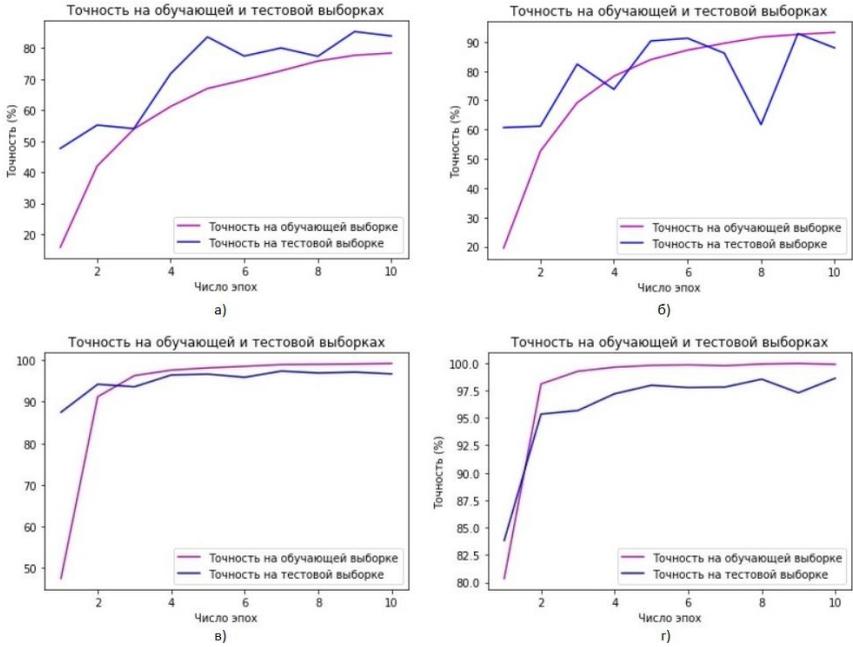


Рис. 2. Сравнение различных конфигураций нейронных сетей:
 а – Эксперимент 1, б – Эксперимент 2, в – Эксперимент 3,
 г – Эксперимент 4

Листинг

Архитектура сверточной нейронной сети в эксперименте 4

```

model = Sequential()
model.add(Conv2D(16, (3, 3), input_shape=(64, 64, 3),
padding='same'))
model.add(PReLU(weights=None, alpha_initializer="zero"))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Conv2D(32, (3, 3), padding='same'))
model.add(PReLU(weights=None, alpha_initializer="zero"))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))
model.add(Conv2D(32, (3, 3), padding='same'))
model.add(PReLU(weights=None, alpha_initializer="zero"))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Conv2D(64, (3, 3), padding='same'))

```

Окончание листинга

```
model.add(PReLU(weights=None, alpha_initializer="zero"))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Flatten())
model.add(Dense(256,activation='elu'))
model.add(Dropout(0.5))
model.add(Dense(60))
model.add(Activation("softmax"))
```

Увеличение количества полносвязных слоев улучшило точность распознавания в эксперименте 2 (87,93%) по сравнению с экспериментом 1 (83,84%). Заметно отличие в результатах полносвязных и сверточных нейронных сетей. Так, разница в точности классификации на тестовом наборе между экспериментами 2 и 3 составила около 9%. Стоит отметить, что увеличение количества слоев свертки позволило уменьшить потери при обучении и тестировании, и, очевидно, повысило процент распознавания в эксперименте 4 по сравнению с экспериментом 3 (98,84% и 96,7% соответственно).

Заключение

Таким образом, сверточные нейронные сети и алгоритмы, реализованные на принципе их функционирования, доказывают свое преимущество в области распознавания изображений. Были найдены параметры сети, обеспечивающие большую скорость обучения и высокий процент распознавания фруктов.

Одной из главных целей будущих исследований является повышение точности нейронной сети. Это предполагает дальнейшие эксперименты со структурой сети. Изменение значений параметров, а также введение новых слоев могут обеспечить совершенно другие результаты.

Другая цель – увеличение количества изображений фруктов, используемых для обучения и тестирования. Это довольно трудоемкий процесс, поскольку заметную пользу принесет добавление тех изображений, которые ранее не использовались.

Список литературы

1. Гудфеллоу Я. Глубокое обучение / Я. Гудфеллоу, И. Бенджио, А. Курвилль. – М. : ДМК Пресс, 2018. – 652 с.
2. Шолле Ф. Глубокое обучение на Python / Ф. Шолле. – СПб. : Питер, 2018. – 400 с.
3. Kaggle [Электронный ресурс] : Fruits 360 dataset. – Режим доступа : <https://www.kaggle.com/moltean/fruits/version/13/data>
4. Mureşan H. Fruit recognition from images using deep learning / H. Mureşan, M. Oltean. – Babes-Bolyai University, 2017. – 13 p.

СООТВЕТСТВИЕ КЛАССИФИКАЦИЙ ГОЛОМОРФНО-ОДНОРОДНЫХ ПОВЕРХНОСТЕЙ В \mathbb{C}^2 И ТРЁХМЕРНЫХ АЛГЕБР ЛИ

В. В. Вишнякова

Студент

А. В. Атанов

Доцент

Введение

Статья посвящена рассмотрению одной достаточно важной задачи комплексной геометрии. В этой области математики особое внимание уделяется изучению геометрических объектов, обладающих свойством симметрии. В частности, важнейшую роль здесь играют поверхности, однородные относительно голоморфных преобразований. Известно, что с каждой голоморфно-однородной поверхностью связана некоторая алгебра Ли (в рассматриваемом здесь случае трёхмерная).

При этом как список всех трёхмерных алгебр, так и список голоморфно-однородных поверхностей в двумерном комплексном пространстве известны. Первая из указанных классификаций была построена Л. Бьянки (см. [1]), вторая – Э. Картаном в работе [2].

Несмотря на то, что обе классификации существуют достаточно давно, явного указания на взаимосвязи между первым («алгебраическим») и вторым («геометрическим») списками не существует. Цель этой работы – установить взаимосвязи между классификациями Картана и Бьянки.

1. Классификация алгебр Ли и аффинно-однородных поверхностей в двумерном комплексном пространстве

Введём сначала определения, которые будут использоваться в дальнейшем.

Определение 1. Поверхность M называется (локально) однородной относительно группы Ли голоморфных преобразований G , если G транзитивно действует на M , то есть любую точку из M можно перевести в любую другую точку поверхности подходящим преобразованием из группы G .

Смысл однородности многообразия – равноправие всех точек этого многообразия и даже их инфинитезимальных окрестностей. Польза свойства однородности заключается в возможности во многих нетривиальных вопросах изучения многообразия ограничиться изучением окрестности только одной точки.

Определение 2. Линейное пространство V , в котором задана кососимметрическая билинейная операция $[\bullet, \bullet]$ (коммутатор), называется алгеброй Ли, если для любых $\xi, \eta, \zeta \in V$ выполняются аксиомы:

- 1) $[\xi, \xi] = 0$;
- 2) $[\xi, [\eta, \zeta]] + [\eta, [\zeta, \xi]] + [\zeta, [\xi, \eta]] = 0$ – тождество Якоби.

Приведём теперь классификацию голоморфно-однородных поверхностей в C^2 .

Теорема (Э. Картан). Любая голоморфно-однородная вещественная поверхность двумерного комплексного пространства голоморфно эквивалентна вблизи произвольной своей точки одной поверхности из следующего списка.

- 1) Гиперплоскость $\text{Im } w = 0$;
- 2) Единичная сфера (квадрика) $|z|^2 + |w|^2 = 1$;
- 3) Поверхности Картана

$$1 + |z|^2 + |w|^2 = a|1 + z^2 + w^2|, a > 1;$$

$$1 + |z|^2 + |w|^2 = a|1 + z^2 - w^2|, a > 1;$$

$$-1 + |z|^2 + |w|^2 = a|-1 + z^2 + w^2|, 0 < |a| < 1;$$

4) Трубки $T = M + iR_y^2$ над аффинно-однородными основаниями, где M – одна из следующих кривых: $y = \ln x$, $y = x \ln x$, $y = x^\alpha$ ($-1 \leq \alpha \leq 1$), $r = e^{\alpha\varphi}$ ($\alpha \geq 1$). Здесь $x = \text{Re } z$, $y = \text{Im } z$, r – полярный радиус, φ – полярный угол.

Теорема (Л. Бьянки). Любая трёхмерная алгебра Ли изоморфна одной (и только одной) алгебре Ли из следующего списка (e_1, e_2, e_3 – базис алгебры Ли):

Тип I. $[e_1, e_2] = 0$, $[e_2, e_3] = 0$, $[e_3, e_1] = 0$.

Тип II. $[e_1, e_2] = 0$, $[e_2, e_3] = e_1$, $[e_3, e_1] = 0$.

Тип III. $[e_1, e_2] = e_2 - e_3$, $[e_2, e_3] = 0$, $[e_3, e_1] = e_2 - e_3$.

Тип IV. $[e_1, e_2] = e_2 + e_3$, $[e_2, e_3] = 0$, $[e_3, e_1] = -e_3$.

Тип V. $[e_1, e_2] = e_2$, $[e_2, e_3] = 0$, $[e_3, e_1] = -e_3$.

Тип VI. $[e_1, e_2] = ae_2 - e_3$, $[e_2, e_3] = 0$, $[e_3, e_1] = e_2 - ae_3$.

Тип VI₀. $[e_1, e_2] = 0$, $[e_2, e_3] = e_1$, $[e_3, e_1] = -e_2$.

Тип VII. $[e_1, e_2] = ae_2 + e_3$, $[e_2, e_3] = 0$, $[e_3, e_1] = e_2 - ae_3$.

Тип VII₀. $[e_1, e_2] = 0$, $[e_2, e_3] = e_1$, $[e_3, e_1] = e_2$.

Тип VIII. $[e_1, e_2] = -e_3$, $[e_2, e_3] = e_1$, $[e_3, e_1] = e_2$.

Тип IX. $[e_1, e_2] = e_3$, $[e_2, e_3] = e_1$, $[e_3, e_1] = e_2$.

2. Техника решения задачи

Как было отмечено, целью данного исследования является установление взаимосвязей между классификацией Картана голоморфно-однородных гиперповерхностей и списком Бьянки вещественных трёхмерных алгебр Ли. Для достижения этой цели была использована техника, использующая алгебры голоморфных векторных полей (см. [3]).

Пусть $g = \langle e_1, e_2, e_3 \rangle$ – трёхмерная алгебра Ли векторных полей

$$e_1 = f_1(z, w) \frac{\partial}{\partial z} + g_1(z, w) \frac{\partial}{\partial w},$$

$$e_2 = f_2(z, w) \frac{\partial}{\partial z} + g_2(z, w) \frac{\partial}{\partial w},$$

$$e_3 = f_3(z, w) \frac{\partial}{\partial z} + g_3(z, w) \frac{\partial}{\partial w},$$

где $f_i(z, w)$, $g_i(z, w)$ ($i = 1, 2, 3$) – голоморфные функции комплексных переменных z, w .

Коммутатор таких полей имеет вид

$$[e_j, e_k] = \left(f_j \frac{\partial f_k}{\partial z} + g_j \frac{\partial f_k}{\partial w} - f_k \frac{\partial f_j}{\partial z} - g_k \frac{\partial f_j}{\partial w} \right) \cdot \frac{\partial}{\partial z} + \left(f_j \frac{\partial g_k}{\partial z} + g_j \frac{\partial g_k}{\partial w} - f_k \frac{\partial g_j}{\partial z} - g_k \frac{\partial g_j}{\partial w} \right) \cdot \frac{\partial}{\partial w}.$$

Дальнейшая работа основывается на использовании следующей известной из теории дифференциальных уравнений теоремы.

Теорема 1. Любое гладкое голоморфное векторное поле можно выпрямить вблизи неособой точки с помощью подходящей голоморфной замены координат, то есть привести поле

$$e_k = f_k(z, w) \frac{\partial}{\partial z} + g_k(z, w) \frac{\partial}{\partial w}$$

к виду

$$e_k^* = \frac{\partial}{\partial w^*}.$$

При этом после выпрямления структура основных полей не меняется. Отметим, что выпрямить таким образом все поля, составляющие базис

алгебры, в данном случае невозможно, так как вещественный ранг алгебры должен оставаться равным трём.

Однако используя теперь выпрямленное поле, мы можем с помощью коммутационных соотношений упрощать и остальные поля.

Помимо рассмотренного основного утверждения, в работе доказана следующая теорема.

Теорема 2. Векторное поле

$$e_k = f_k(z) \frac{\partial}{\partial z} + g_k(z) \frac{\partial}{\partial w},$$

где $f_k(z)$ – не обращающаяся в ноль в некоторой точке p функция, подходящей голоморфной заменой координат можно привести к виду

$$e_k^* = \frac{\partial}{\partial z^*}.$$

Используя этот факт, мы получаем дальнейшее упрощение рассматриваемых векторных полей. Итогом этой части работы стало получение вида векторных полей, соответствующих алгебрам из списка Бьянки.

Тип I: $e_1 = \frac{\partial}{\partial w}, e_2 = \frac{\partial}{\partial z}, e_3 = C_1 \frac{\partial}{\partial z} + C_2 \frac{\partial}{\partial w};$

Тип II: $e_1 = \frac{\partial}{\partial w}, e_2 = \frac{\partial}{\partial z}, e_3 = C_1 \frac{\partial}{\partial z} + (z + C_2) \frac{\partial}{\partial w};$

Тип III: $e_1 = (-z + w + C_1) \frac{\partial}{\partial z} + (z - w + C_2) \frac{\partial}{\partial w}, e_2 = \frac{\partial}{\partial w}, e_3 = \frac{\partial}{\partial z};$

Тип IV: $e_1 = (-z + C_1) \frac{\partial}{\partial z} + (-z - w + C_2) \frac{\partial}{\partial w}, e_2 = \frac{\partial}{\partial z}, e_3 = \frac{\partial}{\partial w};$

Тип V: $e_1 = (-z + C_1) \frac{\partial}{\partial z} + (-w + C_2) \frac{\partial}{\partial w}, e_2 = \frac{\partial}{\partial w}, e_3 = \frac{\partial}{\partial z};$

Тип VI: $e_1 = (-az + w + C_1) \frac{\partial}{\partial z} + (z - aw + C_2) \frac{\partial}{\partial w}, e_2 = \frac{\partial}{\partial w}, e_3 = \frac{\partial}{\partial z};$

Тип VI₀: $e_1 = \frac{\partial}{\partial z}, e_2 = \frac{\partial}{\partial w}, e_3 = (w + C_1) \frac{\partial}{\partial z} + (z + C_2) \frac{\partial}{\partial w};$

Тип VII: $e_1 = (-az - w + C_1) \frac{\partial}{\partial z} + (z - aw + C_2) \frac{\partial}{\partial w}, e_2 = \frac{\partial}{\partial w}, e_3 = \frac{\partial}{\partial z};$

Тип VII₀: $e_1 = \frac{\partial}{\partial z}, e_2 = \frac{\partial}{\partial w}, e_3 = (w + C_1) \frac{\partial}{\partial z} + (-z + C_2) \frac{\partial}{\partial w}.$

Отметим, что в данном списке все векторные поля являются линейными, то есть не просто голоморфными, а аффинными. Два последних типа из списка Бьянки, так как их связь со списком Картана достаточно очевидна (см. [2]).

3. Интегрирование векторных полей

Далее по полученным векторным полям необходимо построить интегральные поверхности.

Факт касания векторным полем e однородной поверхности M записывается в виде основного уравнения

$$\operatorname{Re}\{e(\Phi)\}|_M = 0,$$

где $\Phi = \Phi(z, w) = \Phi(x, y, u, v)$ – вещественнозначная определяющая функция обсуждаемой поверхности. Здесь $x = \operatorname{Re} z$, $y = \operatorname{Im} z$, $u = \operatorname{Re} w$, $v = \operatorname{Im} w$.

Записывая подобные соотношения для всех трёх полей и переходя в получающихся уравнениях от комплексных переменных к вещественным (в том числе и к производным по вещественным переменным), получаем системы дифференциальных уравнений в частных производных. Решая такие системы для каждого случая, мы получаем определяющие функции поверхностей, соответствующих рассматриваемым алгебрам.

После получения интегральных поверхностей мы можем составить полный список соответствия между списком трёхмерных алгебр Ли и голоморфно-однородными поверхностями.

Тип I: плоскость $\operatorname{Im} w = 0$;

Тип II: квадрака $|z|^2 + |w|^2 = 1$;

Тип III: трубка над $y = \ln x$;

Тип IV: трубка над $y = x \cdot \ln x$;

Тип V: плоскость $\operatorname{Im} w = 0$;

Тип VI: трубка над $y = x^\alpha$ ($-1 < \alpha < 1$);

Тип VI₀: трубка над $y = x^\alpha$ ($\alpha = -1$);

Тип VII: трубка над $r = e^{a\varphi}$ ($a \neq 0$);

Тип VII₀: трубка над $r = e^{a\varphi}$ ($a = 0$).

Этап интегрирования векторных полей был реализован в системе символьной математики Maple. Кроме того, для подсчётов коммутаторов и определения некоторых свойств рассматриваемых алгебры использовалась библиотека LieAlgebras (I.M. Anderson, Utah State University).

4. Заключение

Задача о нахождении соответствия между классификациями Бьянки и Картана, таким образом, полностью решена. Были получены в явном виде голоморфные алгебры Ли векторных полей, действующих на рассматриваемых поверхностях. Полученные алгебры были

проинтегрированы, то есть осуществлён переход от векторных полей к соответствующим голоморфно-однородным поверхностям. Процедура интегрирования была реализована в системе символьной математики Maple.

Список литературы

1. Дубровин, Б.А. Современная геометрия : Методы и приложения : [в 3 т.] / Б. А. Дубровин, С. П. Новиков, А. Т. Фоменко. – М. : Эдиториал УРСС, 2001. – Т. 2 : Геометрия и топология многообразий. – 5-е изд., испр. – 2001. – 293 с.
2. Cartan, E. Sur la géométrie pseudoconforme des hypersurfaces de deux variables complexes / E. Cartan // Ann. Math. Pura Appl. – 1933. – V. 11, № 1. – P.17-90.
3. Beloshapka, V. K. Homogeneous Hypersurfaces In C^3 Assotiated With A Model CR-Cubic / V. K. Beloshapka, I. G., Kossovskiy // J. Geom Anal. – 2010. – № 20. – P. 538-564.

ОЦЕНКА ПРОПУСКНОЙ СПОСОБНОСТИ БЕСПРОВОДНОЙ СЕТИ С РЕГУЛЯРНОЙ ТОПОЛОГИЕЙ С ИСПОЛЬЗОВАНИЕМ БУФЕРИЗАЦИИ СООБЩЕНИЙ

С. Р. Войтенко
Магистрант
Ю. Б. Нечаев
Профессор
А. В. Стромов
Ассистент

Введение

В современных телекоммуникационных технологиях, всё более значительную роль начинают играть одноранговые беспроводные сети. Это сети из одинаковых узлов с незначительными (относительно масштабов сети в целом) зонами радиодоступности, характеризующиеся простотой конструкции, малыми размерами и невысокой стоимостью. Передача информации в таких сетях осуществляется посредством множественных ретрансляций между узлами. Одноранговые беспроводные сети пока не получили широкого распространения как конечный продукт, но на разработку алгоритмов их функционирования и устройств, являющихся узлами таких сетей, направлены значительные усилия

научного и инженерного сообщества. Такие сети лежат в основе технологий «умного дома» и сенсорных сетей, способных в масштабе времени, близком к реальному, обеспечить информирование пользователя о состоянии различных параметров естественных и искусственных объектов. Основными задачами, стоящими перед разработчиками одноранговых беспроводных сетей, являются обеспечение маршрутизации информации между узлами и вопросы энергоэффективности, так как в большинстве случаев питание узлов осуществляется от разовых или аккумуляторных батарей.

1. Алгоритм маршрутизации в беспроводной сенсорной сети

Одним из перспективных подходов к решению задачи маршрутизации в одноранговых беспроводных сетях являются алгоритмы, использующие при построении маршрутов координаты узлов сети. Среди них заслуживает внимания идея построения маршрутов по геометрии силовых линий потенциального поля. Модель маршрутизации в одноранговой беспроводной сети как движение информационного потока в потенциальном поле была впервые предложена в [1] и, видимо, независимо этой работы, в [2]. Далее две эти группы авторов развивали данную тему в работах [3-5]. Аналогичный подход, правда, изложенный описательно и без математической модели, приведён в [6]. Привлекательность предложенного метода состоит в возможности его применения к сетям с большим количеством узлов без роста объёма передаваемой сетью служебной информации.

Суть подхода состоит в следующем: при условии стационарности источников и потребителей информации, передачу информации можно рассматривать как векторное поле – информационный поток, а производителей и получателей информации – как источники и стоки соответственно. Силовые линии этого поля представляют собой возможные маршруты передачи информации посредством множественных ретрансляций между ближайшими к силовой линии узлами. Модель позволяет перейти от «микроскопических» характеристик сети (расположение конкретных узлов и пр.) к «макроскопическим», обобщённым характеристикам, не зависящие от конкретного размещения узлов, но определяемым плотностью их размещения и их параметрами.

Дальнейшее развитие рассматриваемый подход к маршрутизации получил в [7], где он был применён к сети находящейся под воздействием источника помех. Далее, в [8], была дана энергетическая трактовка коэффициента в дифференциальном уравнении в частных производных, описывающем маршрутизацию, и предложен метод оценки этого коэффициента, который определяет свойства сети как среды передачи информации. В [9] изложена модификация алгоритма маршрутизации на

основе силовых линий потенциального поля для сетей, компенсирующих влияние единственного источника помех с помощью узлов, оборудованных адаптивными антеннами, и изменения скорости передачи информации в зависимости от отношения сигнал/шум, а также метод моделирования предложенного алгоритма. В [10] подход был расширен на случай нескольких источников помех.

Корректность модели, описываемой в работах [8-10] определяется в том числе правильной оценкой эффективности реализации доступа к разделяемой среде передачи информации (радиоэфир). Это фактически оценка максимальной пропускной способности сети при выбранных параметрах. Если модуль информационного потока в некоторой области превысит эту оценку — результат моделирования будет некорректным, так как сеть просто не сможет обеспечить такую скорость передачи информации. Оценке максимальной пропускной способности сети при многопутевой передаче информации с синхронным и конкурентным доступом к среде, а также в случае буферизации информации в узлах, посвящена эта работа.

2. Рассматриваемая модель сети

Далее будем рассматривать беспроводную сеть с регулярной топологией, при которой устройства располагаются в узлах некоторой сетки, сторона которой в данном случае равна 25. Размер сети – $n \times m$ (6×4 в данном случае), где n – количество узлов «по горизонтали», а m – по «вертикали». Источники информации – узлы в крайнем левом «столбце», а получатели – в крайнем правом. Соответственно, передача осуществляется по параллельным маршрутам. Все узлы работают на одной частоте.

Модель распространения сигнала описывается следующим образом: при передаче сообщения считается, что оно успешно принимается внутри круга радиуса r_1 с центром в передатчике и препятствует приёму другими узлами сообщений (создаёт помехи) в круге радиуса r_2 . В данном случае было принято $r_1=25$, $r_2=30$.

Передача ведётся по кадрам. В каждом кадре один узел передаёт, принимает или ретранслирует только одно сообщение с постоянным объёмом информации.

Наиболее эффективный способ доступа узлов к среде – конкурентный, при котором выход одних узлов в сеть происходит с выдерживанием некоторых пауз, что исключает влияние помех соседних узлов. В таком случае коэффициент пропускной способности для сети с данными параметрами равен $1/4$, то есть на одной линии одно сообщение доходит до получателя раз в четыре кадра. Если же рассматривать конкурентный доступ узлов к среде передачи, при котором выход каждого узла в эфир обуславливается случайным событием, и при этом остальные

узлы, попадающие под зону действия данного узла могут принять сообщение, но не передать его, поскольку в таком случае создаст помеху остальным передающим узлам, получим снижение пропускной способности сети с коэффициентом 0,57. Данный результат был получен как среднее значение по 10 симуляциям в разработанной системе, имитирующей работу беспроводной сенсорной сети [11].

3. Использование буферизации

После анализа полученных результатов возникает следующая задача: поскольку в данной модели используется максимальная загруженность линий (то есть в течение каждого кадра узлы-источники могут отправлять новое сообщение, если получают доступ к радиоэфиру), то пакеты часто «теряются» в результате возникших коллизий, и необходимо как то минимизировать количество потерянных сообщений.

Рассмотрим сеть, в которой каждый узел способен держать в памяти некоторое количество сообщений для последующей передачи (далее — буфер передач). При этом узел так же способен отправлять повторно сообщение, если не «услышал» его ретрансляцию от другого узла, при этом после отправки сообщений они так же заносятся в память (далее — буфер резерва) для некоторого количества повторных отправок. Если узел принял ретрансляцию «своего» сообщения от другого узла, оно удаляется из буфера резерва.

Расчет параметров буферизации можно определить следующим образом: каждый узел-ретранслятор, попадающий в зону видимости других N ретрансляторов, имеет шанс передачи $1/(N+1)$ с учетом того, что эти ретрансляторы так же имеют передаваемое сообщение. Следовательно, одно сообщение может быть отправлено за $N+1$ кадров. Тогда, при заполнении M ячеек буфера передач, самое последнее сообщение может быть передано за $M(N+1)$ кадров. Следовательно, время ожидания сообщения в буфере резерва должно составлять $M(N+1)$ кадров, что практически гарантирует, что следующий ретранслятор не смог передать сообщение дальше по сети. Используя соотношение (1) для расчета количества ячеек буфера резерва k при $M = 10$, $L = 5$, $N = 4$, получим $k = 50$.

$$k = M(N+1)L, \quad (1)$$

где M — количество ячеек буфера передаваемых сообщений; N — количество соседних узлов, конкурирующих за выход в эфир; L — количество повторных отправлений одного сообщения из буфера резерва.

В табл. 1 приведены расчеты количества переданных сообщений в среднем за 10 симуляций без буферизации и с ее использованием за 120 и 300 кадров.

При этом после 120 кадра прекращается генерация новых сообщений в сети и доставляются остаточные. Из нее видно, что процент потерянных

сообщений снизился на 29% в среднем, однако уменьшилось и количество переданных сообщений за 120 кадров на 53%, но к 300 кадрам этот процент снизился до 13%.

Таблица 1

Использование буферизации сообщений

Количество переданных сообщений без буферизации за 120 (300) кадров	Количество переданных сообщений с буферизацией за кол-во кадров		Потеряно без буферизации, %	Потеряно с буферизацией, %
	120	300		
80	43	70	66	37

Для минимизации потерь сообщений введем паузу, которую узел будет выдерживать перед отправкой следующего сообщения для исключения помех. Пусть эта пауза составляет 5 кадров, тогда из табл. 2 видно, что процент переданных сообщений возрос до 90% в среднем по 10 симуляциям. Однако, уменьшилось и общее количество доставленных сообщений на 42%.

Таблица 2

Использование паузы между передачами сообщений

Количество переданных сообщений с ожиданием (300 кадров)	Количество переданных сообщений без ожидания (300 кадров):	Переданных сообщений с ожиданием, %	Переданных сообщений без ожидания, %
41	70	90	66

Заключение

Из полученных результатов можно сделать вывод, что выбор параметров сети зависит от конкретной ситуации и области применения. В дальнейшем целесообразно модернизировать механизм буферизации для увеличения общего количества доставленных сообщений, а также рассчитать коэффициент пропускной способности при различных параметрах сети и с различным расположением устройств в пространстве.

Список литературы

1. Toumpis S. Packetostatics : Deployment of massively dense sensor networks as an electrostatics problem / S. Toumpis, L. Tassiulas // Proc. IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies (Miami, FL, Mar. 2005), Vol. 4. – Miami, 2005. – P. 2290-2301.
2. Kalantari M. Routing in wireless ad hoc networks by analogy to electrostatic theory / M. Kalantari, M. Shayman // Proc. 2004 IEEE International Conference on Communications (Paris, France, June 2004), Vol.7. – Paris, 2005. – P. 4028-4033.
3. Haghpanahi M. Numerical flow optimization in dense wireless sensor networks / M. Haghpanahi, M. Kalantari, M. Shayman // IEEE Globecom 2008 (New Orleans, LA, November 2008). – New Orleans, 2008. – P. 1-6.
4. Kalantari M. A p norm flow optimization problem in dense wireless sensor networks / M. Kalantari, M. Haghpanahi, M. Shayman // IEEE INFOCOM – The 27th Conference on Computer Communications (Phoenix, AZ, April 2008). – Phoenix, 2008. – P. 341-345.
5. Toumpis S. Optimal Deployment of Large Wireless Sensor Networks / S. Toumpis, L. Tassiulas // IEEE Trans. on Inform. Theory. – 2006. – Vol. 52. – No. 7. – P. 2935-2953.
6. Nguyen Nam T. Electric-Field-Based Routing: A Reliable Framework for Routing in MANETs / Nam T. Nguyen et al // ACM SIGMOBILE Mobile Computing and Communications Review. – 2004. – Vol. 8. – No. 2. – P. 35-49.
7. Нечаев Ю.Б. Маршрутизация в беспроводной сенсорной сети на основе геометрии силовых линий электростатического поля / Ю.Б. Нечаев, А.Д. Баев, А.В. Стромов // Вестник ВГУ. Серия: Системный анализ и информационные технологии. – 2010. – № 2. – С. 43-45.
8. Нечаев Ю.Б. Энергетическая трактовка функции потенциала в математической модели сверхбольшой беспроводной сенсорной сети / Ю.Б. Нечаев, А.Д. Баев, А.В. Стромов // Информатика: проблемы, методология, технологии: материалы XI Международной научно-методической конференции, Воронеж, 10-11 февраля 2011 г.: в 3 т. – Воронеж : Издательско-полиграфический центр Воронежского государственного университета, 2011. – Т.2. – С. 120-123.
9. Стромов А.В. Моделирование маршрутизации в беспроводной ячеистой сети, использующей простейшие адаптивные антенны / А.В. Стромов, Ю.Б. Нечаев, А.Д. Баев // Теория и техника радиосвязи, 2013. – Вып. 4. – С. 57-63.
10. Стромов А.В. елирование маршрутизации в беспроводной ячеистой сети с адаптацией к воздействию нескольких источников помех /

А.В. Стромов, Ю.Б. Нечаев, А.Д. Баев // Теория и техника радиосвязи, 2014. – Вып. 4. – С. 46-52.

11. Нечаев Ю.Б. Оценка пропускной способности беспроводной сети с регулярной топологией при конкурентном доступе к каналу / Ю.Б. Нечаев, А.В. Стромов, С.Р. Войтенко // Информатика: проблемы, методология, технологии : сборник материалов XVIII международной научно-методической конференции 8–9 февраля 2018 г. : в 7 т. – Воронеж : Издательство «Научно-исследовательские публикации» (ООО «Вэлборн»), 2018. – Т. 3. – С. 74-79.

СРАВНЕНИЕ МОДЕЛЕЙ МАССОВОГО ОБСЛУЖИВАНИЯ НА ПРИМЕРЕ ЧАТ В ПРОГРАММНОЙ СРЕДЕ ANYLOGIC

Д. А. Высоцкая

Магистрант

И. В Илларионов

Доцент

Введение

Огромное количество людей пользуется различными банковскими услугами и в современном мире появляется всё большая необходимость в решение своих вопросов удаленным способом. Посещение офиса банка является не всегда удобным, так как офисы работают не круглосуточно, а клиенты сталкиваются с серьёзными проблемами, например, в срочной блокировке карты или предотвращению мошеннических действий по своим счетам. Для решения проблем клиентов банку необходимо внедрять новые сервисы для обслуживания и на помощь, конечно же, приходят интернет сервисы. Существует несколько способов связаться с банком по интернету, например, написать по e-mail или оставить обращение на сайте банка по форме обратной связи, но данные ресурсы не могут быстро решить вопрос клиента и в этом случае лучшим решением будет использование онлайн-чата.

В статье будет рассмотрено две модели системы массового обслуживания чата созданные в Anylogic. Использование AnyLogic предоставляет уникальную возможность войти в мир моделирования, имея лишь базовую подготовку в области информационных технологий [1].

Первая модель будет с общей очередью. Модель будет показывать случай, когда по всем вопросам клиенты будут обращаться в общую очередь. Вторая с распределением на две очереди. Модель будет демонстрировать случай, когда вопросы клиентов в чате будут

распределены на два вида в разные очереди: общие (справочные) вопросы и персонализированные.

1. Базовые схемы СМО

Приведены две типовые схемы построения СМО, которые могут быть использованы при обращении клиентов в чат (рис. 1). Дадим этим схемам условные обозначения а и б.

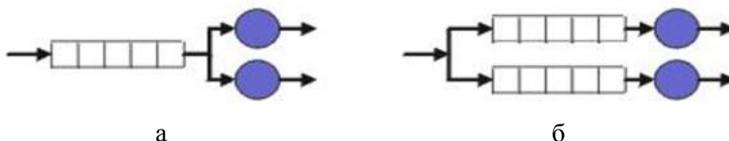


Рис. 1. Схемы обслуживания заявок: а – две точки обслуживания, б – в режиме поступления

Схема типа а: имеется две точки обслуживания. Обращающиеся клиенты образуют общую очередь и обслуживаются в свободной точке.

Схема типа б: клиенты по мере поступления распределяются равномерно (по 50%) по двум очередям с точками обслуживания.

В обоих случаях примем интенсивность обращения клиентов и интенсивность обслуживания постоянной.

2. Построение моделей в AnyLogic

Проведем моделирование в среде AnyLogic. Воспользуемся палитрой Enterprise Library. Для построения СМО используются элементы:

- Source – источник заявок.
- Queue – очередь ожидающих обслуживания заявок.
- Delay – Элемент моделирующий узел обслуживания.
- Sink – Элемент принимающий обработанные заявки [2].

Построим модель, которая соответствует схема а.

Построенная модель приведена на (рис. 2), а на (рис. 3) показаны объекты модели. Значения данных объектов модели приводятся в таблице.

Что бы определить доли заявок в общей совокупности заявок создадим функцию myGetP. Код функции на языке Java примет вид (листинг 1):

Листинг 1

Определение доли заявок

```
double myGetP(double sinkCount, double sourceCount) {  
    double x=0;  
    if (sourceCount!=0)  
        x=sinkCount/sourceCount; return x;  
}
```

Объекты модели

№	Объект	Тип	Назначение
1	mu	Параметр	Интенсивность работы процессоров delay. Принята равной 0,25
2	lambda	Параметр	Интенсивность поступления заявок в систему. Принята равной 0,5
3	qN	Параметр	Длина очереди. Принята равной 100
4	dSum	Динамическая переменная	Содержит вычисленную долю заявок обработанных системой в процентах
5	dPoteri	Динамическая переменная	Потерянные заявки в системе
6	dPoteriProc	Динамическая переменная	Вычисление доли потерянных заявок в процентах
7	N	Динамическая переменная	Количество заявок, которые могут одновременно находиться в системе
8	myGetP	Функция	Используется для вычисления доли заявок
9	data	Набор данных гистограммы	Формируется при уничтожении заявок элементом sink1
10	data1	Набор данных гистограммы	Формируется при уничтожении заявок элементом sink2

Функция содержит два формальных параметра sinkCount – число пришедших заявок, sourceCount – общее число заявок.

С учетом введенной функции код модели примет вид (листинг 2):

Листинг 2

Модель массового обслуживания

```
dSum= myGetP(sink1.count()+sink2.count(),
source.count())*100;
dPoteri=source.count()-(sink1.count()+sink2.count())
dPoteriProc= myGetP(dPoteri,source.count())*100
entity.sP=time()//Вход очереди,
//экземпляр класса MesClass
entity.fP=time()//Выход процессора,
//экземпляр класса MesClass
data.add(entity.fP-entity.sP)
N= lambda*(data.mean()+data1.mean())
```

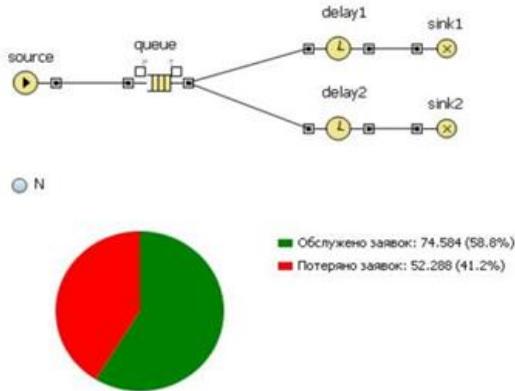


Рис. 2. Модель типа *a*

Количество заявок уничтоженных элементом sink определяется вызовом метода count(). Этот же метод поддерживает элемент source – порождающий заявки модели.

Чтобы получить время нахождения заявок в системе от момента входа до момента ее уничтожения, после обслуживания введем в модель специальный класс заявки MesClass.



Рис. 3. Объекты модели

Спецификация класса показана на (рис. 4).

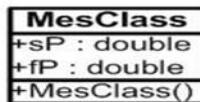


Рис. 4. Спецификация класса заявки

Класс заявки содержит два атрибута вещественного типа. Атрибут sP служит для фиксирования времени поступления заявки в очередь обслуживания, атрибут fP используется для фиксирования времени конца обслуживания заявки в процессоре.

Для наглядного представления процесса обслуживания разместим в модель круговую диаграмму. На диаграмме показана доля обслуженных заявок и доля потерянных заявок.

Модель, показанная на (рис. 5), соответствует схеме б. Объекты модели показаны на (рис. 6). Значение параметров и назначение объектов соответствует таблице.

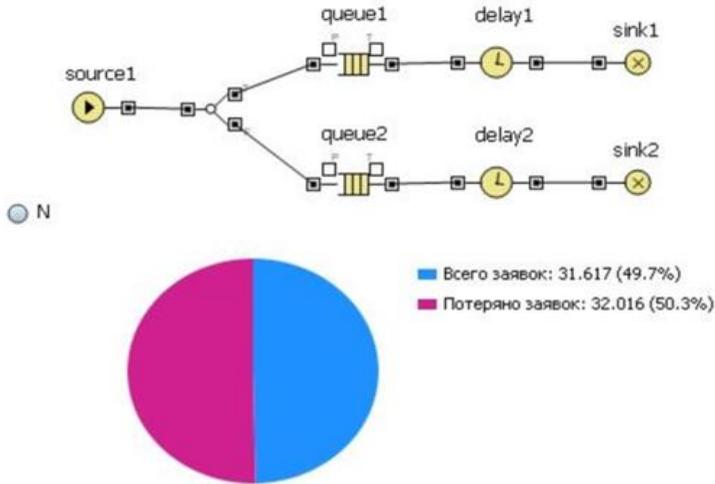


Рис. 5. Модель типа б

3. Анализ результатов моделирования

Результаты моделирования приводятся на (рис. 7) и (рис. 8). Моделирование проводилось в течение конечного промежутка времени равного 500 модельным единицам. Из рисунков видно, что модель с общей очередью позволяет одновременно находиться в системе 9 заявкам, при этом потери составили 3 заявки от общего числа поступивших за исследуемый период времени.

Модель с двумя очередями позволяет находиться в системе 25 заявкам, при этом потери за исследуемый период времени составили минимальное значение 0,4%.

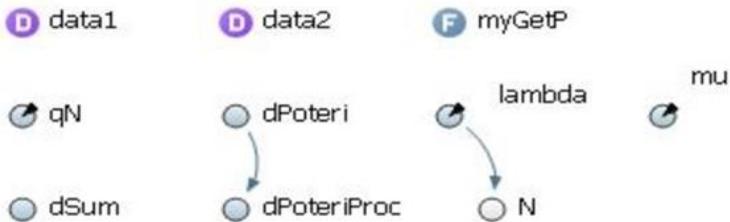


Рис. 6. Объекты модели

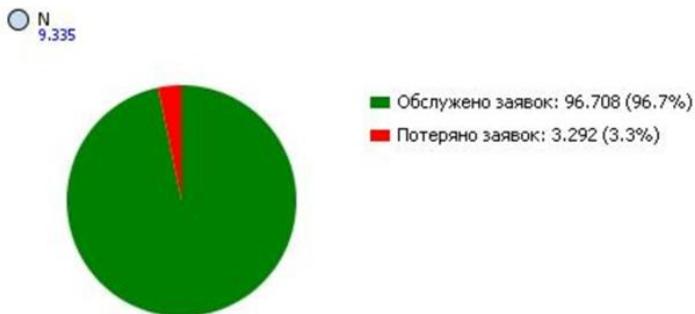


Рис. 7. Моделирование системы с одной очередью

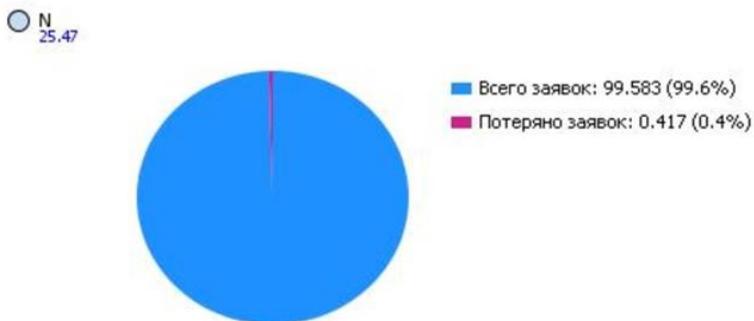


Рис. 8. Моделирование системы с двумя очередями

Следовательно, предпочтительнее использовать систему обслуживания клиентов при равном потоке с несколькими очередями.

Заключение

Моделирование с помощью AnyLogic помогает наглядно посмотреть и оценить преимущества и недостатки созданной модели. Необходимо осознавать, что добавление новых параметров усложняет модель и даёт оценить новые её возможности. С помощью созданной модели начинаешь понимать поведение системы, проверять, как она ведет себя при определенных условиях, сравнивать различные сценарии и оптимизировать ее. Когда оптимальное решение найдено, то можно применить модель в реальном мире. Актуальность данного процесса заключается в том, что при минимальных затратах можно найти идеальное решение. Поиск решения задачи будет осуществлен в защищенном от риска мире моделей, в котором можно ошибаться, отменять операции, возвращаться в прошлое и начинать все сначала [3].

Список литературы

1. Куприяшкин, А.Г. Основы моделирования систем : учеб. пособие / А.Г. Куприяшкин. – Норильск : НИИ, 2015. – 135 с.
2. Мезенцев. К.Н. Учебное пособие «Моделирование систем в среде AnyLogic 6.4.1». – Часть 2 / К.Н. Мезенцев; под редакцией А.Б. Николаева. – М. : МАДИ, 2011. – 103 с.
3. Григорьев И. AnyLogic за три дня. Практическое пособие по имитационному моделированию [Электронный ресурс]. – Режим доступа : <https://www.anylogic.ru/resources/books/free-simulation-book-and-modeling-tutorials/>

ПОДХОДЫ К РАСПОЗНАВАНИЮ ИНДЕНТИФИКАТОРОВ В ЗАДАЧЕ НАПИСАНИЯ КОДА ПРИ ПОМОЩИ ГОЛОСА

Д. Д. Вычиков

Студент

Д. И. Соломатин

Старший преподаватель

Введение

В настоящее время существуют решения, позволяющие с приемлемой точностью распознавать обычную речь человека в текстовое представление. Конечно, что-то исправить в тексте, поменять местами, расставить пунктуацию и т.п. традиционным способом, с помощью клавиатуры и мышки, будет все еще удобнее, однако отдельные слова, фразы и даже предложения вполне можно диктовать компьютеру.

Также все чаще используется голосовое управление и различного рода голосовые помощники, ярким примером которых является «Ok, Google».

Учитывая, что программирование является сферой, предполагающей постоянную работу с компьютером, предпринимались и предпринимаются попытки использования голосового управления для написания программного кода. При этом разработанные решения не получили сколько-либо широкого распространения, но нельзя однозначно сказать, почему: из-за очевидного неудобства предложенных решений или же причина в принципиальной невозможности построения удобного решения.

Основные заинтересованные лица в появлении такого решения – люди с различными заболеваниями, которые не могут свободно использовать клавиатуру. Однако такой подход даст возможность программировать на мобильных устройствах, а также, в теории, может увеличить скорость набора кода за счет диктовки.

1. Особенности распознавания идентификаторов

Построение удобного интерфейса для написания программного кода голосом имеет ряд открытых вопросов, которые надо решить, начиная с самого главного, а как будет удобно? Но одна задача очевидна сразу же – распознавание идентификаторов в речи.

Любой код, помимо всего прочего, состоит из определения классов и функций, полей и переменных, а также обращений к ним по их именам – идентификаторам. Важной задачей является обеспечение удобного доступа к этим именам с помощью голоса.

При этом следует учитывать следующие моменты:

- имена идентификаторов могут представлять собой различные сокращения и являться несуществующими в словарях словами, например, `strlen`, `malloc`;

- в каждой программе могут встречаться совершенно уникальные имена идентификаторов;

- частые ошибки и неточности при распознавании речи сильно затрудняют распознавание даже обычных, словарных идентификаторов.

В данной работе рассматриваются подходы к распознаванию идентификаторов, произнесенных голосом, с учетом обозначенных выше моментов.

2. Возможные подходы к распознаванию идентификаторов

Возможны два подхода для распознавания неизвестных слов.

Первый – это взаимодействие с системой распознавания речи: добавление новых слов в словарь. Эффективность подхода во многом зависит от функциональности используемого распознавателя. Одни

позволяют просто добавить слово в словарь, что увеличит вероятность его распознавания (Google Cloud Speech). Вторые вдобавок требуют предоставить вариант произношения – письменную транскрипцию слова, либо образец звуковой записи с произношением (Sphinx и Dragon Natural Speaking). Подход с голосовой тренировкой в качестве основного решения, очевидно, в рассматриваемой задаче неприменим, поскольку, во-первых, идентификаторы в каждой программе могут быть различны, а во-вторых, требует значительных усилий со стороны пользователя. Но в целом данный подход может улучшить результаты и в перспективе может быть ограниченно использован совместно с подходом, предложенным ниже.

Второй подход основан на применении алгоритмов нечеткого поиска и не зависит от того, какая система распознавания будет использована. Идея в том, что даже в случае некоторой ошибки при распознавании произнесенного идентификатора, скорее всего, будет получен результат, который по произношению очень схож с тем названием, которое необходимо было получить, хотя само написание полученного текста может кардинально отличаться от правильного варианта.

Если следовать этому предположению, то задача сводится к расчету некоторого фонетического индекса распознанного слова или фразы и дальнейшего поиска идентификатора из возможного списка с наиболее схожим индексом.

4. Описание используемых алгоритмов

Существует ряд фонетических алгоритмов, которые могут быть использованы в задаче такого рода, но среди них особенно выделяется Metaphone [1], который позиционируется как алгоритм получения примерного, а не точного, звучания слова. Неоспоримое преимущество этого алгоритма перед родственными ему Caverphone и Soundex заключается в том, что Metaphone позволяет получить индекс произвольной, а не фиксированной длины.

В тестовой реализации применялась вторая версия алгоритма (Double Metaphone), которая учитывает правила произношения английского и группы некоторых других языков. Также существует третья версия, которая, по словам автора, использует больший набор правил и работает более точно, однако является проприетарной.

Для сравнения фонетических индексов используется расстояние Левенштейна [2], которое определяется как число операций вставки / удаления / замены символа, необходимых для преобразований одной строки в другую. Для расчета используется алгоритм Вагнера-Фишера [3].

Конечный алгоритм для выбора идентификатора из набора выглядит следующим образом:

- получить фонетическое представление произнесенного слова;
- рассчитать расстояние Левенштейна для полученного представления с фонетическим представлением каждого идентификатора из списка возможных идентификаторов и выбрать идентификаторы с минимальным расстоянием;
- если в списке «ближайших» идентификаторов только один элемент, то вернуть его;
- если элементов несколько, то рассчитать расстояние Левенштейна между именами идентификаторов и распознанным текстом и выбрать идентификаторы с наименьшим расстоянием;
- если таких идентификаторов будет несколько, то вернуть первый (или произвольный) из списка.

Данный алгоритм в дальнейшем может быть улучшен за счет введения вероятностных коэффициентов для списка идентификаторов. Более того, при тесном взаимодействии с интеллектуальным редактором кода эти коэффициенты могут даже пересчитываться динамически в зависимости от контекста в «набираемом» коде.

5. Методика тестирования и полученные результаты

В качестве источника идентификаторов была взята открытая реализация клиентской библиотеки Google Cloud Platform на C#. Из исходного кода удалены комментарии, а из оставшейся части выбраны слова по некоторому принципу. Выборка, состоящая из имен классов, методов, переменных, а также некоторого неотфильтрованного мусора содержала 13000 уникальных идентификаторов. Также в этот список было добавлено примерно два десятка непроизносимых имен функций, взятых из стандартной библиотеки языка C.

К слову, в реальных программах для любого отдельного модуля (файла исходного кода) пространство имен идентификаторов никогда не содержит такого количества элементов даже с учетом подключенных в этом модуле сторонних модулей или иных файлов. Однако такой объем выборки способен показать точность работы алгоритма.

В качестве системы распознавания речи использовался облачный сервис Google Speech-To-Text. Язык переменных – английский. Язык распознавания – русский. Стоит заметить, что Google при выбранном русском языке речи распознает и английские слова, делая поправку на русский акцент (например, слово «items» можно произнести как «и-тимс», а не обычное «ай-темс»). Сервис позволяет предоставить набор контекстных слов для улучшения их распознавания, однако их количество имеет ограничение 5000, поэтому эта возможность не использовалась.

Для эксперимента использовался микрофон, встроенный в недорогую веб-камеру.

По субъективной оценке, большая часть имен, произнесенных во время эксперимента, оказалась идентификаторами, состоящими как минимум из двух слов более-менее словарного типа.

Для эксперимента было произнесено 370 имен. Из них успешно распознано 244, что составляет примерно 60%. Однако, нужно понимать, что полученный результат зависит от многих факторов, а не только от качества работы примененных алгоритмов. Большую роль безусловно играет микрофон, а также манера произношения.

Около половины всех ошибок распознавания произнесенного идентификатора произошли при некачественном распознавании и были исправлены на повторной попытке при более четком произношении.

Некоторая часть ошибок была получена при попытках распознать одно из ряда очень похожих слов, которые отличаются на 1 символ, как например, «vsProjectKindUnmodeled» (FSPRJTKNTNMTLT) и «csProjectKindUnmodeled» (SPRJTKNTNMTLT) (в скобках записано фонетическое представление). При результате распознавания «cs project canton models» (KSPRJTKNTNMTLS) дистанция составила 2 символа в обоих случаях. Дальнейшее сравнение дистанции между самими именами дало расстояния в 7 символов для обоих вариантов. Теоретически, если вычисление расстояния Левенштейна для некоторых идентификаторов одинаково, можно дополнительно пытаться рассматривать некоторые ключевые позиции в слове (в данном случае это первый символ).

Распознавание коротких и несловарных идентификаторов вроде malloc (MLK), csetjmp (KSTMP), cstdio (KST) оказалось неудовлетворительным. Из-за наличия в выборке большого числа идентификаторов из одного слова, количество схожих вариантов произношения оказалось слишком велико. Полученный результат во многом объясняется тем, что алгоритм Metaphone не учитывает гласные звуки (только если они не в начале слова).

Заключение

Исходя из результатов эксперимента, предложенное решение может быть использовано для распознавания составных идентификаторов даже при большом количестве возможных имен.

Для распознавания коротких и несловарных (сокращенных) имен идентификаторов данный подход в текущей реализации использоваться не может. Для улучшения результатов, вероятно, стоит воспользоваться каким-либо иным фонетическим алгоритмом, который дает на выходе более точное, а не приближенное звучание слова.

Кроме того, очевидно, что однозначно распознать любые произносимые идентификаторы не получится, и необходимо будет использовать контекст для фильтрации идентификаторов, а также

реализовать в редакторе с поддержкой голосового набора удобный механизм выбора среди возможных близких по произношению идентификаторов в неоднозначных ситуациях.

Список литературы

1. Lawrence, P. Hanging on the Metaphone / P. Lawrence // Computer Language. – 1990. – Vol. 7. – № 12. – P. 39-44.
2. Левенштейн, В. И. Двоичные коды с исправлением выпадений, вставок и замещений символов. / В. И. Левенштейн // Доклады Академий Наук СССР. – 1965. – Т. 163. – № 4. – С. 845-848.
3. Donald, E. K. The Art of Computer Programming / E. K. Donald. – Boston : Addison-Wesley, 1998. – Vol. 3. – 391 p.

НОВЫЕ СПОСОБЫ ИНТЕРПОЛЯЦИИ С ПОМОЩЬЮ ОКОННЫХ СИСТЕМ ФУНКЦИЙ

М. А. Галыгина

Студент

Е. А. Киселев

Ассистент

Введение

В настоящее время идет быстрое развитие информационных технологий. Они используются в технике, науке, медицине и т. д. Возникает потребность в корректной обработке цифровых сигналов. Например, медицинские приборы измеряют входной сигнал не непрерывно, а только в дискретные моменты времени. Пользователь при этом сталкивается с проблемой: с одной стороны, прибор должен максимально точно отобразить результаты измерений, с другой – трудно визуально воспринять форму сигнала, изображенного просто набором точек. Здесь на помощь приходит интерполяция, которая использует значения некоторой функции, заданные в ряде точек, чтобы приблизить значения функции между ними.

К сожалению, различные методы интерполяции не лишены недостатков. Во-первых, они требуют для вычисления каждой промежуточной точки суммирования бесконечного ряда слагаемых, учитывающих все возможные дискретные измерения сигнала в прошлом и будущем, что, конечно, физически не реализуемо. И, во-вторых, даже если мы ограничимся только имеющимися точками, вычислительная сложность метода достаточно высока. На практике обычно и с первой, и со второй

проблемами борются использованием оконных функций, ограничивающих область вычислений интерполируемой точки некоторыми небольшими окрестностями, причем вес ближайших узловых точек выше, чем дальних. Наиболее распространенными оконными функциями являются треугольное окно и окно в виде функции Гаусса.

1. Задача интерполяции

Пусть дана некоторая функция $f(x)$. Выберем набор точек (узлов) x_1, x_2, \dots, x_n . Задача интерполяции состоит в построении некоторой функции $\tilde{f}(x)$, которая будет совпадать с $f(x)$ в заданных узлах, т. е.

$$\tilde{f}(x_i) = f(x_i), i = 1, 2, \dots, n. \quad (1)$$

Обычно $\tilde{f}(x)$ строится в виде линейной комбинации некоторых функций $\varphi_j(x), j \in Z$

$$\tilde{f}(x) = \sum_j c_j \varphi_j(x). \quad (2)$$

Тогда задача интерполяции сводится к нахождению коэффициентов c_j . Это удобно делать с помощью узловых функций [1].

Функция $\varphi_j(x)$ называется узловой, если она удовлетворяет соотношению

$$\varphi_j(x_i) = \delta_{i,j}, i, j \in Z. \quad (3)$$

Решение задачи интерполяции строится тогда следующим образом

$$\tilde{f}(x) = \sum_j f(x_j) \varphi_j(x). \quad (4)$$

На практике обычно сигнал задается на равномерной дискретной сетке точек, т. е. с некоторым постоянным шагом. Путем замены переменных сетка всегда может быть сделана целочисленной с шагом 1. Поэтому особо актуальны в задачах интерполяции системы целочисленных сдвигов. Пусть имеется заданная на всей вещественной оси функция $\varphi(x)$ и отвечающая ей система целочисленных сдвигов

$$\varphi_j(x) = \varphi(x - j), j \in Z. \quad (5)$$

В этом случае достаточно построить всего одну функцию $\varphi_0(x)$. Остальные $\varphi_j(x)$ являются просто ее сдвигами

$$\varphi_j(x) = \varphi_0(x - j). \quad (6)$$

В рамках данной работы мы детально рассмотрим систему целочисленных сдвигов функции Гаусса, играющую важную роль в цифровой обработке сигналов.

2. Построение узловой функции

Обычно узловая функция ищется в виде ряда по сдвигам $\varphi(x-j)$. Тогда для ее построения необходимо найти коэффициенты d_k в следующей формуле

$$\Phi_0(x) = \sum_{k=-\infty}^{\infty} d_k \varphi(x-k). \quad (7)$$

В случае функции Гаусса

$$\varphi(x) = \exp\left(-\frac{x^2}{2\sigma^2}\right) \quad (8)$$

коэффициенты $d_k = d_k(\sigma)$ – зависят от σ . Аналитическое выражение для $d_k(\sigma)$ приведено в монографии В. Мазы и Г. Шмидта [2]

$$d_k(\sigma) = \frac{1}{C(\sigma)} \exp\left(\frac{k^2}{2\sigma^2}\right) \sum_{r=|k|}^{\infty} (-1)^r \exp\left(-\frac{(r+0.5)^2}{2\sigma^2}\right), \quad (9)$$

где

$$C(\sigma) = \sum_{r=-\infty}^{\infty} (4r+1) \exp\left(-\frac{(2r+0.5)^2}{2\sigma^2}\right). \quad (10)$$

Оказывается, что коэффициенты $d_k(\sigma)$ уже при $\sigma > 2$ очень большие по модулю (например, $d_{200}(3) \sim 10^{11}$). Это ведет к потере вычислительной устойчивости [3]. Известно, что решение (9) задачи (7) единственно [2], поэтому для построения каких-либо новых видов узловых функций, необходимо выйти за рамки целочисленных сдвигов функции Гаусса.

3. Оконная система

Естественным обобщением семейств сдвигов являются оконные системы функций [4]

$$\varphi_{ab}(x) = g(x-a)e^{ibx}, a, b \in R, \quad (11)$$

где $g(x)$ называется функцией окна. Мы рассмотрим случай, когда $g(x)$ задается формулой (8), а параметр a будет целым числом. Параметр b также удобно выбрать с некоторым постоянным шагом. Тогда семейство целочисленных сдвигов функции Гаусса будет подсистемой для (11).

С помощью оконной системы (11), нами были построены два новых типа узловых функций. Первая из них задается формулой

$$\Phi_a(x) = \frac{1}{4\pi} \Phi_0(x) \sin(4\pi x), \quad (12)$$

где $\varphi_0(x)$ определяется формулами (7) и (9). Функция $\varphi_d(x)$ обращается в нуль во всех целых точка, а узловой она является с точки зрения производной, т. е.

$$\varphi_d(i) = 0, \varphi'_d(i) = \delta_{i,0}, i \in \mathbb{Z}. \quad (13)$$

Тогда путем совместного применения $\varphi_0(x)$ и $\varphi_d(x)$ можно аппроксимировать не только саму исходную функцию $f(x)$, но и ее производную. На рис. 1 приведен график узловой функции $\varphi_d(x)$.

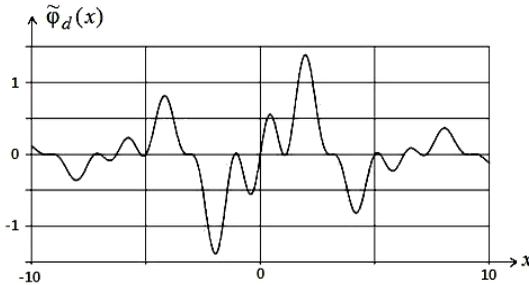


Рис. 1. Узловая функция $\varphi_d(x)$

Вторая разновидность, полученных нами узловых функций, устроена следующим образом. Функция

$$\varphi_{1/2}(x) = \sum_{k=-\infty}^{\infty} d_k \left(\frac{\sigma}{2} \right) \exp \left(- \frac{(x-2k)^2}{2\sigma^2} \right) \quad (14)$$

содержит коэффициенты $d_k \left(\frac{\sigma}{2} \right)$, которые по модулю значительно меньше, чем $d_k(\sigma)$. Однако $\varphi_{1/2}(x)$ будет обращаться в нуль только во всех четных целых точках (кроме нуля), т. е. не является узловой.

Рассмотрим функцию $\varphi_h(x)$

$$\varphi_h(x) = \varphi_{1/2}(x) \cos \left(\frac{\pi x}{2} \right). \quad (15)$$

За счет косинуса $\varphi_h(x)$ будет обращаться в нуль и во всех нечетных целых точках. При том же σ , что и в $\varphi_0(x)$, ряды для $\varphi_h(x)$ будут сходиться быстрее, поскольку $d_k \left(\frac{\sigma}{2} \right)$ по модулю меньше, чем $d_k(\sigma)$. При этом нам пришлось выйти за рамки сдвигов функции Гаусса, добавив

тригонометрические функции. Фактически это означает переход от семейства сдвигов к более общей оконной системе. График функции $\tilde{\varphi}_h(x)$ приведен на рис. 2.

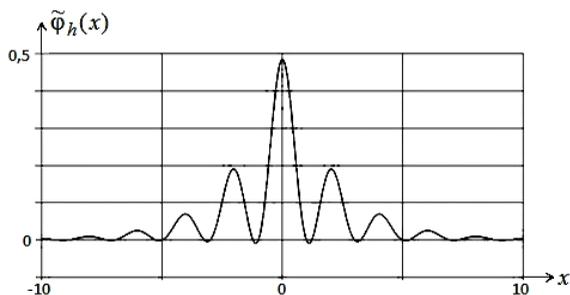
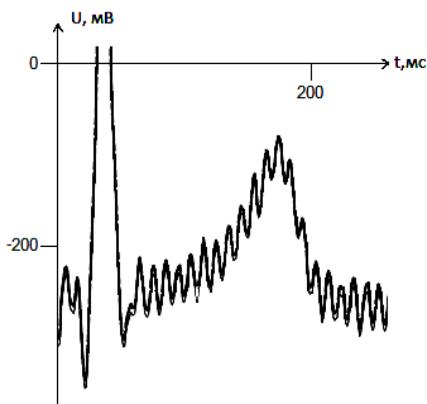


Рис. 2. Узловая функция $\tilde{\varphi}_h(x)$

На рис. 3 приведен пример работы с реальным сигналом ЭКГ при помощи $\tilde{\varphi}_h(x)$. На одном графике изображен увеличенный фрагмент исходной ЭКГ и интерполирующая функция $\tilde{f}(t)$ при $\sigma=1$. По оси абсцисс – время в миллисекундах, по оси ординат – напряжение в милливольтках. Узловая функция $\tilde{\varphi}_h(x)$ построена всего по трем слагаемым, поэтому видны небольшие отличия. В целом, даже при трех слагаемых наблюдаем достаточно хорошее качество аппроксимации.



ЭКГ (жирная линия) и интерполирующая функция (тонкая линия)

Рис. 3. Фрагмент ЭКГ

Заключение

В данной работе нами построены несколько новых типов узловых функций для оконных систем. Первый тип узловых функций позволяет проводить интерполяцию не только значений самой функции, но и ее производных. Это дает возможность проводить более гладкую кривую, повышая точность аппроксимации. Второй тип узловой функции позволяет повысить скорость сходимости рядов.

В работе были рассмотрены всего два наиболее простых примера, однако, с помощью оконных систем можно построить достаточно много других видов узловых функций. Изучение их свойств и выбор наиболее оптимальных вариантов будет являться темой будущих исследований.

Список литературы

1. Бахвалов Н. С. Численные методы / Н. С. Бахвалов, Н. П. Жидков, Г. М. Кобельков. – М. : Наука, 1987. – 598 с.
2. Maz'ya V. Approximate approximations / V. Maz'ya, G. Schmidt. – AMS Mathematical Surveys and Monographs, 2007. – v. 141. – 350 p.
3. Журавлев М. В. О вычислительных особенностях интерполяции с помощью целочисленных сдвигов гауссовых функций / М. В. Журавлев, Л. А. Минин, С. М. Ситник // Научные ведомости Белгородского государственного университета. – 2009. – № 13 (68). – вып. 17/2. – С. 89-99.
4. Добеши И. Десять лекций по вейвлетам / И. Добеши. – Ижевск : НИЦ «Регулярная и хаотическая динамика», 2004. – 464 с.

ФОРМИРОВАНИЕ ИНТЕГРАЛЬНЫХ ОЦЕНОК АЛЬТЕРНАТИВ НА ОСНОВЕ АНАЛИЗА ИХ ПАРАМЕТРОВ

К. Г. Гетманова

Студент

М. Г. Матвеев

Профессор

Введение

Часто возникают ситуации, когда необходимо оценить, упорядочить или сравнить между собой по некоторому, не поддающемуся непосредственному измерению, признаку множество объектов (альтернатив), например, определить благонадежность поставщика, пригодность оборудования по его параметрам для заказчика или компетентность сотрудника по качеству выполнения поставленных перед ним задач и т.п. Рассмотрение именно альтернативных вариантов не

случайно и связано непосредственно с необходимостью обоснованного ранжирования этих вариантов при выборе одного из ряда исключаящих друг друга.

В данном случае речь идет о сложных объектах, характеризуемых рядом показателей (параметров), количество которых велико - порядка нескольких десятков и более. Обобщенный признак, используемый для сравнения альтернатив, условно будем называть «качеством».

Указанная проблема требует исследования следующих аспектов:

- учет возможных связей между показателями, выделение среди множества признаков, характеризующих альтернативу, значимых;
- выбор способа агрегирования значений отдельных параметров для построения интегральной оценки объектов.

В работе представлен подход к формированию интегральных оценок альтернатив, рассмотрен практический пример. Стоит отметить, что приведенный выше перечень аспектов проблемы не является исчерпывающим и включает только вопросы, представляющие интерес для целей данной работы.

1. Постановка задачи

Рассматривается множество альтернатив $A = \{a_1, a_2, \dots, a_n\}$, каждая из которых характеризуется вектором параметров $X = \{x_1, x_2, \dots, x_m\}$; конкретные значения параметров x_1, \dots, x_m альтернативы a_i представляют собой вектор $K^i = \{k_1, k_2, \dots, k_m\}$. Требуется осуществить переход к новому набору параметров $F = \{F_1, F_2, \dots, F_p\}$, $p \ll m$, где F_j являются линейными комбинациями элементов множества X , при условии, что такой переход максимизирует информативность и вариабельность исходных данных.

На основе сформированного вектора $F = \{F_j\}$ необходимо получить интегральную скалярную оценку Y «качества» альтернативы согласно правилу Agg , предварительно определив вид этого правила:

$$Y = Agg(F_1, F_2, \dots, F_p). \quad (1)$$

2. Снижение размерности пространства параметров

Рассмотрим процесс снижения размерности пространства параметров на примере данных по финансовым показателям ($m = 11$) ряда фирм. Анализируются параметры, приведенные в табл. 1.

Таблица 1

Перечень параметров, характеризующих альтернативы

Обозначение	Показатель
x_1	Дебиторская задолженность, пред. год
x_2	Дебиторская задолженность, тек. год
x_3	Кредиторская задолженность, пред. год
x_4	Кредиторская задолженность, тек. год
x_5	Выручка по продажам, пред. год
x_6	Выручка по продажам, тек. год
x_7	Чистая прибыль (убыток), пред. год
x_8	Чистая прибыль (убыток), тек. год
x_9	Наличие чистых активов
x_{10}	Наличие оборотных средств
x_{11}	Задолженность перед бюджетом

К набору тестовых данных, а именно к матрице объекты-признаки $K[n \times m]$, применена процедура автошкалирования. Таким образом, $\sigma(x_i) = 1$, $\bar{x}_i = 0$, $\forall i$.

В результате исследования матрицы K с помощью метода главных компонент средствами надстройки MS Excel XLSTAT был осуществлен переход к новому множеству параметров – к компонентам $F_1 - F_4$.

$$F_1 = 0,845x_1 + 0,907x_2 + 0,786x_3 + 0,922x_4 + 0,868x_5 + 0,935x_6 - 0,407x_7 - 0,015x_8 + 0,242x_9 + 0,858x_{10} + 0,459x_{11}, \quad (2)$$

$$F_2 = -0,065x_1 - 0,076x_2 - 0,053x_3 - 0,099x_4 + 0,05x_5 + 0,134x_6 + 0,761x_7 + 0,925x_8 + 0,911x_9 - 0,063x_{10} + 0,532x_{11}, \quad (3)$$

$$F_3 = 0,292x_1 - 0,223x_2 + 0,52x_3 - 0,15x_4 + 0,371x_5 + 0,044x_6 + 0,388x_7 - 0,028x_8 - 0,027x_9 - 0,322x_{10} - 0,518x_{11}, \quad (4)$$

$$F_4 = 0,277x_1 - 0,012x_2 + 0,042x_3 - 0,162x_4 + 0,006x_5 - 0,059x_6 - 0,067x_7 - 0,057x_8 - 0,097x_9 - 0,249x_{10} + 0,354x_{11}. \quad (5)$$

Как показано на рис. 1, четырех главных компонент достаточно, чтобы сохранить информативность исходных данных на 98%.

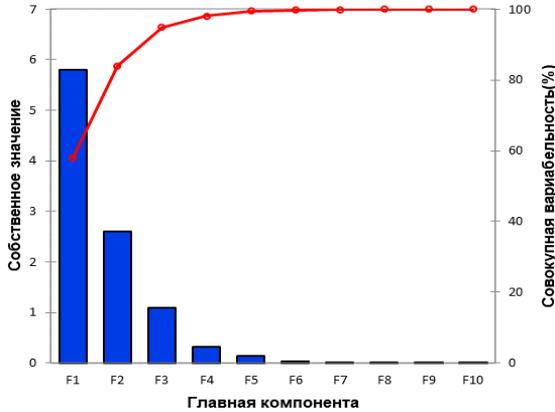
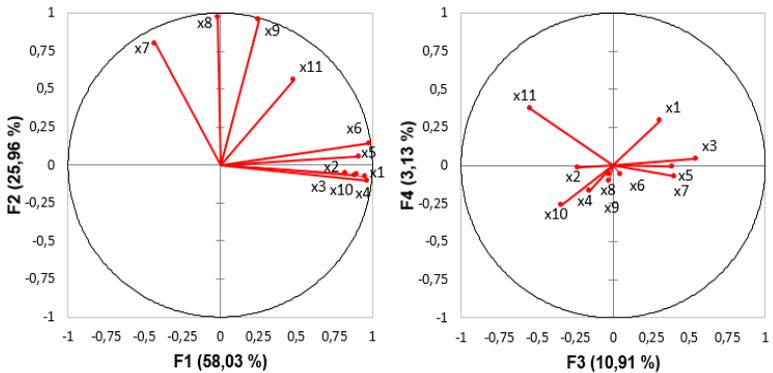


Рис. 1. График собственных значений главных компонент

При этом наиболее сильная связь существует между компонентой F_1 и параметрами $x_1 - x_6, x_{10}$, компонентой F_2 и параметрами $x_7 - x_9, x_{11}$, компонентой F_3 и параметрами x_3, x_7, x_{11} , компонентой F_4 и параметрами x_1, x_{10}, x_{11} , что на рис. 2 соответствует проекциям векторов \overline{Ox} на оси компонент.

Также на рис. 2 в скобках рядом с обозначением компоненты указан процент, сохраняемой ею информативности [1].



слева – корреляция параметров $x_1 - x_{11}$ и компонент F_1, F_2

справа – корреляция параметров $x_1 - x_{11}$ и компонент F_3, F_4

Рис. 2. Схема корреляции исходных параметров и главных компонент

Непосредственное сравнение параметров альтернативы невозможно, так как каждый из параметров интерпретируется по-разному и имеет собственный смысл, потому исходные величины необходимо представить в безразмерном виде. Для этого предлагается на каждом интервальном значении параметра вводить функцию принадлежности $\mu(x_i)$, которая определяет степень соответствия нечетким требованиям фактических значений параметров альтернатив и принимает значения на отрезке $[0, 1]$. Тогда для получения общего представления о «качестве» альтернативы необходимо агрегировать значения функций принадлежности ее параметров.

Так как для сокращения количества рассматриваемых признаков исходные параметры были заменены главными компонентами, то требуется определить, какой вид имеют функции принадлежности главных компонент $\mu(F_j)$.

Для этого выполним следующие действия:

- на функции $\mu(x_i)$ наложим ограничение – все функции принадлежности должны быть треугольными, т.е. описываться тройками (l_i, m_i, r_i) , где l_i и r_i – соответственно левая и правая границы допустимых значений i -го параметра, m_i – наиболее оптимальное значение i -го параметра;

- вычислим значения (L_j, M_j, R_j) главных компонент на основе значений (l_i, m_i, r_i) согласно соотношениям (2-5) и положим, что функции $\mu(F_j)$ также треугольны, а значения $\mu(F_j)$ в рассчитанных точках (L_j, M_j, R_j) , где L_j и R_j – левая и правая границы значений главной компоненты j , M_j – оптимальное значение главной компоненты j , соответствуют значениям $\mu(x_i)$, т.е. $\mu(L_j) = \mu(R_j) = 0$, а $\mu(M_j) = 1$;

- используя тройки (L_j, M_j, R_j) , определим аналитический вид функций принадлежности главных компонент F_j .

В результате получим:

$$\mu(F_1) = \begin{cases} 1 + 0,25F_1, & F_1 \in [-3,96;0] \\ 1 - 0,07F_1, & F_1 \in [0;15,50] \\ 0, & F_1 \notin [-3,96;15,50] \end{cases}, \mu(F_2) = \begin{cases} 1 + 0,36F_2, & F_2 \in [-2,77;0] \\ 1 - 0,14F_2, & F_2 \in [0;7,33] \\ 0, & F_2 \notin [-2,77;7,33] \end{cases}$$

$$\mu(F_3) = \begin{cases} 1+0,89F_3, & F_3 \in [-1,12;0] \\ 1-2,86F_3, & F_3 \in [0;0,35] \\ 0, & F_3 \notin [-1,12;0,35] \end{cases}, \mu(F_4) = \begin{cases} 1+1,3F_4, & F_4 \in [-0,77;0] \\ 1-16,7F_4, & F_4 \in [0;0,06] \\ 0, & F_4 \notin [-0,77;0,06] \end{cases} \quad (6)$$

На основе исходных данных по выражениям (2-5) рассчитываются значения главных компонент для каждой из альтернатив a_i . Для определения степени принадлежности значения F_j подставляются в соответствующую функцию (6). В табл. 2 приведены значения $\mu(F_j)$ первых пяти альтернатив из набора тестовых данных.

Таблица 2

Значения функций принадлежности главных компонент

Альтернатива	$\mu(F_j)$			
	$j=1$	$j=2$	$j=3$	$j=4$
a_1	0,26	0,72	0,54	0,95
a_2	0,67	0,91	0,56	0,53
a_3	0,26	0,69	0,51	0,91
a_4	0,81	0,88	0,79	0,91
a_5	0,31	0,79	0,70	0,86

3. Агрегирование оценок по отдельным компонентам

Под агрегированием понимается отображение $Agg : [0,1]^p \rightarrow [0,1]$, ставящее в соответствие вектору $F = \{F_1, F_2, \dots, F_p\}$ скалярную величину Y , которая называется интегральной оценкой объекта. Выполняется агрегирование, как правило, по одному из направлений: арифметическое, геометрическое, гармоническое.

Оператор агрегирования будем выбирать в классе средних, то есть удовлетворяющий ограничению

$$Agg(F_1, F_2, \dots, F_p) \in [\min\{F_1, \dots, F_p\}, \max\{F_1, \dots, F_p\}] \quad (7)$$

Рассмотрим среднее α -порядка

$$Agg_\alpha(F_1, F_2, \dots, F_p) = \left(\frac{\sum_{j=1}^p F_j^\alpha}{p} \right)^{\frac{1}{\alpha}} \quad (8)$$

При $\alpha = 1$ выражение (8) представляет среднее арифметическое, при $\alpha = -1$ – среднее гармоническое, при $\alpha = -\infty$ – операцию минимума, при $\alpha = \infty$ – операцию максимума [2-3].

Для агрегирования степеней соответствия главных компонент будем использовать так называемый «пессимистический подход», т.е. операцию поиска минимума.

В табл. 3 указано агрегированное значение принадлежности, характеризующее «качество» альтернатив и определенное согласно табл. 2.

Таблица 3

Интегральные оценки альтернатив

Альтернатива	Интегральная оценка Y
a_1	0,26
a_2	0,53
a_3	0,26
a_4	0,79
a_5	0,31

Таким образом, из рассмотренных $a_1 - a_5$ альтернатив четвертая обладает лучшим «качеством».

Заключение

В данной работе был рассмотрен один способ обобщенного оценивания объектов-альтернатив. В качестве базовых подходов были выбраны метод главных компонент для сокращения размерности признакового пространства и операция минимума для агрегирования оценок по отдельным компонентам. В среде MS Excel с помощью статистических надстроек был реализован описанный алгоритм.

Результаты проведенного исследования могут быть использованы для формирования матриц коммерческого и технического соответствия при решении задачи, сформулированной в работе [4].

Список литературы

1. Прикладная статистика: Классификация и снижение размерности: справ. изд. / С.А. Айвазян [и др.]. – М. : Финансы и статистика, 1989. – 607 с.
2. Леденева, Т.М. Агрегирование информации в оценочных системах / Т.М. Леденева, С.Л. Подвальный // Вестник ВГУ. Серия: Системный анализ и информационные технологии. – 2016. – № 4. – С. 155-164.

3. Леденева, Т.М. Аксиоматический подход к построению функций агрегирования для оценочных систем / Т.М. Леденева, Д.А. Денисихина // Вестник ВГУ. Серия: Системный анализ и информационные технологии. – 2014. – № 3. – С. 33-39.

4. Будяков, А.Н. Решение задачи выбора ресурсов и их поставщиков в условиях противоречивости технических и коммерческих требований / А.Н. Будяков, К.Г. Гетманова, М.Г. Матвеев // Вестник ВГУ. Серия: Системный анализ и информационные технологии. – 2017. – № 2. – С. 66-71.

ИССЛЕДОВАНИЕ ЭФФЕКТА ГИББСА ПРИ ВОССТАНОВЛЕНИИ РАЗРЫВНЫХ ФУНКЦИЙ

Е. А. Гребенщикова

Студент

А. Ф. Клинских

Профессор

Введение

Задачи приближения разрывных функций ставятся значительно чаще, чем задачи приближения непрерывных функций. А явление Гиббса однозначно наблюдается при восстановлении разрывных функций и встречается в таких предметных областях как обработка рядов Фурье, обработка и восстановление изображений, сжатие видео и изображений, преобразование сигналов [1-3]. В задачах обработки изображений эффект Гиббса проявляется как эффект ложного оконтуривания, возникающий при недостатке информации о высоких частотах изображения и проявляющийся в виде ореолов возле резких контуров. Также эффект Гиббса возникает при преобразовании сигналов. Следовательно, проблема преодоления эффекта Гиббса при восстановлении разрывных функций является актуальной.

Цель работы – с использованием конкретных примеров разрывных функций проанализировать предлагаемые в литературе алгоритмы преодоления эффекта Гиббса. А также провести сравнение, обсудить возможности практических приложений и рассмотреть пути дальнейшего исследования этого эффекта.

1. Проявления эффекта Гиббса

Эффект Гиббса – это особенность поведения частичных сумм ряда Фурье в окрестности точки разрыва функции. Рассмотрим явление Гиббса на примере частного случая разложения в ряд Фурье разрывной функции с периодом 2π на ось t (см. рис. 1).

$$f(t) = \text{sign}(t) = \begin{cases} -1 \rightarrow n\pi - \pi < t < 0, \\ 0 \rightarrow n\pi = 0, \\ +1 \rightarrow n\pi < t < \pi, \end{cases} \quad (1)$$

Функция (1) является нечётной, поэтому её разложение в ряд Фурье будет содержать только синусы:

$$f(t) = \sum_{m=1}^{\infty} b_m \sin mt, \quad (2)$$

где коэффициенты разложения b_m определяются, с учётом (1), выражениями

$$b_m = \frac{1}{\pi} \int_{-\pi}^{\pi} f(t) \sin mtdt = \frac{2}{\pi} \int_0^{\pi} \sin mtdt = \frac{2}{\pi} \frac{1 - \cos \pi n}{m} = \frac{2}{\pi n} [1 - (-1)^m]$$

т. е. в разложении (2) отличны от нуля только слагаемые с нечётными номерами $m = 2n - 1$, следовательно, разложение в ряд Фурье приводится к следующему виду

$$f(t) = \lim_{N \rightarrow \infty} S_N(t),$$

где $S_N(t)$ – сумма первых N ненулевых слагаемых ряда Фурье:

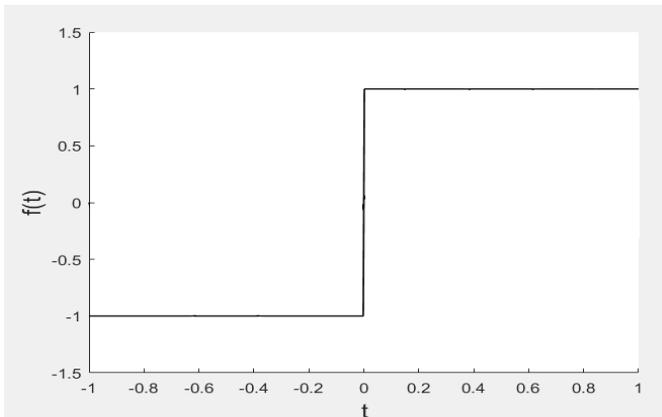


Рис. 1. Периодическая функция (1), разлагаемая в ряд Фурье (2)

$$S_N(t) = \frac{4}{\pi} \sum_{n=1}^N \frac{\sin(2n-1)t}{2n-1}. \quad (3)$$

Характер приближения функции (1) частичными суммами (3) её ряда Фурье изображён на рис. 2.

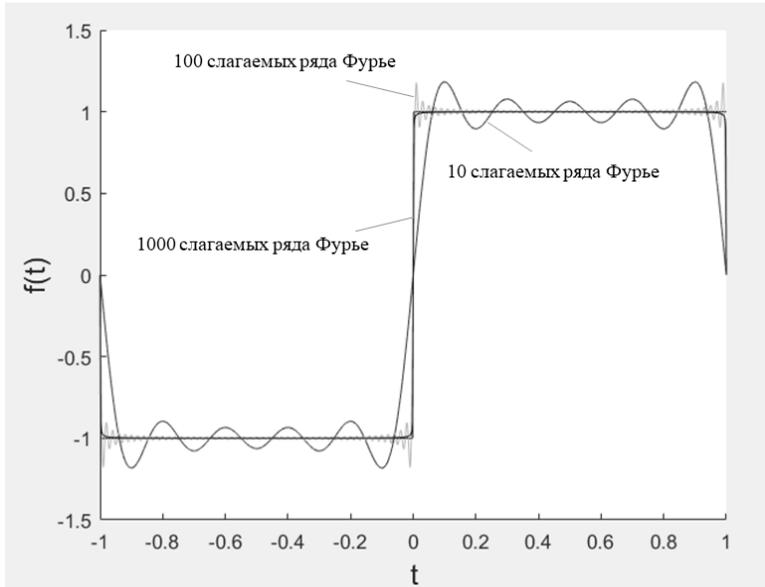


Рис. 2. Разлагаемая функция (1) и частичные суммы S_{10} , S_{100} , S_{1000} , её ряда Фурье (3)

Как видно из рис. 2, частичные суммы S_N осциллируют вблизи значений ± 1 разлагаемой функции. Исследуя, как ведут себя экстремумы $S_N(t)$ (в которых S_N наиболее сильно отклоняется от разлагаемой функции) при $N \rightarrow \infty$, можно сделать вывод, что функция при разложении в ряд Фурье в точках разрыва имеет скачки значений суммы ряда, которые превышают величину скачков значений самой функции на 17,9%.

2. Пути устранения пульсаций вблизи точки разрыва

В данной статье рассмотрим три алгоритма.

- Сглаживание Ланцоша

Ланцош заметил, что пульсации в сумме усеченного ряда имеют период либо первого отброшенного слагаемого, либо последнего удержанного слагаемого. Он доказал, что в любом случае, сглаживая

частичную сумму путем интегрирования (усреднения) по этому периоду, можно устранить основные всплески.

Рассмотрим применение этой идеи к усеченному ряду Фурье в общем виде

$$g_N(t) = \frac{a_0}{2} + \sum_{k=1}^N (a_k \cos kt + b_k \sin kt).$$

В качестве сглаженного значения возьмем среднее значение на интервале длиной $2\pi/N$ с центром в точке t (здесь мы принимаем N за номер последнего удерживаемого слагаемого, чтобы продемонстрировать исчезновение этого слагаемого, но на практике N берется как номер первого отбрасываемого слагаемого). Поэтому, сглаженное значение $h(t)$ получим как среднее от $g(t)$:

$$h_N(t) \frac{N}{2\pi} = \int_{t-(\pi/N)}^{t+(\pi/N)} g_N(s) ds.$$

В подробной записи получим

$$h_N(t) = \left(\frac{a_0}{2} \right) \left(\frac{N}{2\pi} \right) \int_{t-(\pi/N)}^{t+(\pi/N)} ds + \frac{N}{2\pi} \sum_{k=1}^N \left[a_k \int_{t-(\pi/N)}^{t+(\pi/N)} \cos ks ds + b_k \int_{t-(\pi/N)}^{t+(\pi/N)} \sin ks ds \right],$$

$$h_N(t) = \frac{a_0}{2} + \frac{N}{2\pi} \sum_{k=1}^N \left[a_k \left\{ \frac{\sin k[t + (\pi/N)] - \sin k[t - (\pi/N)]}{k} \right\} - \right. \\ \left. - b_k \left\{ \frac{\cos k[t + (\pi/N)] - \cos k[t - (\pi/N)]}{k} \right\} \right].$$

Применяя тригонометрические формулы для разности двух синусов и двух косинусов, получаем

$$h_N(t) = \frac{a_0}{2} + \sum_{k=1}^N \sigma(N, k) [a_k \cos kt + b_k \sin kt],$$

где $\sigma(N, k)$ – называется *сигма-факторами*

$$\sigma(N, k) = \frac{\sin \pi k/N}{\pi k/N}.$$

Следовательно, сглаженный ряд Фурье есть исходный ряд Фурье с коэффициентами, умноженными на соответствующие сигма-факторы. Нетрудно заметить, что результирующая функция $h_N(t)$ – сглаженная кривая Ланцоша, имеет, как показано на рис. 3, немного ослабленные высокочастотные пульсации.

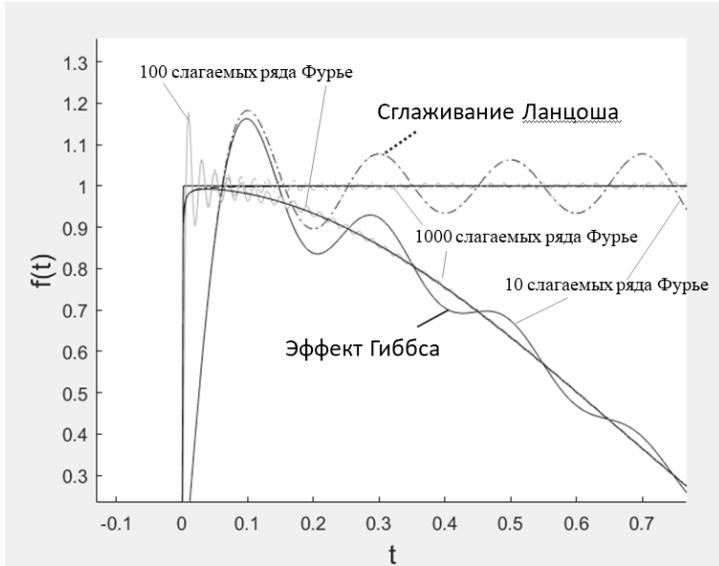


Рис. 3. Сглаживание Ланцоша

– Окно Хэмминга (см. рис. 4).

Это разновидность преобразования Фурье, определяемая следующим образом:

$$F(t, \varpi) = \int_{-\infty}^{\infty} f(\tau) W(\tau - t) e^{i\varpi\tau} ds,$$

где $W(s - t)$ – некоторая оконная функция. В случае дискретного преобразования оконная функция используется аналогично:

$$F(m, \varpi) = \sum_{n=-\infty}^{\infty} f[n] \varpi[n - m] e^{j\varpi n}.$$

Как видно из рис. 5 сглаживание Ланцоша работает более эффективно по сравнению с окном Хэмминга.

– Ядро Фейера

Сглаживание Фейера производит взвешивание коэффициентов ряда множителями

$$c(N, k) = (N - k) / N.$$

Таким образом, мы имеем сглаженный ряд Фейера

$$F_N(t) = \frac{a_0}{2} + \sum_{k=1}^N \frac{N - k}{N} [a_k \cos kt + b_k \sin kt].$$

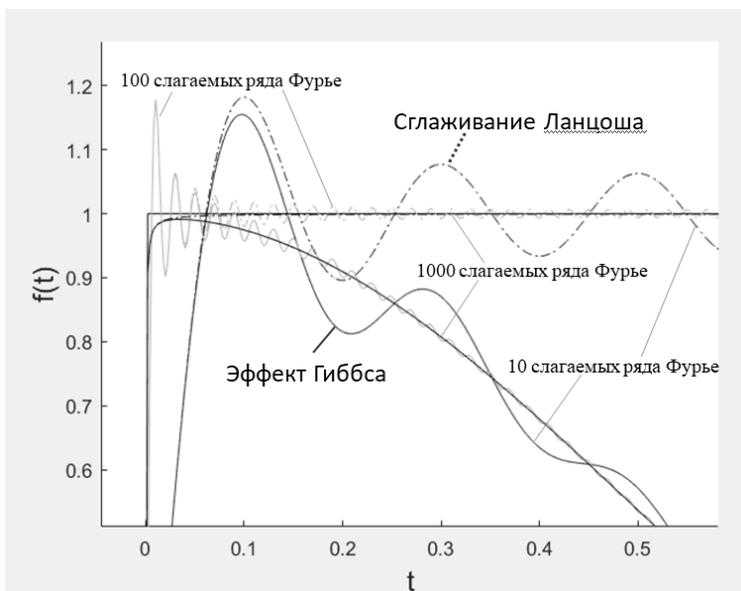


Рис. 4. Окно Хэмминга

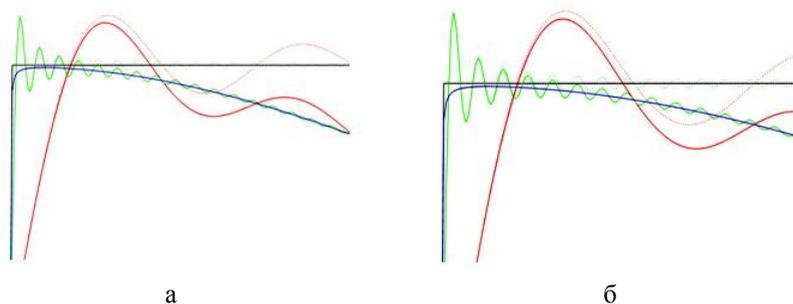


Рис. 5. Сравнение окна Хэмминга и сглаживания Ланцоша: а – окна Хэмминга, б – сглаживания Ланцоша

Сглаживание Фейера подтверждает, что время нарастания при сглаживании Фейера значительно больше, чем при сглаживании Ланцоша. Поэтому сглаживание Фейера редко применяется на практике. В пределе, когда $N \rightarrow \infty$, эта кривая обладает интересными математическими свойствами, однако при конечных значениях N она приближается к своему предельному значению слишком медленно.

Заключение

Данная статья посвящена исследованию эффекта Гиббса при восстановлении разрывных функций, а также поиску путей преодоления данного эффекта. В статье рассмотрено проявление эффекта Гиббса на примере разложения Фурье, реализованы алгоритмы, обеспечивающие подавление эффекта: сглаживание Ланцоша, окно Хэмминга, ядро Фейера и проведено их сравнение.

Список литературы

1. Поршнеv, С.В. Цифровая обработка изображений / С.В. Поршнеv. – М. : Телеком, 2003. – 592 с.
2. Гонсалес, Р. Цифровая обработка изображений в среде Matlab / Р. Гонсалес, Р. Вудс, С. Эминс. – М. : Техносфера, 2006. – 616 с.
3. Хемминг, Р. Цифровые фильтры / Р. Хемминг. – М. : Советское радио, 1980. – 224 с.

ИССЛЕДОВАНИЕ УГРОЗ ПОВЫШЕНИЯ ПРИВИЛЕГИЙ В ВЫЧИСЛИТЕЛЬНОЙ СЕТИ ВОРОНЕЖСКОГО ГОСУДАРСТВЕННОГО УНИВЕРСИТЕТА

М. А. Грибкова

Студент

С. А. Вялых

Доцент

Введение

Проблема защиты информационных систем (ИС) является одной из наиболее актуальных и глобальных задач информационных технологий. В настоящее время известно большое количество возможных уязвимостей и возможностей реализации угроз, и их число растет с каждым годом. Всё это выдвигает решение вопроса обеспечения безопасности на первый план.

До недавнего времени, основная защита ИС в зависимости от защищаемой информации сводилась к разработкетребуемого пакета документов, а требования по анализу уязвимостей защищаемых систем и оценке реальных возможностей реализации угроз безопасности, в том числе и угроз повышения привилегий, практически не выполнялись. Указанные обстоятельства определяют актуальность выбранной темы исследований.

1. Информационная сеть Воронежского государственного университета

Для проведения тестирования была выбрана информационная сеть факультета компьютерных наук Воронежского государственного университета. Данная сеть имеет в своем составе довольно большое число рабочих машин и является одной из самых современных в рамках высших учебных заведений.

Типовая структура сети включает в себя компьютерные классы с рабочими станциями и сетевое оборудование (маршрутизаторы, роутеры и так далее). Кроме того, компьютерами также обеспечены рабочие кабинеты сотрудников и административные помещения (деканат, аудитории кафедр). Основной операционной системой на всех машинах является OSWindowsXP, которая содержит стандартный пакет MicrosoftOffice, а также профильной программное обеспечение для разработки, необходимое в процессе обучения (VisualStudio, Matlab и другие) и виртуальные машины. За компьютерными классами осуществляется постоянный надзор – лаборатории оборудованы камерами видеонаблюдения и сигнализациями. В сети реализованы технологии разграничения уровней доступа пользователей, производится регулярное обновление программного обеспечения, используются межсетевые экраны для защиты от внешних угроз. Кроме того в системе действует парольная политика, подразумевающая регулярную смену пароля и регламентирующая требования к его необходимой сложности.

Для проведения испытания была составлена модель типового нарушителя информационной безопасности и проведена классификация угроз повышения привилегий.

2. Модель нарушителя информационной безопасности

Под понятием «модели нарушителя информационной безопасности» следует понимать совокупность предположений об одном или нескольких возможных нарушителях информационной безопасности, их мотивации, квалификационной подготовке, доступных технических, программных и аппаратных средств [1]. С точки зрения возможностей доступа нарушителей к информационной системе можно предложить следующую их классификацию (рис. 1).

Именно возможности доступа нарушителя определяют возможные методы и средства реализации угроз повышения привилегий.

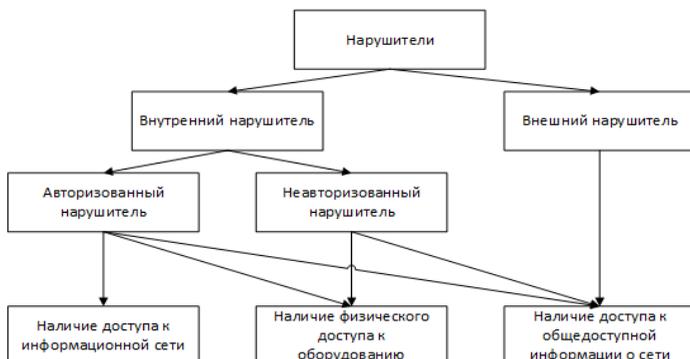


Рис. 1. Классификация нарушителей

3. Классификация существующих угроз повышения привилегий

Под угрозами безопасности информации следует понимать «...совокупность условий и факторов, создающих потенциальную или реально существующую опасность нарушения безопасности информации» [2].

В качестве основы для исследования был выбран Банк данных угроз безопасности информации, разработанный ФСТЭК России [2]. На данный момент в базе содержится описание 208 угроз (по данным на 24.05.2018), которые могут нести потенциальную опасность. Из указанных угроз были выбраны актуальные для информационных систем персональных данных Воронежской области [1] и проведена их классификация по возможным объектам воздействия (рис. 2).

Рассмотрим подробнее каждую категорию.

1. Угрозы BIOS

Вид нарушителя: внутренний.

Описание: угрозы доступа к системе BIOS (УБИ.018). В результате атаки, например, установке другой операционной системы с носителя, может быть получен доступ к системе, в том числе к защищенным политикой безопасности системным файлам, с правами администратора.

Способы противодействия: установка пароля на вход в BIOS, ограничение физического доступа к компьютерам.

2. Угрозы программного обеспечения

Вид нарушителя: внутренний, внешний.

Описание: угрозы внедрения вредоносного кода или данных (УБИ.006), использование информации и механизмов аутентификации (УБИ.008, УБИ.030), несанкционированный доступ к реестру (УБИ.089), учетным записям (УБИ.074, УБИ.090) и системным утилитам (УБИ.178) и так далее.

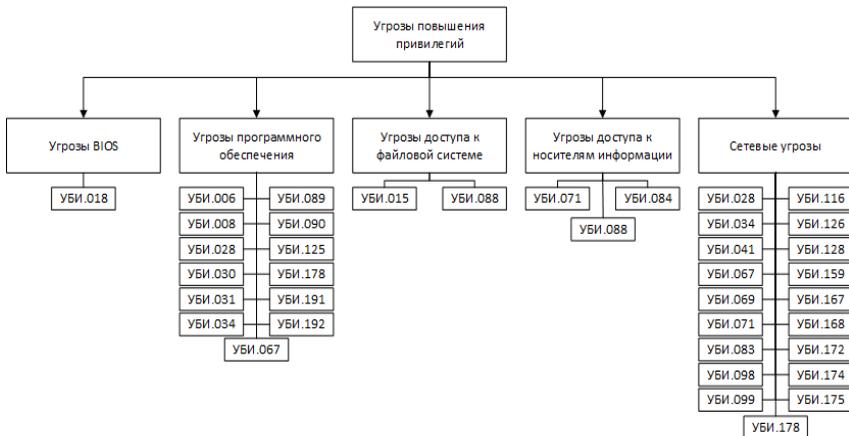


Рис. 2. Классификация актуальных угроз повышения привилегий

В качестве примера можно привести реализацию повышения привилегий за счет ошибок разграничения доступа в настройках ПО, позволяющих выполняющему операции от лица администратора.

Способы противодействия: установка актуальных обновлений ПО, использование только лицензионного ПО, запрет использования настроек по умолчанию, использование антивирусов, ограничение доступа, контроль на этапах разработки, внедрения и сопровождения ПО.

3. Угрозы доступа к файловой системе

Вид нарушителя: внутренний.

Описание: угрозы доступа к защищаемым файлам с использованием обходного пути (УБИ.015), использование альтернативных путей доступа к ресурсам (УБИ.028), несанкционированный доступ и копирование защищенных файлов (УБИ.088). Как пример можно привести копирование системного файла гибернации `hyberfil.sys`, с целью извлечения хэшей паролей пользователей.

Способы противодействия: парольная защита файлов, скрытие файлов от пользователей с низким уровнем доступа или их удаление, если это не вредит работоспособности системы, использование «сложных», с точки зрения возможности подбора, паролей.

4. Угрозы доступа к носителям информации

Виднарушителя: внутренний, внешний.

Описание: угрозы доступа и копирования защищенной информации, доступа к информации на виртуальных ресурсах (УБИ.084), утраты носителей (УБИ.156). В результате могут быть похищены данные, способные привести к реализации угроз повышения привилегий.

Способы противодействия: обеспечение физического контроля и парольной защиты носителей информации, криптозащита информации, запрет использования нештатных носителей.

5. Сетевые угрозы

Вид нарушителя: внутренний, внешний.

Описание: угрозы несанкционированного удаленного доступа к каналам связи (УБИ.083), открытым портам и хостам (УБИ.098, УБИ.099), перехвата данных, фишинга, фарминга, «почтовых червей» и так далее (УБИ.172, УБИ.174, УБИ.175). Зачастую, угрозы используются в качестве разведки для получения более полной информации о сети, ее топологии, пользователях и структуре. В результате атак может быть установлено вредоносное ПО, код или данные, похищена информация, в том числе, способная привести к реализации угроз повышения привилегий.

Способы противодействия: использование межсетевых экранов, антивирусов, криптозащита. Запрет использования файлов полученных из сети «Интернет», работа только на доверенных сайтах.

Результаты анализа возможностей реализации данных категорий угроз, позволяют утверждать, что Банк данных угроз безопасности информации, разработанный ФСТЭК России не охватывает весь спектр угроз, которые возможны применительно к рассматриваемым категориям.

4. Оценка возможностей реализации угроз повышения привилегий

При проведении исследований были выполнены попытки реализации нескольких различных угроз в рамках тестируемой информационной сети. При кажущемся постоянном контроле, студенты имеют практически неограниченный физический доступ к оборудованию и рабочим машинам.

При изучении компьютеров факультета компьютерных наук Воронежского государственного университета было выяснено, что практически на всех рабочих машинах, с которыми работают студенты, имеется доступ либо к аппаратному сбросу конфигурации BIOS, либо существующие настройки BIOS позволяют обойти загрузку штатных средств. Указанная уязвимость позволяет не только получить несанкционированный доступ к конфигурационным файлам операционной системы, программному обеспечению, но и позволила на один из компьютеров загрузить свою операционную систему в обход установленной политики безопасности - дистрибутив KaliLinux [3]. Указанная операционная система включает широкий спектр утилит предназначенных для проведения тестирования на проникновение и исследования возможностей повышения привилегий. Так с помощью утилиты nmap было проведено сканирование сети, показавшее, что она абсолютно незащищена от атак внутренних нарушителей: была получена топология сегмента сети (рис. 3), а также имелась возможность просмотра информации о каждом хосте (рис. 4).

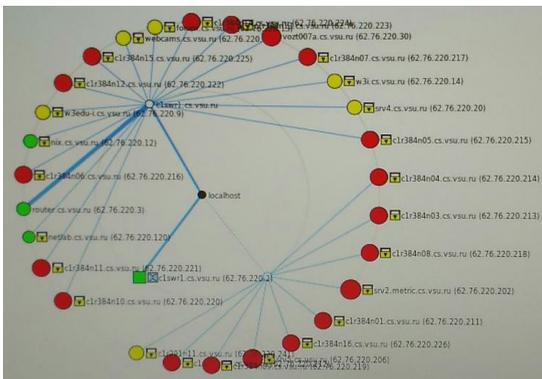


Рис. 3. Пример использования nmap для сканирования сети ФКН

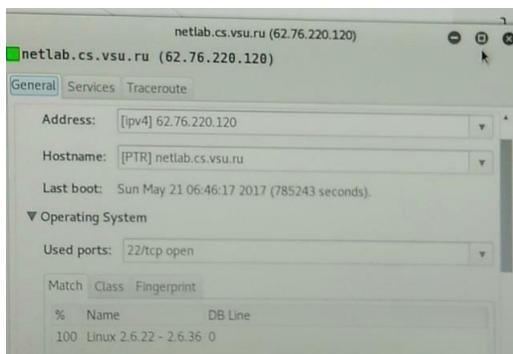


Рис. 4. Пример использования nmap для сканирования сети ФКН

Таким же образом был загружен дистрибутив OphCrack, позволивший извлечь данные и хэш пароля администратора (рис. 5). Хэш не был расшифрован в связи с нехваткой вычислительных ресурсов, однако угроза от этого не теряет своей опасности.

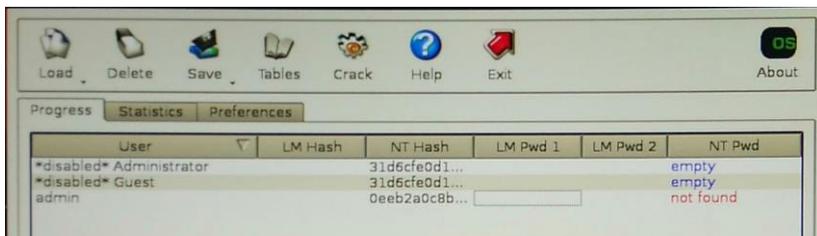


Рис. 5. Получение хэшей паролей

Кроме этого, установленные на компьютерах виртуальные машины, позволяют получить права администратора виртуальной машины, что позволяет использовать разные средства осуществлять различные атаки.

В процессе проведенных исследований была выявлена уязвимость парольной защиты, заключающаяся в том, что пара логин-пароль, используемая для авторизации в сети, также используется для доступа к сети Wi-Fi факультета и еще нескольким ресурсам, что увеличивает опасность хищения этих данных.

5. Рекомендации по защите

Анализ текущего состояния типового рабочего места пользователя компьютерной сети факультета компьютерных наук Воронежского государственного университета позволяет утверждать, что практически каждое из них является уязвимым для описанных выше атак.

В связи с этим в интересах парирования выявленных угроз безопасности информации рекомендуется обеспечить проведение следующих мероприятий по защите информации:

- ограничение свободного физического доступа к рабочим машинам и сетевому оборудованию;
- осуществление парольной защиты BIOS и контроль ее наличия для предотвращения возможности несанкционированных загрузок и установок сторонних операционных систем;
- ограничение возможности доступа к системным файлам, особенно к данным, содержащим хеши паролей, и, по возможности, их удаление;
- использование средств криптозащиты информации;
- сегментирование ИС и разграничение прав доступа к ее сегментам;
- защита сети от атак с использованием виртуальных машин;
- запрет использования одних и тех же паролей в различных ИС;
- регулярное проведение исследований сети на возможность проникновения и осуществления угроз повышения привилегий.

Заключение

Таким образом, результаты проведенных исследований показывают наличие уязвимостей безопасности информации компьютерной сети факультета компьютерных наук Воронежского государственного университета, которые могут быть реализованы внутренним нарушителем с различными правами доступа. В ряде случаев повышение привилегий тестируемой сети было осуществлено с применением общедоступных программных средств. На основе полученных результатов даны рекомендации по повышению эффективности защиты информации от выявленных угроз.

Список литературы

1. Об определении угроз безопасности персональных данных, актуальных при обработке персональных данных в информационных системах персональных данных правительства Воронежской области, исполнительных органов государственной власти Воронежской области и подведомственных им организаций [Электронный ресурс] : Постановление правительства Воронежской области от 02.11.2017 №869. – Режим доступа : <http://pravo.gov.ru/content/postanovlenie-pravitelstva-voronezhskoy-oblasti-ot-02112017-№869>
2. Банк данных угроз безопасности информации [Электронный ресурс]. – Режим доступа: <http://bdu.fstec.ru/threat>
3. Информационная безопасность и защита систем [Электронный ресурс]. – Режим доступа: <https://codeby.net>

МАГНИТОМЕТР В ТЕХНИКЕ И МЕДИЦИНЕ

А. В. Данилова

Студент

И. Г. Мещеряков

Студент

В. С. Господарикова

Студент

С. А. Зув

Доцент

Введение

Свойства магнитных полей используют во многих областях: в связи, электромашинных генераторах и электродвигателях, компасах, магнито-электрических приборах, магнитохимии, технике сверхвысокочастотного диапазона. Также в последнее время наблюдается их обширное внедрение в медицину. Магнитные поля используются для оздоровления и лечения организма человека. Очень чувствительна к магнитным полям сердечно-сосудистая система, при заболевании сердца улучшается коронарное кровообращение, снижается потребность миокарда в кислороде, повышается устойчивость организма к физической нагрузке. При воздействии на сосуды достигается их расширение, в крови понижается вязкость, снижается способность тромбоцитов образовывать тромбы в сосудах, улучшается местное кровообращение и доставка к тканям и

органам кислорода. Таким образом, магнитные поля оказывают противовоспалительное, болеутоляющее действие. Чтобы правильно подобрать необходимый для лечения прибор, необходимо знать, какая напряженность у магнитного поля, которое он излучает. Для этих целей используют магнитометры – приборы для измерения характеристик магнитного поля и магнитных свойств веществ (магнитных материалов).

1. Цель проекта

Мы поставили перед собой цель: создать магнитометр, который сможет обнаруживать и измерять сильные магнитные поля. Изучив различные виды магнитометров и принципы их работы, мы решили для достижения цели использовать аналоговый датчик Холла. В связи с этим мы выдвинули следующую задачу: с помощью аналогового датчика Холла обнаружить и измерить сильные магнитные поля напряженностью от 0,02 Тл до 0,1 Тл.

2. Изучение принципов работы датчика Холла

Магнитоизмерительным преобразователем в этом датчике является гальваномагнитный преобразователь Холла.

Принцип его работы основан на появлении ЭДС в проводящих средах, помещенных в зону влияния магнитного поля (эффект Холла) [1]. При этом вектор магнитной индукции искомого магнитного поля должен быть перпендикулярен пластине полупроводника. Устройство магнитометра с преобразователем Холла приведено на рис. 1. В устройстве был использован полупроводниковый датчик Холла типа 3503 в корпусе ТО-92УА.

Если через тело полупроводника протекает электрический ток I , то в результате на боковых гранях пластины образуется разность потенциалов, которую называют ЭДС Холла. После усиления, ЭДС определяется компенсационным методом или милливольтметром, шкала которого отградуирована в теслах.

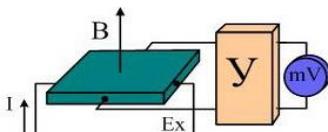


Рис. 1. Магнетометр с преобразователем Холла

На практике ЭДС Холла E_x зависит от следующих параметров:

$$E_x = C \cdot I \cdot B, \quad (1)$$

где C – коэффициент, учитывающий конструктивные параметры пластины полупроводника; I – сила тока, А; B – магнитная индукция, Тл.

Зная силу тока I , коэффициент C и значение E_x , прибор градуируют в единицах измерения магнитного поля, при условии, что сила тока фиксирована при каждом измерении.

Магнитометры с преобразователем Холла просты в применении, имеют небольшие размеры и малую потребляемую мощность. С их помощью определяют параметры постоянных, переменных и импульсных полей, применяя соответствующее математическое обеспечение.

3. Сборка магнитометра

Датчик Холла имеет три вывода [2] (два из них подаются на питание, третий – сигнальный), что упрощает процесс подключения. В итоге, для подключения датчика Холла нам потребовался сам датчик, плата Arduino pro mini, необходимая для управления собираемым прибором, 16×2-символьный ЖК-дисплей, батарейный отсек на 8 элементов типа АА, потенциометр 10кОм и соединительные провода. С помощью потенциометра можно менять яркость дисплея. Схема подключения приведена на рис. 2 [3].

Данные с датчика передаются на дисплей. В коде программы, которая загружается на плату, необходимо было дополнительно прописать операцию перевода единиц измерения из Гс в Тл, так как датчик Холла передает значения в Гс. Также необходимо произвести калибровку датчика. Для этого мы воспользовались следующим методом: завели переменную «а», в которую запомнили начальные значения, считанные с датчика Холла, далее необходимо вычитать «а» из новых показаний.

Для более удобного использования нашего прибора, как в лабораторных, так и полевых условиях, мы поместили собранную установку в специально предназначенный для этого корпус (рис. 3).

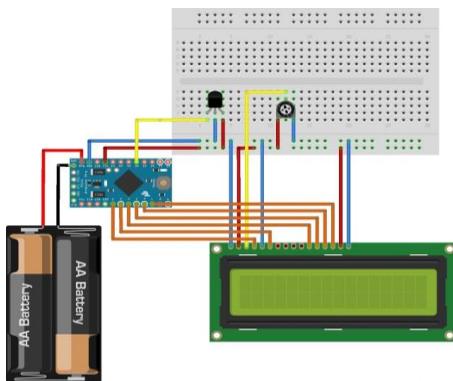


Рис. 2. Схема подключения



Рис. 3. Собранная установка в специальном корпусе

Заключение

Собранный нами прибор способен регистрировать магнитные поля, созданные сильными магнитами, также он обладает достаточно высокой чувствительностью. Его можно использовать на практике, в частности, в медицине для измерения магнитных полей приборов, предназначенных для лечения.

Список литературы

1. Измерение магнитной индукции и напряженности магнитного поля [Электронный ресурс] : статья в Интернете. – Режим доступа : <http://forum220.ru/measurement-magnetic-induction.php>
2. Датчики Холла [Электронный ресурс] : статья в Интернете. – Режим доступа : <http://robocraft.ru/blog/electronics/594.html>
3. Блум, Дж. Изучаем Arduino: инструменты и методы технического волшебства / Дж. Блум. – СПб. : БХВ-Петербург, 2017. – 336 с.

МАГНИТОМЕТР ДЛЯ ГЕОЛОГОРАЗВЕДКИ

А. В. Данилова

Студент

И. Г. Мещеряков

Студент

В. С. Господарикова

Студент

С. А. Зуев

Доцент

Введение

Свойства магнитных полей используются во многих областях, одной из которых является геологоразведка. Как известно, магнитное поле Земли для ограниченного участка местности однородно, поэтому любой ферромагнитный объект или объект, имеющий собственное магнитное поле, будет создавать локальную магнитную аномалию. Она, в свою очередь, может свидетельствовать о залежах того или иного полезного ископаемого. Для обнаружения и исследования таких аномалий используют магнитометры. Это специализированный, магниточувствительный прибор геофизической направленности, используемый для измерения характеристик магнитного поля и анализа различных магнитных аномалий (ферромагнетиков и пустот) в различных структурах грунта, включая грунт с высокой плотностью и минералосодержанием. Самым главным параметром магнитометра является его чувствительность. Её принято измерять величиной магнитной индукции поля, которое способен зарегистрировать прибор. Обычно чувствительность измеряют в нанотеслах (нТл). Для определения качества прибора также используют такой параметр, как разрешающая способность, которая тоже измеряется в нанотеслах и определяет ту минимальную разницу индукции, которую возможно зарегистрировать данным прибором.

1. Цель проекта

Мы поставили перед собой цель: создать магнитометр, который сможет обнаруживать локальные магнитные аномалии. Он должен обладать достаточно высокой чувствительностью. Проанализировав доступные нам ресурсы, мы поставили следующую задачу: измерить напряженность магнитного поля Земли с помощью датчика GY-273 HMC5883L, который дает показания с точностью до 1 мкТл.

2. Изучение принципов работы датчика, его калибровка

Датчик GY-273 HMC5883L измеряет магнитное поле в 3 осях, с точностью 1° - 2° [1]. Так как он работает по протоколу I2C, то сначала необходимо узнать его адрес. Для этого мы использовали специальную программу [2]. Процесс подключения датчика является нетрудоемким, потому что он уже содержит все необходимые элементы, такие как резисторы и конденсаторы. Сложность заключается в калибровке датчика, без которой его показания не достоверны. Можно выделить следующие факторы, приводящие к неправильной работе датчика: магнитное наклонение, искажения магнитного поля (hard iron и soft iron) [1] и разница между географическим и магнитным полюсами земли. Рассмотрим причины их возникновения.

В целом интенсивность магнитного поля неодинакова на всей планете и варьируется от 0.25Гс до 0.65Гс. Чтобы получить верные данные с датчика, необходимо ликвидировать Hard Iron искажения: при считывании данных необходимо увидеть смещение и увеличить/уменьшить данные на величину смещения (для простоты стоит визуализировать показания магнитометра).

Искажение типа soft iron дают ферромагнитные материалы недалеко от датчика, например, болты крепления, провода или опорная металлическая рама устройства. Чтобы избавиться от искажения, нужно умножить показания датчика по всем осям на коэффициент. Для того чтобы найти все коэффициенты (для X, Y и Z), необходимо выявить ось с наибольшей разностью между максимальным и минимальным значением, и затем воспользоваться формулой 1.

$$Y_{\text{scale}} = (A_{\text{max}} - A_{\text{min}}) / (Y_{\text{max}} - Y_{\text{min}}), \quad (1)$$

Географический и магнитный полюса на Земле сильно различаются. Для нивелирования различий необходимо прибавить (или вычесть) угол, называемым магнитным склонением. Для Воронежа магнитное склонение составляет $+8^{\circ}$, следовательно, получившиеся показания надо будет уменьшить на 8 градусов.

К одной из причин неправильной работы датчика также относится неперпендикулярность осей внутри корпуса микросхемы.

Для устранения этих неполадок необходимо рассчитать матрицы трансформации и смещения [3]. Мы использовали специально созданную программу MagMaster, которая на основе показаний датчика из двенадцати различных положений рассчитывает коэффициенты матрицы. На рис. 1 приведен пример работы программы, а на рис. 2 — специальная установка, на которой был закреплен датчик, чтобы было удобно фиксировать его различные положения [4].

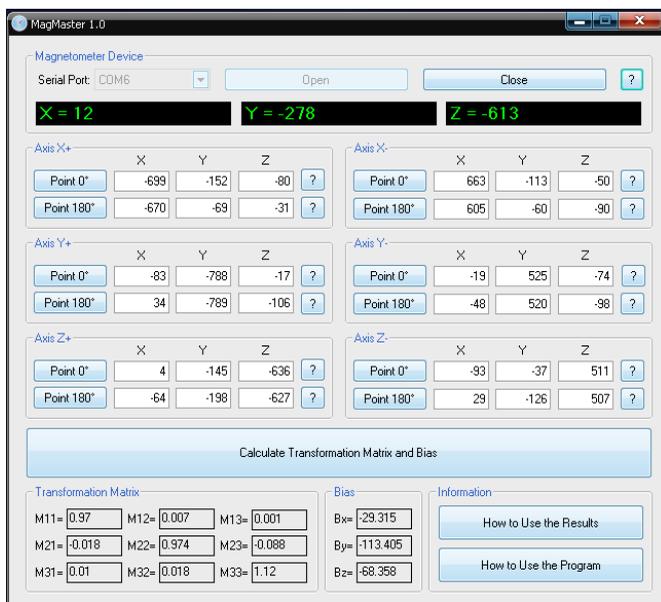


Рис. 1. Программа MagMaster

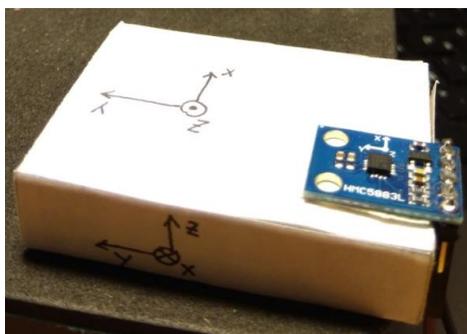


Рис. 2. Установка с датчиком GY-273 HMC5883L

Для того чтобы проверить правильность работы датчика GY-273 HMC5883L, мы запрограммировали его таким образом, чтобы он работал как компас, то есть сделали так, чтобы на экран выводился угол отклонения от севера. Для того чтобы его рассчитать, необходимо вычислить арктангенс от частного значений по оси Y и X. С помощью описанного метода мы смогли проверить правильность и точность работы датчика GY-273 HMC5883L. Следовательно, калибровка выполнена верно. Теперь прибор готов к практическому использованию.

3. Сборка магнитометра

Как отмечалось ранее, процесс подключения датчика GY-273 HMC5883L не является сложным и под силу даже студентам 1 курса. Все необходимые компоненты (плата Arduino pro mini, резистор 1кОм, 16×2-символьный ЖК-дисплей, батарейный отсек на 8 щелочных элементов типа AA фирмы Duracell, соединительные провода) мы собрали по схеме, приведенной на рис. 3.

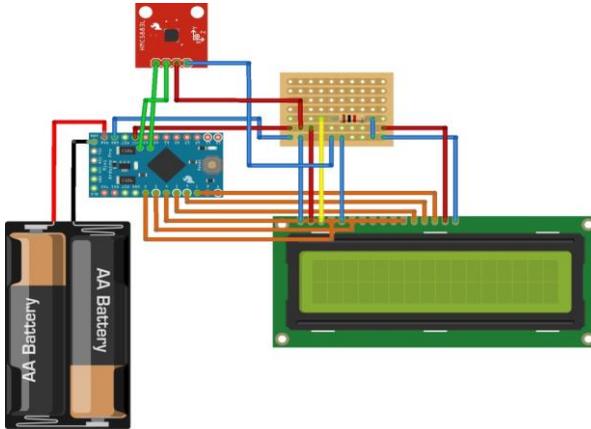


Рис. 3. Схема подключения

Далее, для более удобного использования нашего прибора на практике, мы поместили собранную установку в специально предназначенный для этого корпус (рис. 4 и 5).



вид изнутри

Рис. 4. Собранная установка



вид снаружи

Рис. 5. Собранная установка в специальном корпусе

Заключение

Собранный нами прибор способен регистрировать локальные магнитные аномалии, измеряя напряженность магнитного поля Земли. При его калибровке были учтены факторы, которые могут привести датчик к неправильной работе. Таким образом, этот прибор готов к практическому применению и может быть использован в геологоразведке.

Список литературы

1. Магнитометры: принципы действия и компенсация ошибок [Электронный ресурс] : статья в Интернете. – Режим доступа : <http://www.rlocman.ru/review/article.html?di=143960>
2. I2C_scanner [Электронный ресурс] : статья в Интернете. – Режим доступа : <https://playground.arduino.cc/Main/I2cScanner>
3. Калибровка типовых магнитометров на примере НМС5883L [Электронный ресурс] : статья в Интернете. – Режим доступа : http://earchive.tpu.ru/bitstream/11683/36934/1/conference_tpu-2016-C04_V1_p345-346.pdf
4. Advanced hard and soft iron magnetometer calibration for dummies [Электронный ресурс] : статья в Интернете. – Режим доступа : <https://diydrone.com/profiles/blogs/advanced-hard-and-soft-iron-magnetometer-calibration-for-dummies>

СИСТЕМА ОЦЕНКИ ИНФОРМАЦИОННОЙ ЗАЩИЩЕННОСТИ КОМПАНИИ НА ОСНОВЕ ИНТЕГРАЦИИ РАЗНОРОДНЫХ ДАННЫХ И ПОСТРОЕНИЯ ВЫВОДОВ ПО ОНТОЛОГИИ СТАНДАРТА ISO/IEC 27001

А. Ю. Данковцева

Студент

В. В. Гаршина

Доцент

Введение

Одной из наиболее актуальных сфер применения интеллектуальных систем обработки информации и поддержки принятия решений является область информационной безопасности. Построение программных систем анализа информационной защищенности связано с обработкой разнородной информации (данных), которые обладают следующими особенностями: могут поступать из разных источников (технические средства, экспертные мнения, результаты методик расчета, хранилища данных, результаты измерений и т.д.); данные могут обладать неполнотой, нечеткостью и неточностью; иметь разный уровень надежности; вступать в противоречие друг с другом; быть статическими и динамическими; быть объективными (результаты измерений) и субъективными (результаты экспертного оценивания).

В связи с этим, за основу модели знаний предметной области информационной защищенности компании, в которую будет сводиться вся исходная для аналитики информация, предлагается использовать онтологию. Она будет наполняться данными разного типа и из разных источников, условно назовем их – объективные и субъективные. Объективные поступают из надежных источников, являются результатами измерений и оценок, обладающих свойствами точности, полноты, четкости (однозначной интерпретации). Субъективные – данные базирующиеся на экспертных оценках квалифицированных специалистов в данной области, требуют дополнительной обработки, согласований мнений и т.д.

Желаемым результатом работы системы является не построение количественных оценок безопасности анализируемой системы, которые не дают информации о причинах и наиболее значимых параметрах, которые привели к итоговому результату, а построение смыслового (семантического) анализа, основанного на анализе процессов в предметной области.

1. Онтология стандарта ISO/IEC 27001

Для реализации данного подхода была выбрана онтология, основанная на стандарте по информационной безопасности ISO/IEC 27001 [1]. Данный стандарт используется в качестве модели для разработки, внедрения, функционирования, мониторинга, анализа, поддержки и улучшения системы менеджмента в области информационной безопасности. Данная онтология разработана в Линчёпингском университете Швеции [2]. Она предоставляет возможность анализа и оценки рисков и уязвимостей, что необходимо для решения задач:

1. Оценки ущерба для системы, который может быть нанесен в результате сбоя обеспечения безопасности, с учетом возможных последствий нарушения конфиденциальности, целостности или доступности;

2. Оценки реальной вероятности сбоя обеспечения безопасности с учетом угроз, уязвимостей и их последствий;

3. Оценки уровня рисков.

Для работы была выбрана ветвь онтологии, которая включает информацию о различных рисках и угрозах (рис. 1).

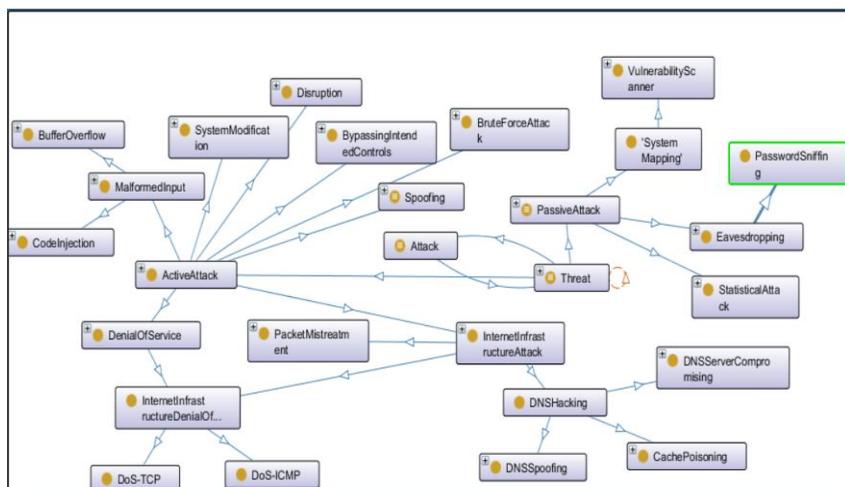


Рис. 1. Используемая ветвь онтологии

Используемая ветвь предоставляет возможность оценки 58 угроз, самыми распространёнными можно выделить следующие: System Modification; DNS Spoofing; IP Address Spoofing; Cache Poisoning; Trojan Horse; SQL Injection; Stack Overflow; DNS Hacking; DoS-TCP; Stack Attack; Password Sniffing.

2. Архитектура программной системы анализа информационной защищенности компании на основе онтологии стандарта ISO/IEC 27001

Основной программный модуль системы реализована на языке Java. Архитектура информационной системы представлен на рис. 2. На вход системы поступают Экспертная оценка и/или Объективные данные измерений. Экспертные оценки – это результаты опроса экспертов в виде матрицы парных сравнений, что реализовано в самом Программном модуле. Для хранения полученных результатов выбрана база данных PostgreSQL (взаимодействие с java-кодом через библиотеку jdbc). Для доступа к owl-файлам, который осуществляется с помощью SPARQL-запросов, используется библиотека jena. Вывод онтологии строится на основе запросов SPARQL и правил SWRL, которые отсутствовали в исходной онтологии, поэтому требовалась самостоятельная разработка.



Рис. 2. Архитектура программной системы анализа информационной защищенности компании на основе онтологии

3. Механизмы логического вывода на онтологии

Механизмы логического вывода [3] позволяют вычислять значения логических выражений, проверять правильность онтологии и автоматически помещать в нее новую информацию в соответствии с правилами, позволяют оперировать именами классов, свойств и сущностей, и «задавать модели вопросы», абстрагируя пользователя от подробностей внутреннего строения модели. Осуществляется с помощью SPARQL. На рис. 3 показан пример SPARQL-запроса используемого в java-коде для получения факторов используемой ветви.

```
String queryString =
    "prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#\n" +
    "select ?subclass where {\n" +
    "  ?subclass rdfs:subClassOf" +
    "<http://www.ida.liu.se/~iislab/projects/secont/Security.owl#Threat> + "\n" +
    "};";
```

Рис. 3. Пример SPARQL-запроса

Также существует возможность определять собственные правила, в соответствии с которыми должны получаться логические выводы. Такая технология называется SWRL. Каждое правило состоит из двух частей – условия и вывода, который формируется, если условие выполнено. И условие, и вывод могут состоять из нескольких атомов – элементарных логических выражений. Каждый атом представляет собой предикат – утверждение о каких-либо объектах онтологии. Правила SWRL позволяют создавать весьма гибкие условия для получения новых знаний. На основе таких правил строится вывод о защищенности системы.

4. Обработка экспертных мнений на основе матрицы парных сравнений

Задача обработки и интеграции субъективных экспертных мнений для внесения в онтологию связана с решением ряда задач: определение компетентности экспертов; построение обобщенной ранжировки объектов каждым экспертом; определение согласованности мнений экспертов; определение зависимости между ранжировками. На рис. 4 показан интерфейс разработанной программы для оценки субъективных экспертных мнений.

В работе обработка экспертных мнений проводится на основе метода парных сравнений [4]. Таким образом, формируется общая оценка каждого объекта (рис. 5).

Значение элемента, стоящего на пересечении i -й строки и j -го столбца, определяется по формуле:

$$a_{ij} = \begin{cases} 0, & A_i < A_j, \\ 1, & A_i \cong A_j, \\ 2, & A_i > A_j. \end{cases}$$

Метод вычисления весовых коэффициентов для каждого объекта представляет собой итерационную процедуру:

$$p^i = Ap^{i-1},$$

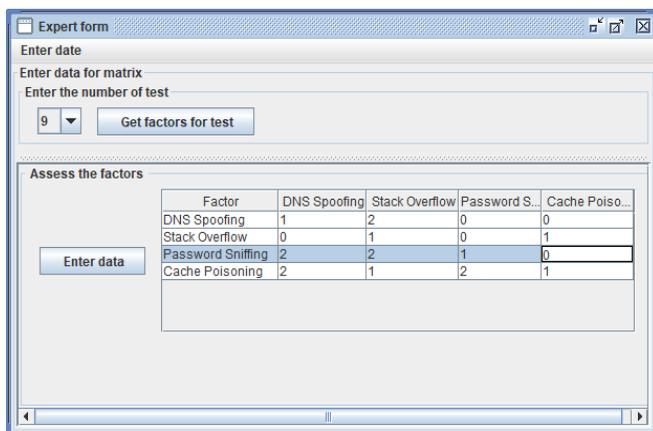


Рис. 4. Интерфейс программной реализации ввода экспертами данных для матрицы парных сравнений

Объекты	A_1	A_2	A_n
A_1	a_{11}	a_{12}	a_{1n}
A_2	a_{21}	a_{22}	a_{2n}
.
.
.
A_n	a_{n1}	a_{n2}	a_{nn}

Рис. 5. Матрица парных сравнений

Компоненты вектора нормируются на каждом шаге путем деления на сумму:

$$\lambda^t = \sum_i \sum_j a_{ij} p_j^{t-1} = \sum_i a_i p_i^{t-1}.$$

С учетом нормирующего множителя процедура вычисления весовых коэффициентов записывается следующим образом:

$$p^t = \frac{1}{\lambda^t} A p^{t-1}.$$

Вычисление продолжается до тех пор, пока для соседних двух итераций не будет выполнено:

$$\max |p^t - p^{t-1}| < \varepsilon,$$

где ε – достаточно малое положительное число, задающее точность расчетов.

В случае, если оцениваемые критерии защищенности системы и угрозы риска поступают от группы экспертов, их необходимо свести к единой итоговой оценке (рис. 6).

Объекты	Эксперты			
	Ξ_1	Ξ_2	Ξ_m
A_1	p_{11}	p_{12}	p_{1m}
A_2	p_{21}	p_{22}	p_{2m}
.	.	.		.
.
.	.	.		.
A_n	p_{n1}	p_{n2}	p_{nm}

Рис. 6. Результаты опроса группы экспертов

Для этого необходимо вычислить:

$$P^t = \frac{1}{\lambda^t} P P^t P^{t-1},$$

где P – прямоугольная $m \times n$ матрица с элементами p_{ij} , представляющими собой оценки i -го объекта j -м экспертом.

При оценке объектов эксперты обычно расходятся во мнениях по решаемому вопросу. Важно понять, насколько велико это различие. Если мало - усреднение мнений экспертов позволит выделить то общее, что есть у всех экспертов, отбросив случайные отклонения в ту или иную сторону. Если велико – усреднение является чисто формальной процедурой. В последнем случае усреднение приведет к результату, которого не придерживается ни один эксперт. Следовательно, проблема проверки согласованности мнений экспертов, является важной. В программе реализован ряд методов такой проверки:

- с использованием коэффициентов конкордации [5]:

$$W = \frac{12S}{m^2(n^3 - n)},$$

где
$$S = \sum_{i=1}^n \left(\sum_{j=1}^m p_{ij} - \bar{p} \right)^2;$$

- с использованием коэффициентов Спирмена [6]:

$$\tau = \frac{2}{n(n-1)} \sum_{i=1}^n \sum_{j=1}^n \text{sign}[(p_{i1} - p_{j1})(p_{i2} - p_{j2})].$$

Заключение

Разработано приложение, позволяющее строить выводы о защищенности анализируемой системы на основе онтологии. Данные от группы экспертов обрабатываются методом матрицы парных сравнений, и сводятся к единому мнению с помощью итерационных процедур. Итоговая оценка поступает на вход онтологии, основанной на стандарте информационной безопасности ISO/IEC 27001. Разработанное приложение предоставляет качественный анализ безопасности анализируемой системы с построением обоснованного вывода мнения о защищенности информационной системы организации. Использование метода экспертных оценок помогает формализовать процедуры сбора, обобщения и анализа мнений специалистов с целью преобразования их в форму, наиболее удобную для принятия обоснованного решения.

Список литературы

1. Документация стандарта по информационной безопасности ISO/IEC [Электронный ресурс]. – Режим доступа : 27001https://docviewer.yandex.ru/view/166366363/?*=BF2WqK%2FONuvJpwE%2BroP96gB4QAp7InVybcI6InlhLW1haWw6Ly8xNjU3ODg3NjEyODI1O DE0MzIvMS4yIiwidGl0bGUiOiIxMy5HT1NlUX1JfSVNpX01FS18yNzAwMS0yMDA2LnBkZiIsInVpZCI6IjE2NjM2NjM2MyIsInl1IjoiMjM0MjM5OTU2MTQ1MzA0ODUyNSIsIm5vaWZyYW11IjpmYWxzZSwidHMiojE1MjcyODc1MDE4NDJ9
2. Security Ontology [Электронный ресурс] : база данных. – Режим доступа : <https://www.ida.liu.se/divisions/adit/security/projects/secont/>
3. Документация онтологического моделирования [Электронный ресурс] : база данных. – Режим доступа : https://docviewer.yandex.ru/view/166366363/?*=1NNQ9QkItpV46ax1cmv0RdrRhvJ7InVybcI6InlhLW1haWw6Ly8xNjU1MDcyODYzMDU4NzA2NzUvMS4yIiwidGl0bGUiOiJlZlJlbnRpY0ludHJvLnBkZiIsInVpZCI6IjE2NjM2NjM2MyIsInl1IjoiMjM0MjM5OTU2MTQ1MzA0ODUyNSIsIm5vaWZyYW11IjpmYWxzZSwidHMiojE1MjcyODc1MDE4NDJ9
4. Дэвид Г. Метод парных сравнений / Г. Дэвид. – М. : Статистика, 1978. – 144 с.
5. Большев Л.Н. Таблицы математической статистики / Л.Н. Большев, Н.В. Смирнов. – М. : Наука, 1983. – 416 с.

6. Кендэл М. Ранговые корреляции / М. Кендэл. – М. : Статистика, 1975. – 216 с.

РАЗРАБОТКА БИОМЕТРИЧЕСКОЙ СИСТЕМЫ ДЛЯ АУТЕНТИФИКАЦИИ ПОЛЬЗОВАТЕЛЯ ПО КЛАВИАТУРНОМУ ПОЧЕРКУ

Е. С. Данковцева

Студент

Е. Ю. Митрофанова

Доцент

Введение

Авторизация и аутентификация являются неотъемлемой частью нашей жизни. И существует несколько традиционных способов аутентификации личности: по знаниям, по собственности, по биометрическим параметрам. Но использование только какой-либо собственности или секретной информации не является абсолютной гарантией того, что посторонний не сможет получить конфиденциальную информацию или чужую собственность.

Из-за сложности имитации биометрических параметров применение биометрии представляется перспективным. Использование биометрии, например, в социальных сетях, мобильных приложениях, может повысить надежность идентификации, а также повысить доверие пользователей.

В связи с этим задача создания системы, идентифицирующей человека по клавиатурному почерку представляется актуальной. Использование подобных продуктов должно повысить надежность систем, но при этом не возникает необходимости покупки какого-либо дополнительного оборудования [1, 2].

1. Клавиатурный почерк

Для входа в систему пользователь вводит определенную парольную фразу. При этом основными характеристиками клавиатурного стиля являются время нажатия каждой клавиши и время между нажатиями клавиш [3, 4]. Также возможны перекрытия по времени, если клавиши находятся рядом. Тогда интервал между нажатием клавиш будет отрицательным.

Так на рис. 1 представлена диаграмма набора парольной фразы. На оси абсцисс отмечено время нажатия и отпускания клавиш в миллисекундах.

© Данковцева Е. С., Митрофанова Е. Ю., 2018

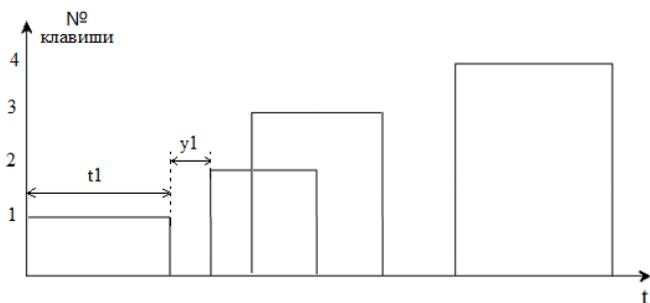


Рис. 1. Временная диаграмма набора парольной фразы

На основе замеренных временных характеристик строится периодическая функция с периодом равным времени набора парольной фразы. На рис. 2 мы видим график периодической функции $F(x)$. На оси абсцисс отмечено время в миллисекундах, а на оси ординат располагается значение функции в момент времени t .

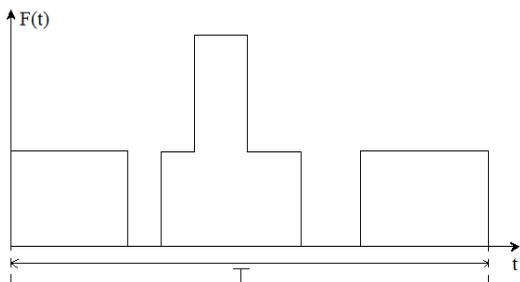


Рис. 2. Функция $F(t)$

На рис. 1 видно, что произошло перекрытие по времени между клавишей два и клавишей три. Поэтому на рис. 2 на промежутке перекрытия функция имеет значение вдвое большее, чем значение функции при обычном нажатии клавиши.

2. Аутентификация пользователя на основе измерения близости образа к биометрическому эталону мерой Хэмминга

Расстояние Хэмминга E_v – это неотрицательное число, которое меняется в интервале от 0 до n . Определяется расстояние Хэмминга, как общее число несовпадений с биометрическим эталоном.

Максимальное и минимальное значение для параметра v_i вычисляется, как:

$$\min(v_i) = m(v_i) - C[L, (1 - P_i)] \times \sigma(v_i); \quad (1)$$

$$\max(v_i) = m(v_i) + C[L, (1 - P_i)] \times \sigma(v_i), \quad (2)$$

где L – число использованных примеров при обучении, $C[L, (1 - P_i)]$ – коэффициенты Стьюдента, P_i – заданное значение вероятности ошибок первого рода.

При использовании достаточно большого числа примеров распределение значений меры Хэмминга близко к нормальному. Тогда можно вычислить пороговую величину меры Хэмминга E_n :

$$E_n = m(E_v) + C[L, (1 - P_i)] \times \sigma(E_v). \quad (3)$$

Тогда для авторизации пользователя полученное для него значение меры Хэмминга E_v будет сравниваться с пороговой величиной E_n . Если значение E_v будет не больше E_n тогда система примет решение, что авторизация пройдена успешно. В другом случае, будет принято решение об отказе пользователю в авторизации [5].

3. Аутентификация пользователя на основе контроля попадания в область распределения эталонных образцов

Будем считать, что распределение вектора V в N -мерном пространстве близко к нормальному, а векторы лежат внутри N -мерной области, которая при $L \rightarrow \infty$ описывается гиперэллипсоидом рассеивания в ортогональной системе координат.

Функция плотности распределения имеет вид:

$$f(v_1, \dots, v_n) = \frac{1}{\sqrt{(2\pi)^n \det_j(\alpha_{jk})}} \exp\left[-\frac{1}{2} \sum_{j=1}^n \sum_{k=1}^n \beta_{jk} (v_j - \xi_j)(v_k - \xi_k)\right], \quad (4)$$

$$\alpha_{jk} = \alpha_{kj} = M(v_k - \xi_k)(v_j - \xi_j). \quad (5)$$

Коэффициенты α_{jk} составляют корреляционную матрицу $[\alpha]$, а коэффициенты β_{jk} составляют матрицу $[\beta]$, обратную корреляционной матрице.

Гиперэллипсоид рассеивания имеет равную плотность распределения N -мерных случайных величин, поэтому выражение для него можно получить из условия $f(v_1, v_2, \dots, v_n) = \text{const}$. Данное условие может выполняться, если:

$$\frac{1}{2} \sum_{j=1}^n \sum_{k=1}^n \beta_{jk} (v_j - \xi_j)(v_k - \xi_k) = 1. \quad (6)$$

Так как существует вероятность того, что образец, предъявленный самим же пользователем при аутентификации, выйдет за пределы зафиксированного в эталоне диапазона, то для такого случая

дополнительно задается величина в виде коэффициента Стьюдента $C[L, (1 - P_i)]$.

Теперь, зная уравнение, мы можем проводить авторизацию пользователя. Для этого необходимо выполнение следующего условия для вектора V :

$$\frac{1}{2} \sum_{j=1}^n \sum_{k=1}^n \beta_{jk} (v_j - \xi_j)(v_k - \xi_k) = 1. \quad (7)$$

$$\frac{1}{2} \sum_{j=1}^n \sum_{k=1}^n \beta_{jk} (v_j - \xi_j)(v_k - \xi_k) = C[L, (1 - P_i)]^2. \quad (8)$$

Если данное неравенство выполняется для вектора V , то считается, что этот вектор принадлежит зарегистрированному ранее пользователю, и ему доступ разрешается.

4. Реализация

В качестве средств реализации использовались:

- Microsoft Visual Studio Enterprise 2017;
- Visio Professional 2016.

На рис. 3 представлена диаграмма вариантов использования, которая отражает возможные действия пользователя в системе.

Если пользователь не был ранее зарегистрирован, то он может пройти регистрацию, введя логин и несколько раз пароль. При этом у пользователя нет возможности вставить символы через комбинацию клавиш $\text{Ctrl}+V$ и через правую кнопку мыши. Также пользователь может менять настройки регистрации, а именно выбрать режим работы и использовать настройку для получения пароля.

Пользователь может выбрать один из трех режимов: низкий, средний, высокий. В зависимости от выбранного режима меняется пороговая величина, определяющая, какая из гипотез является верной, а так же количество векторов, которые необходимо получить от пользователя для успешной регистрации.

Если пользователь уже зарегистрировался, то он может попробовать авторизоваться в системе. После ввода логина и пароля, сначала система проверит, есть ли в базе данных пользователь, с такими данными. Если такого пользователя нет, то система выведет об этом сообщение. Если существует такой пользователь, то система начнет процесс верификации. Для этого сначала будет создан вектор биометрических параметров, а после будет использован один из методов аутентификации. Далее будет выведено сообщение с информацией об успешности операции.

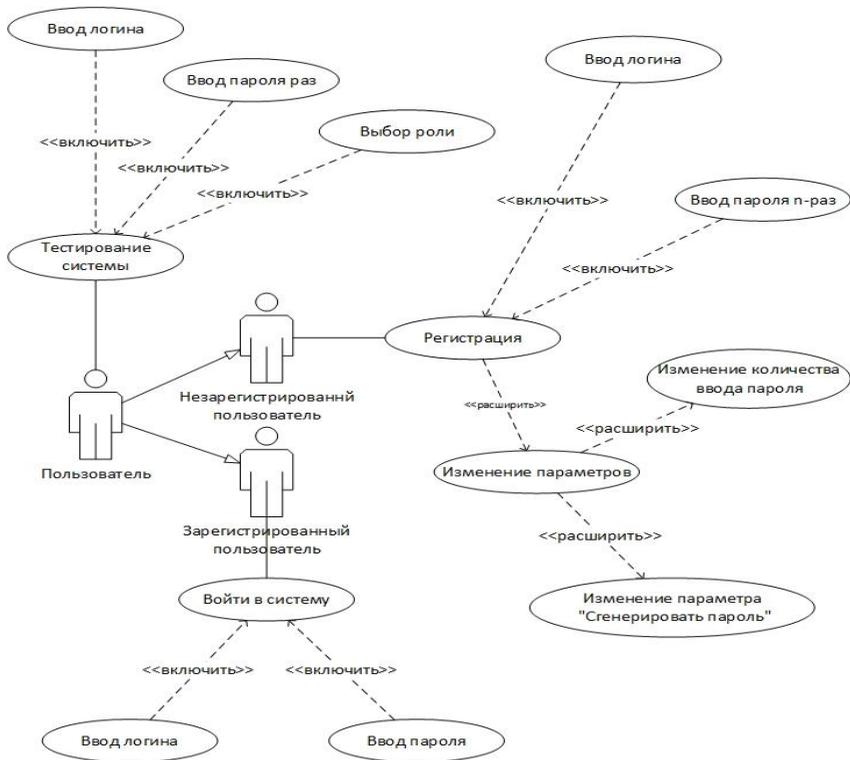


Рис. 3. Диаграмма вариантов использования

Заключение

Разработано приложение, которое при авторизации пользователя использует не только логин и пароль, но и клавиатурный почерк человека. При вводе пароля формируется вектор биометрических параметров клавиатурного почерка, основанный на получении дискретной функции от первоначальных данных таких, как время нажатия и удерживания клавиш. Данная функция раскладывается в ряд по функциям Хаара. Члены данного ряда, полученные при разложении функции, являются компонентами вектора биометрических параметров.

Следующие методы были подробно рассмотрены и реализованы:

- на основе измерения близости образа к биометрическому эталону мерой Хэмминга;
- на основе контроля попадания в область распределения эталонных образцов.

Список литературы

1. Лакин Г. Ф. Биометрия : Учебное пособие для биол. спец. вузов – 4-е изд., перераб. и доп. – М. : Высшая школа, 1990. –352 с.
2. Руководство по биометрии / Р. М. Болл, Дж. Х. Коннел, Ш. Панканти [и др.]; пер. с англ. Н. Е. Агаповой. – М. : Техносфера, 2007. – 368 с.
3. Lee H. C. Advances in fingerprint technology. Second edition / H. C. Lee, R. E. Gaensslen. – Boca Raton: CRC Press, 2001. – 425 с.
4. Иванов А.И. Биометрическая идентификация личности по динамике подсознательных движений / А.И. Иванов. – Пенза : ПГУ, 2000. – 186 с.
5. Брюхомицкий Ю. А. Исследование биометрических систем динамической аутентификации пользователей ПК по рукописному и клавиатурному почеркам / Ю. А. Брюхомицкий, М. Н. Казарин. – Таганрог : ТРТУ, 2004. – 38 с.

АДАПТИВНЫЙ АЛГОРИТМ РАЗДЕЛЕНИЯ СОПРИКАСАЮЩИХСЯ ЗЕРЕН НА ЦИФРОВОМ ИЗОБРАЖЕНИИ

Е. А. Добрынин

Студент

А. А. Крыловецкий

Доцент

Введение

Определение типа зерна, очищение его от посторонних примесей и некачественных образцов – необходимые этапы, предшествующие дальнейшей обработке в современной пищевой промышленности. Среди применяемых очистительных (сортирующих) устройств - сепараторы механического, аэродинамического типов и фотосепараторы. Фотосепаратор позволяет выбраковывать трудноотделимые примеси, которые очень сложно или невозможно убрать с помощью машин иного класса. В работе фотосепаратор использует фотографии зерна, проходящего через распределительный лоток, получаемые с помощью камер различного спектра. Таким образом становится возможным применение методов цифровой обработки изображений, позволяя разрабатывать сложные и действенные алгоритмы сортировки. Однако не всегда удается добиться раздельного движения объектов вдоль

распределительного лотка, обеспечить между их границами расстояние, необходимое для однозначного выделения объектов, что сильно препятствует сортировке.

В литературе описывается несколько методов для отделения «слипшихся» зерен друг от друга, которые будут рассмотрены более подробно далее. Между тем эти методы не очень хорошо показывают себя при работе в таких условиях, как большая площадь соприкосновения, множество одновременно соприкасающихся объектов и негладкие контуры. Подобные ситуации неизбежны на реальном производстве и потому верная их обработка – обязательное свойство алгоритма, пригодного для работы в составе программного обеспечения фотосепаратора. В статье описывается новый комплексный подход к разделению «слипшихся» зерен на основе существующих алгоритмов.

1. Существующие методы разделения зерен

1.1 Последовательное применение операций эрозии и дилатации

В [1] рассматривается алгоритм сегментации объектов, основанный на применении базовых морфологических операций – эрозии и дилатации - на бинарном изображении.

Описание алгоритма:

1. К бинарному изображению соприкасающихся зерен (пиксели со значением яркости 0 обозначают фон, пиксели с другим значением - объекты) несколько раз применяется эрозия с постепенно увеличивающимся структурообразующим множеством (ядром), пока не произойдет разделение всех элементов.

2. Эрозия разделила зерна, но при этом сильно исказила их. Поэтому далее применяется дилатация для расширения объектов и заполнения маленьких отверстий в них, не допуская повторного соединения зерен.

Метод тестировался исследователями на образцах зерна нескольких классов, полученные результаты приведены в табл. 1.

Таблица 1

Результаты разделения соприкасающихся зерен различных культур с помощью метода на основе эрозии и дилатации

Зерновая культура	Точность разделения, %
Твердая яровая пшеница	95
Твердая пшеница	95
Ячмень	94
Овес	79
Рожь	89

Демонстрируя хорошую точность разделения для пшеницы и ячменя, заметно хуже алгоритм справился с овсом и рожью. В заключении работы [1] отмечается низкая эффективность алгоритма на длинных кластерах объектов и на кластерах, содержащих длинные, узкие перешейки.

1.2 Сегментация по морфологическим водоразделам

Понятие водораздела основано на представлении изображения как трехмерной поверхности, где в качестве высоты используется уровень яркости пикселя. В этом случае на поверхности можно обнаружить три типа точек:

- точки локального минимума,
- точки, находящиеся на склоне, с которых вода сливается к центру водоема,
- точки, находящиеся на гребне возвышенности.

Линии, образованные точками-гребнями, представляют собой линии водоразделов, поэтому основной задачей данного метода является именно поиск линий водоразделов.

Описание алгоритма (рис. 1):

1. В местах локального минимума образуем отверстия, через которые вода начнет заполнять трехмерную поверхность.
2. Если вода с двух сторон гребня готова объединиться в один бассейн, устанавливаем перегородку.
3. Когда над водой останутся только перегородки, останавливаем алгоритм. Полученные таким образом перегородки и есть требуемые линии водоразделов.

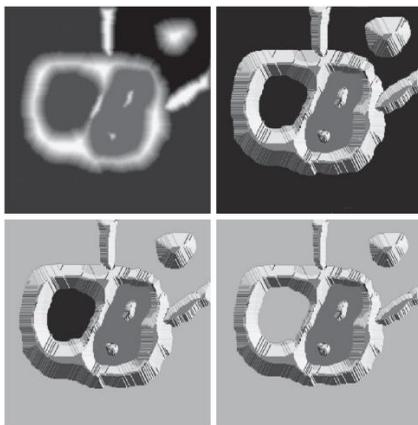


Рис. 1. Демонстрация алгоритма сегментации по морфологическим водоразделам

Алгоритм склонен к чрезмерной сегментации изображения и, как в случае с эрозией-дилатацией, не в состоянии разделить длинные соприкасающиеся зерна и кластеры, имеющие длинные, узкие перешейки между объектами, входящими в их состав.

1.3 Разделение на основе сглаживания границ объекта с помощью эллиптического ряда Фурье и последующего анализа их кривизны

Описание алгоритма, предложенного в работах [2, 3]:

1. Контур объекта представляется в виде цепного кода (chain code). Таким образом достигается инвариантность к размеру объектов, сохраняя лишь описания формы.

2. Аппроксимация исходного контура эллиптическим рядом Фурье на основе полученного кода может быть проведена с использованием формул, определенных в [4, 5].

3. В каждой точке восстановленного контура вычисляется кривизна. Далее на основе нее, с использованием некоторого порогового значения ищутся узловые точки – они будут кандидатами на вершины разделяющих линий.

4. Строится множество потенциальных разделяющих линий, равное декартовому произведению множества узловых точек самого на себя. Из множества пар точек получается упорядоченный по расстоянию между узлами список.

5. Первая пара точек этого списка определяет первую разделяющую линию. При дальнейшем прохождении по списку из него удаляются пары, содержащие хотя бы одну точку, принадлежащую уже определенным разделяющим линиям, создавая таким образом новые.

На рис. 2 изображен пример удачного разделения. Видно, что алгоритм справляется со случаями множественного соприкосновения и кластерами, имеющими длинные, узкие перешейки. Однако из-за того, что метод сильно полагается на вычисленные значения кривизны, сравнивая их с некоторым пороговым, возможны случаи, когда кривизна в точке лишь немного недостает до пороговой, в то время как сама точка явно составляет пару с другой, формируя разделяющую линию.

2. Предлагаемый адаптивный алгоритм

Ограничения рассмотренных методов мешают их эффективному использованию в составе программного обеспечения фотосепаратора. В связи с этим предлагается комбинированный подход, на высоком уровне абстракции представимый в виде последовательных этапов:

1. Анализ кривизны предварительно сглаженного эллиптическим рядом Фурье контура и разделение контура с использованием полученной информации.

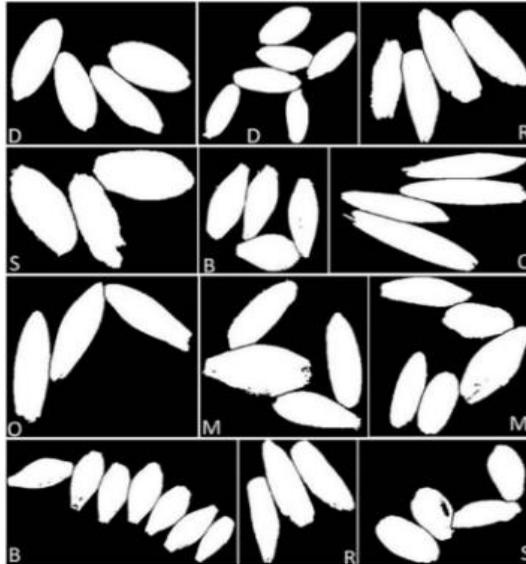


Рис. 2. Примеры удачного разделения соприкасающихся зерен методом на основе сглаживания границ объекта с помощью эллиптического ряда Фурье и последующего анализа их кривизны

2. Если необходимо, «доразделение» последовательно применяемыми операциями эрозии и дилатации.

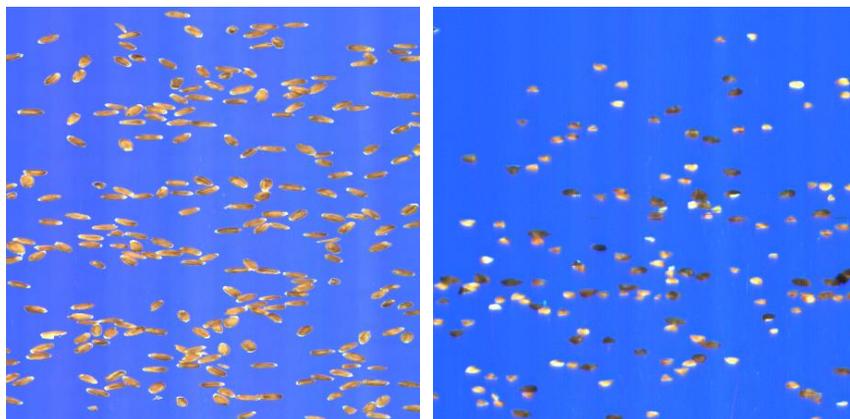
3. Если предыдущих шагов оказалось недостаточно, построение разделяющей линии по малой оси эллипса, описанного вокруг исследуемого контура.

Стоит отметить, что второй и третий этапы являются опциональными и целесообразность применения каждого из них определяется результатами работы предыдущего: находится площадь каждого контура и, если она превышает некоторое пороговое значение, задаваемое для каждой смеси зерна, к контуру применяется следующий этап алгоритма.

3. Результаты

Были проведены тестовые испытания алгоритма с использованием изображений пшеницы и гречневая крупа + гречка при высокой плотности потока (рис. 3).

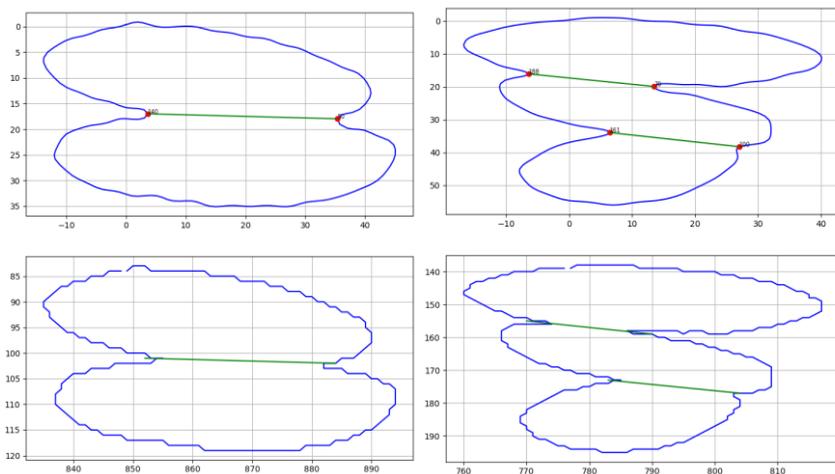
На рис. 4-рис. 6 показаны сложные для разделения случаи, с которыми предлагаемый метод справился успешно. Верхние изображения – построение разделяющих линий по сглаженному контуру, нижние – перенос построенных разделяющих линий на исходный контур.



а

б

Рис. 3. Пример изображений потока зерна: *а* – гречневая крупа + гречка, *б* – пшеница



а

б

Рис. 4. Сложные для разделения случаи: *а* – пример малой кривизны в конечной точке одной из разделяющих линий, *б* – пример большой площади соприкосновения зерен

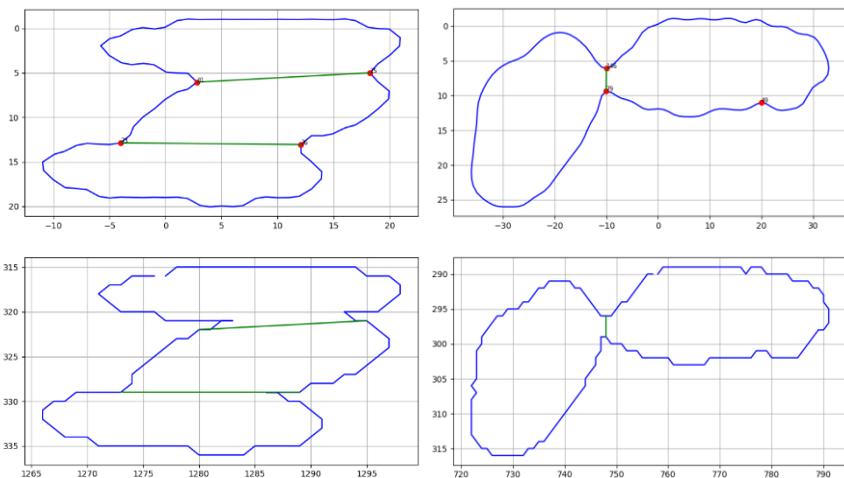


Рис. 5. Примеры большой площади соприкосновения зерен

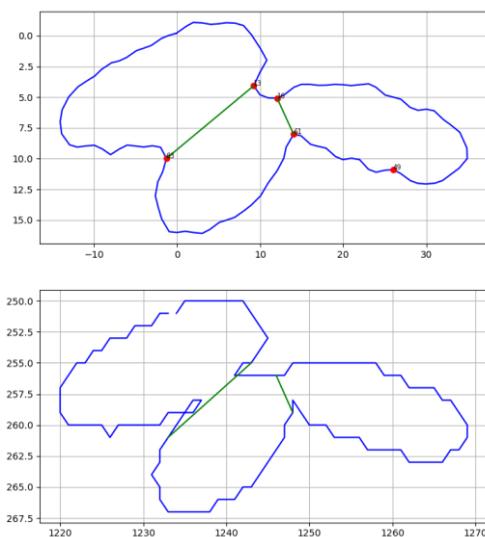


Рис. 6. Пример множественного соприкосновения зерен

Важная особенность предлагаемого алгоритма: в связи с применением грубого разделения по малому радиусу эллипса (3 этап) в случае, если более сложные подходы не сработали, вероятность неразделения зерна практически стремится к нулю.

Заключение

В данной работе представлен алгоритм разделения соприкасающихся зерен, благодаря своим свойствам пригодный для использования в условиях реального производства. Например, возможно его применение в составе программного обеспечения фотосепаратора. Наличие такого предварительного этапа перед выполнением непосредственно классификации зерен должно способствовать значительному увеличению ее качества.

Список литературы

1. Shatadal P. Digital image analysis for software separation and classification of touching grains / P. Shatadal, D. S. Jayas, N. R. Bulley // Transactions of the ASAE. – 1995. – № 38 (2). – P. 635-643.
2. Wang W. Separation and identification of touching kernels and dockage components in digital images / W. Wang, J. Paliwal // Canadian Biosystems Engineering. – 2006. – № 48. – P. 1-7.
3. Mebatsion H. K. Fourier analysis based algorithm to separate touching kernels in digital images / H. K. Mebatsion, J. Paliwal // Biosystems Engineering. – 2011. – № 108. – P. 66-74.
4. Hiraoka Y. Identification of Rhus succedanea L. cultivar using elliptic Fourier descriptors based on fruit shapes / Y. Hiraoka, N. Kuramoto // Silvae Genetica. – 2004. – № 53. – P. 221-226.
5. Plant species identification using Elliptic Fourier leaf shape analysis / J. C. Neto et al. // Computers and Electronics in Agriculture. – 2006. – № 50. – P. 121-134.

ЭФФЕКТИВНЫЙ ПОИСК СООТВЕТСТВИЙ МЕЖДУ ТРЕХМЕРНЫМИ ОБЪЕКТАМИ МЕТОДОМ ТРАНСФЕРА ФУНКЦИЙ

М. В. Землянухин

Студент

Д. И. Соломатин

Старший преподаватель

Введение

В последние годы происходит стремительное развитие в области компьютерной графики и визуальных эффектов, применяемых в кинематографе и игровой индустрии. Современные видеокарты позволяют отображать трехмерные модели, состоящие из огромного количества

вершин и полигонов и имеющие высоко-детализированные текстуры. С развитием аппаратных средств растет нагрузка на разработчиков, которые осуществляют наполнение игр искусственно созданными детализированными объектами, которых для больших проектов требуются тысячи и десятки тысяч.

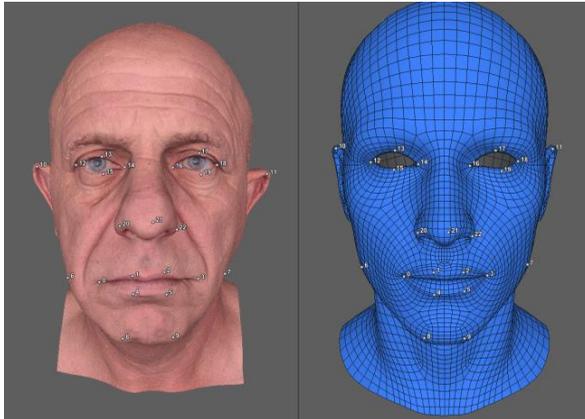
Трехмерное сканирование – процесс, в ходе которого объекты захватываются в реальном мире и воссоздаются в виртуальном мире. В результате применения трехмерного сканирования, с относительно небольшими временными затратами создаются объекты, отвечающие самым высоким требованиям пользователей.

Программный продукт компании Russian3DScanner под названием Wgr 3 является инструментом, упрощающим обработку трехмерных сканов. Инструмент применяется сразу после получения трехмерного объекта. Вместо обработки и исправления топологии объекта, полученного в результате реконструкции, используется другой (заранее подготовленный) трехмерный объект с удобной для работы и анимирования топологией, который с помощью Wgr 3 автоматически с высокой точностью принимает форму исходного объекта (рис. 1).

Существующий алгоритм совмещения базового и исходного трехмерных объектов, помимо самих объектов, принимает на вход выставленные вручную некоторые пары соответствующих точек, а в процессе выполнения в зависимости от своей вариации использует геометрическую и/или текстурную информацию об объектах. В данной работе исследуются пути уменьшения потребности в ручной установке ключевых точек и пути уменьшения времени работы алгоритма. Основной задачей является разработка эффективного (с точки зрения времени выполнения) алгоритма, который способен найти грубые соответствия между трехмерными моделями в виде пар соответствующих точек.

1. Введение в метод трансфера функций

Впервые метод трансфера функций был описан в [1] в 2012 году. Метод используется для нахождения плотных соответствий между трехмерными объектами в виде пар точек. Вместо работы над попарными соответствиями прямо, метод трансфера функций рассматривает перенос функций с вещественным значением с одного объекта на другой с помощью компактной *матрицы трансфера*, нахождение которой и является основной задачей метода. После нахождения матрицы трансфера с ее помощью возможен переход к попарным соответствиям. Далее рассматриваются некоторые понятия, требуемые для краткого описания метода.



а

б

базовая модель в процессе выполнения алгоритма принимает форму исходной модели, сохраняя при этом связь между вершинами и их количество

Рис. 1. Трехмерные модели лица человека: а – исходная модель, б – базовая модель

2. Сохранение действия дескрипторов

В большинстве задач на сравнение и поиск соответствий между объектами в трехмерном мире для оценки «похожести» вершин используются *дескрипторы* – наборы вещественных чисел, характеризующих каждую вершину. Предполагается, что вершины с близкими значениями дескрипторов с большей вероятностью "похожи" (или соответствуют) друг другу, чем вершины с отдаленными значениями дескрипторов – в этом и заключается смысл введения дескрипторов. Дескрипторы, как правило, могут учитывать два вида информации:

- Локальная информация – информация о вершине в контексте ее роли среди соседних вершин.
- Глобальная информация – информация о вершине в контексте всего объекта в целом.

Примеры дескрипторов без их описания: кривизна в вершине, WKS, HKS, SHOT, DEP.

Оператор Лапласа-Бельтрами в непрерывном мире является дивергенцией градиента функции в вершине, а в дискретном мире вычисляется для каждого меша отдельно как разреженная матрица, состоящая из n строк и n столбцов, где n – количество вершин у объекта. Количество ненулевых элементов в матрице относительно мало и равно

сумме количества вершин и ребер в меше. Подробнее об операторе Лапласа-Бельтрами на трехмерных объектах и его стандартной дискретизации описано в [2], а для данной работы важно разложение матрицы оператора на собственные вектора и значения. Матрица оператора является симметричной положительно определенной, то есть все ее собственные значения неотрицательны. При расположении собственных значений матрицы по возрастанию соответствующие им собственные вектора будут описывать функции на трехмерном объекте, расположенные по степени гладкости от низкочастотных к высокочастотным. Интуитивно, базис оператора Лапласа-Бельтрами для трехмерных объектов можно сравнить с базисом Фурье.

При расположении собственных значений оператора Лапласа-Бельтрами по возрастанию, можно взять несколько (50-100) первых значений и использовать соответствующие им собственные функции как уменьшенный базис для записи в нем значений, определенных в каждой вершине (*дескрипторов*). При этом потери данных, связанные с переходом в уменьшенный базис, уменьшаются с увеличением количества собственных функций все более незначительно, другими словами, собственные вектора оператора Лапласа-Бельтрами являются оптимальным базисом для аппроксимации функций на трехмерном объекте.

При наличии дескрипторов достаточного количества и качества в теории возможно вычислить матрицу трансфера уже на данном этапе, используя формулу (1). Для лучшего понимания матрицы трансфера, можно для пары трехмерных объектов с известными соответствиями в качестве дескриптора использовать матрицу перестановки, которая устанавливает соответствия между вершинами двух объектов. Визуализация некоторых вычисленных таким образом матриц трансфера представлена на рис. 2.

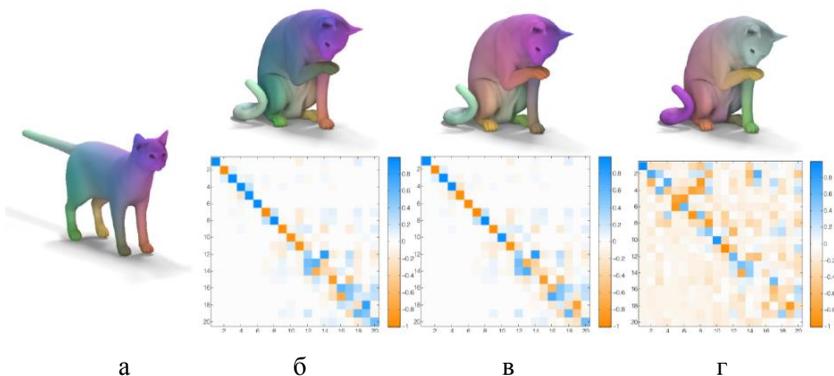
$$E_1(C) = \|CA - B\|^2, \quad (1)$$

где C – искомая матрица трансфера функций;

A и B – матрицы дескрипторов исходного и целевого мешей соответственно, записанные в новом базисе, образованном первыми 50-100 векторами оператора Лапласа-Бельтрами.

3. Коммутирование матрицы трансфера с оператором Лапласа-Бельтрами

В добавление к требованию на сохранение действия дескрипторов, существует и используется требование на коммутирование линейных операторов с матрицей трансфера функций. Первый пример – оператор симметрии, который накладывает ограничение симметричности на оба меша.



цветная версия рисунка доступна по ссылке: <https://clck.ru/DSJkq>

Рис. 2. Примеры матриц трансфера: *а* – исходная модель, *б* – соответствующие точки, *в* – симметричное отображение точек, *г* – соответствие от хвоста к голове

Второй пример – оператор Лапласа-Бельтрами (и выведенные из него операторы, например, оператор распространения тепла), действия которых сохраняются при условии, что деформация одного меша в другой близка к изометрической (то есть примерно сохранены дистанции между точками). И наоборот, если деформация близка к изометрической, значит, оператор Лапласа должен коммутировать с матрицей трансфера. Аналогично введению в систему уравнения требования на сохранения действия дескрипторов, требование на коммутирование оператора Лапласа-Бельтрами вводится как энергия, которую требуется минимизировать (формула 2).

$$E_2(C) = \sum_{i,j} C_{i,j}^2 (L_i^N - L_j^M)^2, \quad (2)$$

где L^M , L^N – собственные значения дискретной версии оператора Лапласа-Бельтрами для первого и второго меша.

В отличие от введения энергии сохранения дескрипторов в качестве слагаемого в левой части системы уравнений, введения энергии коммутирования оператора с матрицей трансфера самого по себе недостаточно для нахождения искомой матрицы трансфера, так как тривиальное решение $C=0$ также является решением системы уравнений.

4. Получение попарных соответствий

Наиболее базовый способ составления системы уравнений ограничивается использованием суммы двух энергий. Ограничения, накладываемые на систему уравнений, включают в себя в данном случае:

- Сохранение действия дескрипторов
- Сохранение требования на коммутирование дискретной версии оператора Лапласа-Бельтрами с матрицей трансфера.

Формула приобретает вид (3).

$$C = \arg \min_x E_1(X) + E_2(X) = \|XA - B\|^2 + \alpha \|L^N X - XL^M\|^2, \quad (3)$$

где C – искомая матрица трансфера, а слагаемые соответствуют уравнениям (1) и (2), α – весовой коэффициент, балансирующий слагаемые.

Энергия, которую требуется минимизировать, является выпуклой, поэтому поддается прямой оптимизации одним из методов линейной алгебры (например, с помощью QR-разложения).

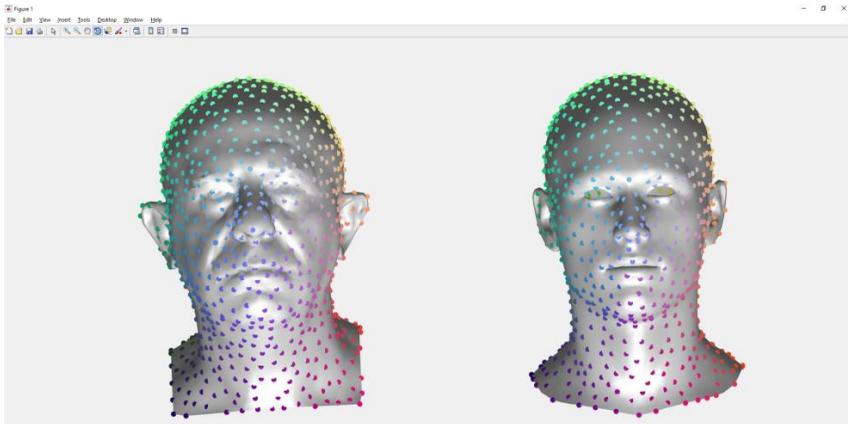
Переход к попарным соответствиям осуществляется с помощью формулы (4), подробнее о переходе можно найти в [1].

$$C\phi_M \rightarrow \phi_N, \quad (4)$$

где C – матрица трансфера, полученная на предыдущем этапе; ϕ_M – матрица собственных функций дискретной версии оператора Лапласа-Бельтрами первого меша, в которой каждая колонка соответствует точке первого меша, а каждая строка – собственной функции; ϕ_N – аналогичная матрица для второго меша, в которой каждая колонка соответствует собственной функции, а каждый столбец – точке второго меша; \rightarrow – так обозначено то, что для дальнейшего получения попарных соответствий требуется для каждого столбца матрицы, полученной в левой части, найти ближайший в смысле Евклидова расстояния столбец матрицы в правой части. Это можно сделать быстро, используя одну из подходящих структур данных, например, kd-tree.

Матрица трансфера, полученная путем решения системы уравнений (3), может наилучшим образом удовлетворять наложенным на нее ограничениям, но не быть подготовленной для дальнейшего извлечения из нее попарных соответствий. Поэтому перед поиском ближайших векторов в k -мерном пространстве требуется осуществить выравнивание облаков точек. Используется жесткое выравнивание итеративным методом ближайших соседей.

Пример результата работы алгоритма показан на рис. 3.



а

б

цветная версия рисунка доступна по ссылке: <https://clck.ru/DSJnq>

*Рис. 3. Результат выполнения алгоритма (найденное соответствие точек):
а – исходная модель, б – базовая модель*

Важно отметить, что после оптимизации матрицы трансфера и перехода к задаче поиска ближайших соседей с предварительным выравниванием облаков точек в k -мерном пространстве, получаемые путем поиска ближайших соседей попарные соответствия не являются непрерывными для трехмерного объекта. Для сглаживания соответствий используется метод, описанный в [3] и представляющий собой отдельную объемную задачу, решение которой включает минимизацию невыпуклой энергии. Рассмотрение метода выходит за рамки данной статьи, однако, он был использован для получения финальных результатов.

Заключение

В ходе работы разработан эффективный алгоритм, который способен найти грубые соответствия между трехмерными моделями в виде пар соответствующих точек. Среднее время выполнения алгоритма для объектов, содержащих суммарно более миллиона вершин, на четырех-ядерном процессоре Intel i3-8100, 3.6 GHz составило 6.03 секунды.

Список литературы

1. Computing and processing correspondences with functional maps / M. Ovsjanikov [and all] // SIGGRAPH Asia 2016 Course Notes [Электронный ресурс]. – Режим доступа : http://www.lix.polytechnique.fr/~maks/fmaps_course/notes/siggraph_notes_course_fmmaps.pdf

2. Rustomov R. M. Laplace-Beltrami eigenfunctions for deformation invariant shape representation / R. M. Rustomov // SGP '07 Proceedings of the fifth Eurographics symposium on Geometry processing. – Barcelona, Spain, July 04-06, 2007. – P. 225-233.

3. Efficient Deformable Shape Correspondence via Kernel Matching / Z. Löhner [and all] // Computing Research Repository (CoRR). – 2017. [Электронный ресурс]. – Режим доступа : <https://arxiv.org/abs/1707.08991>

ИССЛЕДОВАНИЕ ПОМЕХОУСТОЙЧИВОСТИ АЛГОРИТМА МІМО ДЛЯ BPSK И 8-PSK МОДУЛЯЦИИ

Д. А. Зимин

Студент

Д. Н. Борисов

Доцент

Введение

В современных системах беспроводной связи постоянно ставится вопрос об улучшении надежности и увеличении скорости передачи данных, следовательно, повышаются требования к пропускной способности беспроводных сетей. Увеличение ширины канала и использование модуляций более высокого порядка – очевидные варианты увеличения пропускной способности – не позволяют полностью решить поставленную задачу, тогда как технология МІМО полностью справляется с этими вопросами [1]. МІМО требует использования нескольких антенных элементов на передающей и принимающей стороне.

Основной проблемой в системах беспроводной связи является искажение передаваемого сигнала, вызванное различного рода помехами, замираниями исходного сигнала и его затуханием. Наличие искажений в принятом сигнале препятствует правильной работе алгоритмов их обработки, соответственно, распознаванию исходного сигнала. Внедрение сложных технологий требует теоретического и имитационного исследований. Данная работа посвящена исследованию возможности компенсации ошибок в переданном сигнале за счет изменения количества антенн на передающей и приемной стороне для разных видов модуляции.

1. Матрица Аламути

Общей особенностью радиоволн частотных диапазонов, используемых в системах беспроводной связи, является наличие многолучевого распространения сигнала, которое возникает вследствие

многократных переотражений исходного сигнала от различного рода естественных препятствий, что приводит к тому, что на приемную антенну разными путями приходит множество измененных копий одного сигнала [2]. Поскольку отраженные сигналы проходят различные расстояния, результирующий сигнал в точке приема представляет собой интерференцию собственных смещенных по времени копий исходного сигнала с разными амплитудами и фазами. Суммирование или взаимное поглощение амплитуд сигналов в точке приема называют замиранием. Глубокие замирания, в основном, и влияют на качество передачи информации [3]. В ММО передачу осуществляют одновременно несколько антенн, поэтому на каждую приемную антенну приходят переотраженные копии сигнала с каждой передающей антенны. Для возможности распознавания исходных сигналов в технологии ММО используется матрица Аламоути [4]. Принцип кодирования по Аламоути состоит в том, что подлежащая передаче последовательность символов разбивается на группы из нескольких символов, объединение которых по определенному правилу формирует матрицы, определяющие алгоритм передачи для каждой антенны. На рис. 1 представлена матрица Аламоути для ММО структуры 2x2.

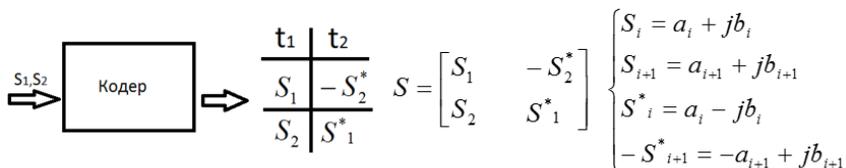


Рис. 1. Матрица Аламоути для ММО 2x2

Из математического описания (рис. 1) видно, что если в первом временном интервале сигналы синфазны, то во втором они обязательно будут в противофазе, т. е. если в первом интервале сигналы ортогональны, то они такими останутся и во втором временном интервале с той лишь разницей, что опережение фазы сигнала поочередно будет возникать, то на первой, то на второй антенне. В результате, такие манипуляции с компонентами исходного сигнала во много раз увеличивают возможность распознавания сигнала на приемной стороне [4].

2. Моделирование ММО 2x2 в Simulink

Реальная модель ММО 2x2, представленная основными физическими устройствами и описывающая ее общую структуру, изображена на рис. 2.



Рис. 2. Реальная модель технологии MIMO 2x2

Источник информации включает в себя генератор, обеспечивающий подачу сигналов с определенной частотой, кодер, гарантирующий их последующую модуляцию и составление матриц Аламути, и антенную решетку, состоящую из 2 передающих антенн. Канал представляет собой воздушную среду, в которой реализуется многолучевое распространение. В приемник входят антенная решетка с 2 приемными антеннами, декодер, производящий распознавание принятого сигнала и его дальнейшую демодуляцию, а также накопитель.

В имитационной среде моделирования Simulink существуют некоторые особенности для интерпретации физических процессов. Например, Simulink не позволяет производить пространственное сложение сигналов, так как передача информации осуществляется по проводам. В связи с чем, имитационная модель MIMO структуры 2x2, показанная на рис. 3, будет иметь большее количество блоков, чем в реальной модели.

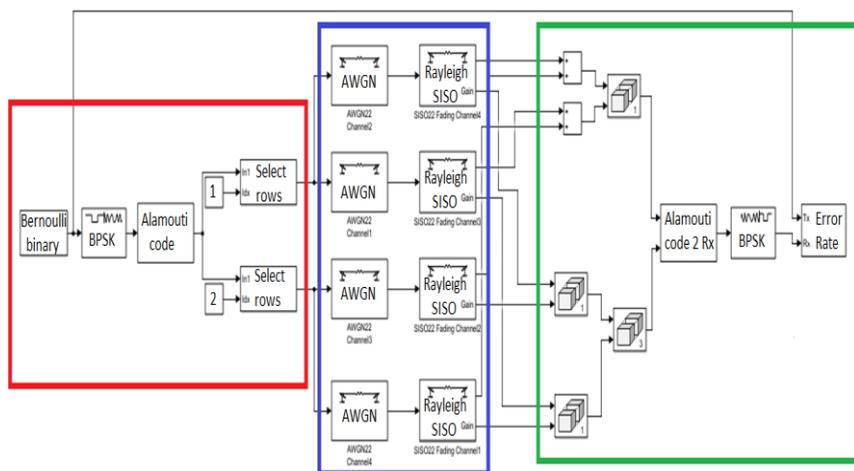


Рис. 3. Имитационная модель MIMO 2x2

Рассмотрим свойства используемых блоков модели Simulink. Генератор представлен блоком Bernoulli binary, который выдает заданное количество случайных бит с определенной частотой. Кодер – это блок BPSK-modulation, отвечающий за модуляцию сигнала, идущий далее блок Alamouti code, обеспечивающий генерацию матрицы Аламоути для пришедшей пары модулированных сигналов. Затем производится имитация разделения передающих антенн в антенной решетке путем выбора из матрицы определенных строк, причем номер строки в матрице Аламоути соответствует номеру передающей антенны. Поскольку в Simulink нельзя учесть временной фактор, то все составляющие сигнала, полученные после Alamouti code, передаются одновременно в виде вектора. Среда передачи имитируется блоком AWGN, добавляющим в исходный вектор сигналов Гауссовский шум, и следующим за ним блоком Rayleigh SISO, создающим замирания, распределенные по Релеевскому закону. Приемник состоит из двух сумматоров, реализующих сложение сигналов от двух излучателей антенной решетки, и блока слияния векторов в матрицу. Далее используется блок Alamouti Code 2 Rx, который определяет исходную пару модулированных сигналов из матрицы зашумленных сигналов. В итоге производится демодуляция сигналов с помощью блока BPSK-demodulation. В качестве накопителя (получателя информации) используется блок Error Rate, который подсчитывает отношение ошибочных бит к исходным битам.

3. Фазовая манипуляция PSK

Фазовая манипуляция (Phase-Shift Keying, PSK) – один из видов фазовой модуляции, при котором фаза несущего колебания меняется скачкообразно [5]. Далее представлен вид фазомодулированного сигнала $S_m(t)$.

$$S_m(t) = g(t) \cos[2\pi f_c t + \varphi_m(t)],$$

где $g(t)$ – огибающая сигнала, f_c – частота несущей, t – время, $\varphi_m(t)$ – модулирующий сигнал.

Сигнал $\varphi_m(t)$ может принимать M дискретных значений. Так, если $M=2$, то мы имеем дело с BPSK – двоичной фазовой манипуляцией. При $M=4$ будет QPSK – квадратурная фазовая манипуляция. При $M = 8$ формируется восьмипозиционная фазовая модуляция (8-PSK). M означает количество идеальных образов в сигнальном созвездии для модулированного сигнала, представленных на рис. 4.

Обычно, фазовая модуляция более высокого порядка не используется, а количество одновременно передаваемых бит можно посчитать с помощью формулы $n = \log_2 M$.

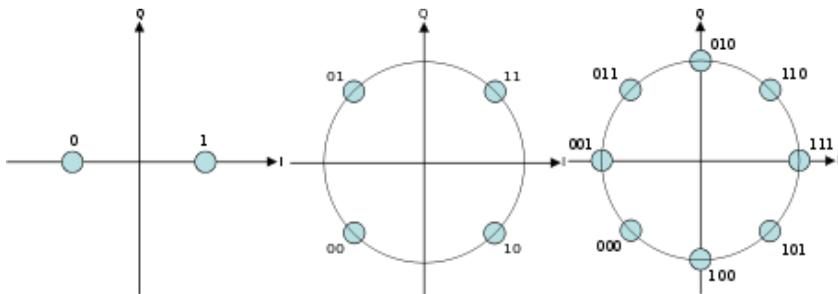
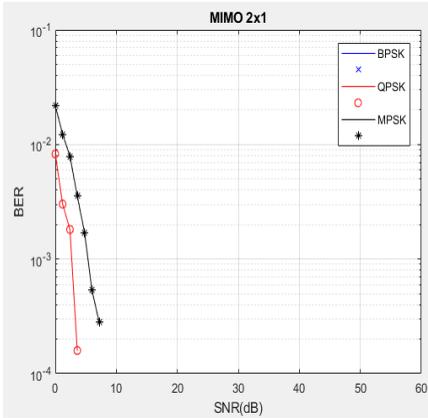


Рис. 4. Сигнальное созвездие для BPSK, QPSK, 8-PSK

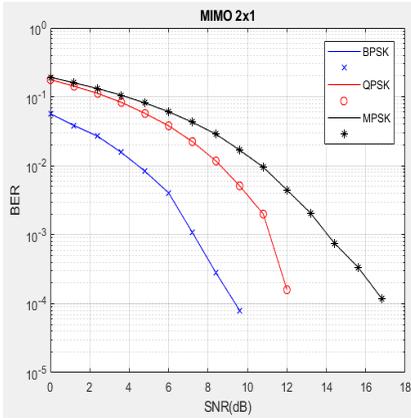
4. Помехоустойчивость для BPSK, QPSK, 8-PSK модуляций

Определим, какой вид модуляции наиболее эффективно использовать с точки зрения помехоустойчивости. В данной работе рассматриваются 3 вида модуляции, рассмотренные в разделе 3. Для анализа эффективности рассматривалась оценка величины BER (Bit Error Rate) – отношение ошибочных бит к исходным. Для того, чтобы результат исследования можно было считать правдоподобным, во всех следующих исследованиях для каждого фиксированного значения отношения сигнал/шум (SNR) было проведено 1000 испытаний для усреднения результата по стохастическим характеристикам. Эксперимент проводился для случая MIMO 2x1, чтобы исключить улучшение помехоустойчивости, реализованное добавлением антенн на приемной стороне. Для каждого численного эксперимента моделирование проводилось за фиксированное время t . Для BPSK, QPSK, 8-PSK модуляции для описания количества переданных бит используется отношение 1 : 2 : 3. На рис. 5 представлен результат эксперимента для сигнала мощностью P_s в 0.05 Вт, 0.2 Вт и 0.5 Вт.

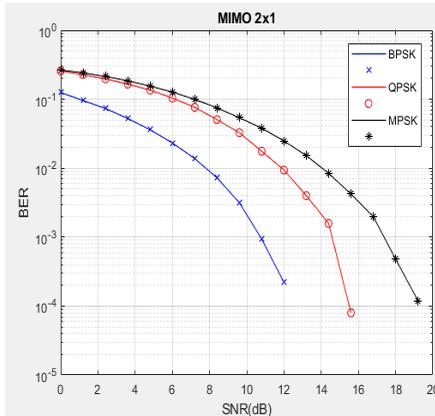
Анализ результатов моделирования показывает, что при $P_s = 0.05$ Вт (рис. 5) для BPSK модуляции ошибки при передаче отсутствуют. Для полученных результатов явно наилучшими будут те, значения ошибки BER которых будут меньше, при одних и тех же значениях SNR. В результате можно сделать вывод, что для реализации улучшенной помехоустойчивости необходимо выбирать BPSK модуляцию. С другой стороны, скорость передачи информации при BPSK модуляции наименьшая.



а



б



в

Рис. 5. Зависимость BER от SNR: а – $P_s = 0.05$ Вт, б – $P_s = 0.2$ Вт, в – $P_s = 0.5$ Вт

5. Компенсация ошибок для 8-PSK модуляции

Реализация алгоритма MIMO способна улучшать помехоустойчивость путем увеличения количества приемных и передающих антенн. Проведем исследование возможности компенсации ошибок за счет изменения количества приемных антенн для BPSK и 8-PSK модуляции. На рис. 6 представлен результат численного эксперимента для случая MIMO $2 \times N$ и MIMO $3 \times N$ с 8-PSK модуляцией и сигналом мощностью $P_s = 0.5$ Вт.

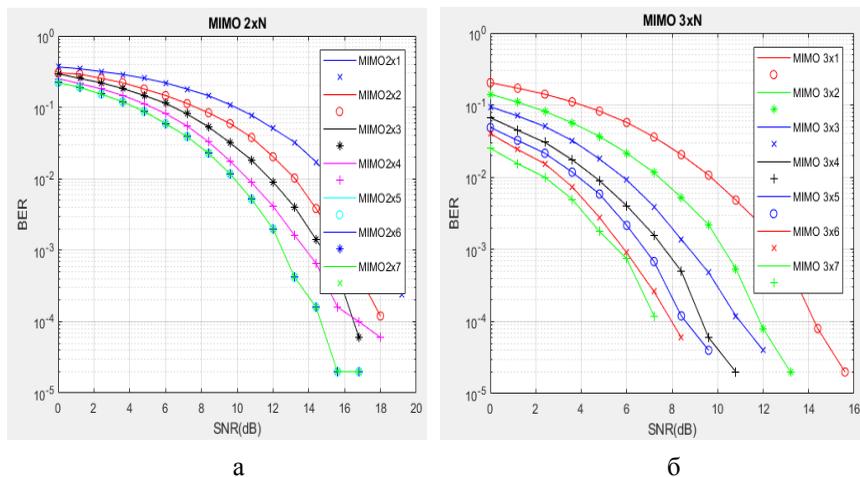


Рис. 6. Зависимость BER от SNR для 8-PSK: а – MIMO 2×N, б – MIMO 3×N

Сравнивая результаты моделирования для MIMO 3xN с 8-PSK и MIMO 2x1 с BPSK модуляцией можно сделать вывод, что ошибка для 8-PSK модуляции меньше. Анализ результатов показывает, что уже для случая MIMO 3x4 с 8-PSK модуляцией ошибка меньше, чем для случая MIMO 2x1 с BPSK модуляцией.

Основываясь на полученных результатах, можно сделать вывод о том, что увеличение количества антенн на передающей стороне на 1 и на приемной стороне на 3 способно полностью компенсировать разницу в ошибках для 8-PSK и BPSK модуляции.

Заключение

На основе данного исследования можно сделать вывод о том, что наиболее эффективным методом модуляции с точки зрения помехоустойчивости является BPSK модуляция. Связано это с тем, что на декодере Аламути происходит распознавание образов принятых зашумленных сигналов, каждый из которых описан действительной и мнимой частью. Также известно, что вероятность ошибки при распознавании тем меньше, чем меньше количество образов, к которым можно отнести оцениваемые данные. Так, для BPSK модуляции количество образов минимально. В то же время, скорость передачи информации для BPSK модуляции наименьшая. Существует возможность улучшить помехоустойчивость, увеличивая количество задействованных антенн. Кроме того, для каждого фиксированного количества антенн на

передающей стороне существует их предельное количество в приемной антенной решетке, увеличение которых уже не влияет на помехоустойчивость.

Список литературы

1. Банкет В.Л. Методы пространственно-временного кодирования для систем радиосвязи : учеб. пособие / В.Л. Банкет, Н.В. Незгазинская. – М. : Проспект, 2010. – 246 с.
2. Многолучевое распространение радио-сигналов классификация [Электронный ресурс] : база данных. – Режим доступа : <https://studfiles.net/preview/4349837/page:28/>
3. Сущность замираний и их классификация [Электронный ресурс] : научная библиотека. – Режим доступа : http://sernam.ru/book_fink.php?id=51
4. Пространственно-временное кодирование [Электронный ресурс] : база данных. – Режим доступа : <https://studfiles.net/preview/5157419/page:24/>
5. Рашич А.В. Сети беспроводного доступа WiMAX : учеб. пособие / А.В. Рашич. – М. : Спутник Плюс, 2011. – 125 с.

ИССЛЕДОВАНИЕ И ОЦЕНКА УГРОЗ БЕЗОПАСНОСТИ БЕСПРОВОДНЫХ WI-FI СЕТЕЙ (НА БАЗЕ СТАНДАРТА IEEE 802.11)

Е. Е. Ивашкина

Студент

С. А. Вялых

Доцент

Введение

Стремительное развитие беспроводных сетей привело к тому, что отсутствие поддержки технологии Wi-Fi в современных выпускаемых устройствах является серьезным недостатком для потенциального пользователя. Возможность освободиться от размещения кабелей или проводов и перейти к использованию сети практически в любом месте, привлекает всё больше и больше желающих воспользоваться этим. Простота, мобильность и удобство размещения новой точки доступа позволяют использовать беспроводные сети во многих сферах жизни: от маленькой квартиры до огромного завода.

Такое широкое применение притягивает не только обычных пользователей, но и желающих проникнуть в такие сети и перехватить информацию внутри нее. Параллельно с развитием беспроводных технологий развивались и средства перехвата беспроводных сигналов. И уже здесь возникают сложности, связанные с тем, что единственной физической границей Wi-Fi сети является сам уровень сигнала, который ограничен лишь мощностью, но не помещениями или другими физическими препятствиями. Это позволяет атакующим, находящимся в непосредственной близости от беспроводных структур, производить целый ряд нападений, которые были невозможны в проводном мире.

Стоит отметить, что, согласно банку данных угроз информационной безопасности [1], некоторые угрозы безопасности беспроводных соединений может осуществить внутренний или внешний нарушитель с низким потенциалом.

Последние нормативные документы, связанные с оценкой информационной безопасности, включают требования анализа уязвимостей и тестирование информационной системы на проникновение.

Таким образом, задача защиты беспроводной сети является актуальной и неотъемлемой частью построения такого соединения.

1. Классификация методов реализации угроз безопасности беспроводных Wi-Fi сетей

Под *угрозой безопасности информации* понимают совокупность условий и факторов, создающих потенциальную или реально существующую опасность, связанную с утечкой информации и несанкционированными или непреднамеренными воздействиями на нее [2].

Угроза с определенной вероятностью может быть реализована в той или иной беспроводной сети. В зависимости от уровня подготовки потенциального нарушителя и имеющегося у него оборудования, может быть осуществлена любая атака на сеть. Стоит отметить, что в отличие от проводных сетей, нарушителю не нужно находиться в непосредственной близости к своей цели, ему достаточно быть в зоне приема сигнала сети.

Классификация методов реализации угроз представлена на рис. 1.

2. Описание и анализ методов реализации угроз безопасности беспроводных Wi-Fi сетей

Рассмотрим подробнее методы реализации угроз безопасности беспроводных Wi-Fi сетей. В качестве оценки опасности каждого из методов используются следующие значения: высокая – злоумышленник может получить практически неограниченный доступ к системе — права санкционированного пользователя (администратора), как для управления, так и для получения доступа к информации в ней, средняя – утечка части ценной информации, низкая – утечка не критичной информации.

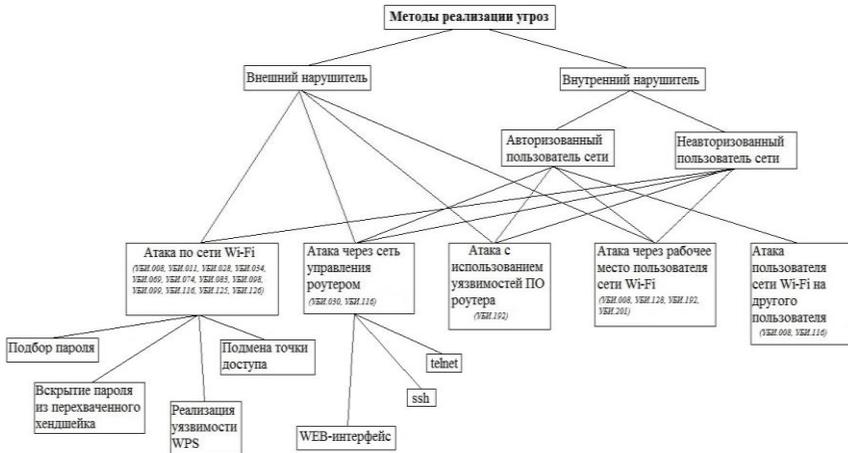


Рис. 1. Классификация методов реализации угроз с точки зрения типовых ситуаций

1. Атака по сети Wi-Fi

Включает в себя подбор пароля, вскрытие пароля из перехваченного хендшейка (рукопожатия), реализацию уязвимости WPS и подмену точки доступа.

Оценка опасности: средняя.

Способы противодействия: для исключения возможности подбора пароля или, по крайней мере, для усложнения самого процесса перебора, рекомендуется использовать пароль длиной не менее 10 символов, он не должен содержать осмысленные слова и фразы, а также слова и наборы цифр, имеющие отношения к личной жизни пользователя, отключение технологии WPS, повышенный контроль доступа, скрытие SSID, фильтрация по MAC-адресам, использование по минимуму открытых точек доступа.

2. Атака через сеть управления роутером

Данная атака может быть осуществлена тремя самыми известными способами: через HTTP (Web-интерфейс), через Telnet и через SSH.

Каждый роутер имеет некоторый Web-интерфейс, позволяющий настроить по многим параметрам точку доступа беспроводной сети. Главная проблема заключается в том, что многие пользователи пренебрегают корректной конфигурацией и оставляют все по умолчанию. Окно настроек может быть даже не защищено паролем и, следовательно, попасть в него может любой желающий, нужно только знать IP-адрес роутера. Как правило, вся необходимая информация для входа есть на

нижней панели маршрутизатора. Более того, бывают такие случаи, что данные логина и пароля указаны при первом входе на страницу конфигурации роутера. Попадая в указанные настройки, злоумышленник получает крайне широкий спектр возможностей работы с Wi-Fi: от просмотра служебной информации до возможности менять учетные данные сети или закрытия доступа санкционированным клиентам. Но даже не зная модель роутера, находясь не рядом с ним, злоумышленник может найти стандартные комбинации, которых не так уж и много, и методом подбора попасть в панель управления маршрутизатором.

Telnet – сетевой протокол, реализующий текстовый интерфейс через командную консоль с возможностью использования дополнительных команд для более тонкой настройки роутера.

SSH – аналог Telnet, но обладающий шифрованием трафика при авторизации и работе в сети.

Последние два сетевых протокола, как правило, на роутере по умолчанию не включены, и поэтому могут потребоваться некоторые дополнительные действия для их использования.

Оценка опасности: высокая;

Способы противодействия: использование логина и пароля администратора, отличных от заданных по умолчанию, запрет или ограничение сетевого доступа к роутеру.

3. Атака с использованием уязвимостей программного обеспечения (ПО) роутера

В качестве примера можно рассмотреть уязвимость микропрограммного обеспечения маршрутизатора D-Link DSR-500 [3], позволяющая злоумышленнику получить права администратора. Она заключается в том, что скрипт «/scgi-bin/platform.cgi» микропрограммного обеспечения маршрутизатора должным образом не фильтрует данные, вводимые пользователем в поле «Password», в результате чего злоумышленник может обойти процедуру аутентификации и получить доступ к устройству с привилегиями администратора. Часто такое оборудование продолжает эксплуатироваться без установки обновлений несмотря на то, что указанная уязвимость уже устранена производителем.

Оценка опасности: в зависимости от типа уязвимости опасность может быть низкой, средней или высокой.

Способы противодействия: установка актуальных обновлений ПО позволит избежать атак, направленных на обнаруженные уязвимости.

4. Атака через рабочее место пользователя сети Wi-Fi

Программное обеспечение, которое используют пользователи, не всегда может гарантировать безопасную работу и передачу данных. Некоторые приложения предлагают сохранять или сохраняют по

умолчанию пароли пользователей с целью упростить задачу авторизации. Даже сами пользователи могут неосознанно подставить себя под угрозу, используя одни и те же пароли в разных системах. Нельзя также быть уверенным в том, что никто или ничто не отслеживает действия или данные, которые вводит пользователь. Например, существует такой тип программ, как *кейлогеры* – ПО, регистрирующее различные действия пользователя, включая нажатые клавиши и открытые им приложения. Подобные программы размещены в свободном доступе и могут работать в невидимом для пользователя режиме.

Оценка опасности: средняя.

Способы противодействия: постоянное обновление антивирусных программ, установка приложений из доверенных источников, использование нестандартных, сложных и различных паролей позволит пользователю самому обеспечить требуемую безопасность при работе на любом устройстве.

5. Атака пользователя сети Wi-Fi на другого пользователя

В качестве примера данной атаки можно привести процесс перехвата информации по беспроводным каналам. Перехват информации может осуществляться различными средствами мониторинга сетевого трафика. Особую ценность для злоумышленника представляют логины и пароли пользователей различных сетевых сервисов. Так, например, при попытке аутентификации в каком-либо сервисе пользователь вводит и отправляет свои данные. Эта информация находится внутри сетевых пакетов и, так как не защищена, может быть перехвачена.

Оценка опасности: средняя.

Способы противодействия: использование надежного шифрования данных, например, технологию VPN.

3. Практическая оценка возможностей использования программ и методов проникновения

Для оценки возможностей несанкционированного доступа в Wi-Fi сети ВГУ была использована операционная система Kali Linux и встроенные в нее средства. В качестве объекта тестирования выбраны беспроводные сети факультета компьютерных наук: CS-Wireless и CS-Guest-WPA. Особенности данных сетей представлены в табл. 1.

В качестве средства, реализующего взлом учетных данных, был использован пакет программ Aircrack-ng. В процессе мониторинга сети была выбрана целевая точка доступа CS-Guest-WPA с наибольшим количеством передаваемых данных. Как только удалось перехватить хендшейк, был запущен процесс его расшифровки с использованием словаря, содержащего около 400000 различных комбинаций [4]. Через 18 секунд программа успешно завершила подбор пароля (рис. 2).

Таблица 1

Сравнительная характеристика Wi-Fi соединений ФКН

Критерии	CS-Wireless	CS-Guest-WPA
Протокол шифрования	WPA2	WPA2
Уникальность пароля	Индивидуален для каждого клиента сети	Общий
Доступ	Зарегистрированный преподаватель или студент ФКН ВГУ	Возможность авторизации для всех, кому известен пароль сети

```

Aircrack-ng 1.2 rc4
[00:00:18] 73924/389376 keys tested (4025.70 k/s)
Time left: 1 minute, 18 seconds                               18.99%
KEY FOUND! [ 201120** ]

Master Key      : 63 A0 7A 18 7A 05 A8 62 60 A6 B3 E1 7D 56 03 85
                  1F F3 7A 08 9F 6D 3A 39 4F 1F 5C 89 14 CB 1A 8F

Transient Key   : 65 A8 6E 65 8E 32 1E 7C BA F0 60 93 F3 74 17 BF
                  E8 21 19 F3 5B EB 38 4A 44 8F 1D AA 2B 55 EF 57
                  C1 2C 32 19 A8 23 FE EB C2 DF 56 71 5A 7C 57 8C
                  39 D7 7F FB E1 A9 CB CD 7F B4 53 AE 5F 9A 8B 37

EAPOL HMAC     : 5B 4D 6B C8 0C 21 A2 5C E9 90 2A EF C1 25 70 B5
  
```

Рис. 2. Успешное завершение расшифровки рукопожатия

Также было проведено исследование данного метода для сети CS-Wireless. В ходе работы выяснилось, что перехват хендшейка вполне осуществим, но из-за особенностей реализации подключения к данной сети вскрыть логин и пароль пользователя этой сети не удалось.

Реализация уязвимости WPS оказалась невозможной, так как ни одна точка доступа в зоне действия сетей CS-Wireless или CS-Guest-WPA не применяет данную технологию.

Для перехвата информации, передаваемой по беспроводному соединению, была использована программа Wireshark. Так как данная программа позволяет получить информацию только при подключении к целевой сети, то ее тестирование проводилось на CS-Guest-WPA, пароль к которой был получен в результате атаки на вскрытие пароля из хендшейка.

В результате работы с программой Wireshark удалось получить информацию о сайтах, которые посещали пользователи, но ввиду использования на найденных сайтах защищенного протокола HTTPS все данные оказались зашифрованы.

Результаты проведенного исследования отображены в табл. 2.

Таблица 2

Результаты тестирования на проникновение в Wi-Fi сети ФКН ВГУ

Метод тестирования на проникновение	Wi-Fi соединение	
	CS-Wireless	CS-Guest-WPA
Взлом пароля	–	+
Реализация атаки на уязвимость WPS	–	–
Перехват информации	–	+

4. Рекомендации по повышению эффективности защиты информации

Согласно полученным результатам практической оценки тестирования, на проникновение в беспроводные Wi-Fi сети ВГУ, можно составить следующие общие рекомендации по повышению эффективности защиты информации:

- использование сложных паролей длиной не менее 10 символов и не содержащих осмысленных слов или фраз;
- периодическое тестирование на проникновение с целью поиска новых уязвимостей и их устранения;
- использование фильтрации MAC-адресов и режима скрытого идентификатора SSID для предотвращения атак на поднятие ложной точки доступа;
- сегментирование беспроводной сети.

Заключение

В результате исследований было выяснено, что сеть CS-Wireless представляет собой сеть с высоким уровнем защищенности от несанкционированного доступа, в то время как CS-Guest-WPA имеет уязвимости. Таким образом, результаты анализа показывают высокую уязвимость технологий, реализующих Wi-Fi соединение, и требуют принятия дополнительных мер защиты, связанных с анализом и устранением постоянно выявляемых уязвимостей. По мере возможности рекомендуется отказаться от использования таких технологий в информационных системах, требующих обеспечить высокий класс или уровень защищенности.

Список литературы

1. Банк данных угроз безопасности информации [Электронный ресурс]. – Режим доступа : <http://bdu.fstec.ru/threat>
2. Щербаков, В. Б. Безопасность беспроводных сетей: стандарт IEEE 802.11 / В. Б. Щербаков, С.А. Ермаков. – М. : РадиоСофт, 2010. – 255 с.
3. Банк данных угроз безопасности информации [Электронный ресурс]. – Режим доступа : <http://bdu.fstec.ru/vul/2014-00002>
4. Словари для расшифровки рукопожатий [Электронный ресурс]. – Режим доступа : <https://wiki.skullsecurity.org/Passwords>

МОДЕЛИРОВАНИЕ РИСКОВ СКЛАДСКИХ ПРОЦЕССОВ В ПРОГРАММНОЙ СРЕДЕ ANYLOGIC

Д. В. Игнатова

Студент

И. В. Илларионов

Доцент

Введение

Практически каждое предприятие, так или иначе, сталкивается с необходимостью организации хранения на складе товаров, комплектующих и готовой продукции. От того, в какой степени налажено складское хозяйство, во многом зависит рациональное использование материально-технических ресурсов, повышение производительности труда и качество готовой продукции.

Деятельность любого склада представляет собой совокупность таких бизнес-процессов, как приемка товара и его размещение, пополнение запасов, маркировка, комплектация заказов, отгрузка, доставка и др. Непредвиденная ситуация, произошедшая на одном из участков этой цепи, может привести к нарушению работы всего склада. Поэтому чрезвычайно важно заранее предвидеть основные риски и учитывать их уже на этапе создания проекта склада.

Существует несколько способов проектирования складского помещения. Самый распространенный - аналитический, т. е. расчет параметров склада по специальным формулам. Основное достоинство подхода – простота. От аналитика требуется владение только общепринятыми программными средствами, например, таблицами Excel. Однако, такой подход не способен учесть ни динамику процессов, ни их стохастическую компоненту – отказы оборудования, нарушение

расписаний, погодных условий и др., т. е. такая модель практически не способна учитывать риски.

С развитием информационных технологий появились более удобные, быстрые и точные способы решения логистических проблем, в частности проектирования складов. Рассмотрим далее построение имитационной модели для решения этой задачи.

1. Имитационное моделирование

Компьютерные модели являются эффективным методом управления, прогнозирования и познания окружающей действительности и позволяют более четко осознать сущность изучаемых явлений.

В настоящее время в имитационном моделировании выделяют три подхода: системной динамики, дискретно-событийный и агентный [1]. Системная динамика предполагает высокий уровень абстракции и используется для задач стратегического уровня. Дискретно-событийный подход используется в на операционном и тактическом уровне. Агентное моделирование является подходом более универсальным и мощным, так как оно позволяет учесть любые сложные структуры и поведения.

Использование решений имитационного моделирования на этапе проектирования склада позволяет произвести расчеты таким образом, чтобы каждый кубический метр объема помещения и каждая единица грузоперерабатывающего оборудования использовались с максимальной эффективностью, раскрыть недостатки модели и помочь их устранению [2]. Также возможно произвести расчет рисков, которые всегда сопутствуют проектированию и эксплуатации склада.

Построение компьютерной моделей и проведение имитационных экспериментов выполняется в программной среде AnyLogic, которая дает возможность быстрого создания профессиональных агентных моделей в графической среде, а также поддерживает языковые конструкции для задания поведения агентов, их взаимодействия, и имеет богатейшие анимационные возможности [3].

Использование моделирования дает возможность оперативно изменять структуру складского комплекса: менять конфигурацию склада, увеличивать или уменьшать количество привлеченных рабочих и техники, не подвергая риску процесс производства.

2. Риски, возникающие в процессе работы склада

Проблема рисков является одной из наиболее важных в функционировании складов, ведь на эффективную работу складского хозяйства влияет множество факторов. Оптимальное управление складом и материально-техническими ресурсами зависит от процесса выявления источников риска и оценки значимости факторов риска.

При рассмотрении проблемы рисков в логистике часто применяют существующие разработки риск-менеджмента, ориентируя популярные классификации экономических рисков к области исследования – конкретным участникам/звеньям логистических систем: поставщикам/потребителям, провайдером логистических услуг, производственным и торговым предприятиям, объектам производственно-логистической инфраструктуры, подразделениям [4]. Логистические риски объединяют в себе различные виды рисков всех составляющих звеньев и элементов как в процессе изменения материальных, финансовых и информационных потоков, так и в процессе собственно управления рисками, возникающими в логистической системе [5].

Под рисками понимаются нежелательные (непредвиденные) события, которые могут существенно ухудшить те или иные стороны проекта (стратегии) или даже привести к его срыву [6].

Рассмотрим некоторые факторы риска, оказывающие влияние на деятельность складских систем:

- риски, связанные с отклонением времени поставок;
- риски, связанные с объемами поставок;
- риски, связанные со способом транспортировки;
- риски, связанные с наличием и типом погрузочно-разгрузочных средств;
- риски, связанные с организацией размещения товаров на складе;
- риски изменения количества и качества товара в зависимости от условий хранения.

3. Учет факторов риска в имитационной модели

Фактически, любые риски можно смоделировать, внося некоторую случайную составляющую в имитационную модель.

Складские процессы в рассматриваемой модели оптового склада выглядят следующим образом. Поддоны с грузами доставляются на склад грузовиками, обслуживающий персонал разгружает и транспортирует поддоны в зону приемки. Поддоны обрабатываются, отмечаются регистраторами, и затем перемещаются в зону хранения с помощью вилочных погрузчиков. Заказы принимаются на складе. Для каждого заказа погрузчик выбирает один поддон из хранилища и переносит его в зону отправки. В этой области контролирующий персонал сверяет содержимое поддона с заказом. После проверки на склад приходит грузовик и в него грузится поддон с заказом.

Рассмотрим добавление в модель оптового склада такой вероятный риск, как выход из строя складского оборудования, например, погрузчика. Учет подобного риска в имитационной модели позволит наглядно увидеть

его влияние на работу склада и скорректировать модель таким образом, чтобы последствия сводились к минимуму.

Ниже приводится пример созданных динамических событий, имитирующих: поломку вилочного погрузчика – ForkliftCrash (листинг 1) и возобновление работы отремонтированного погрузчика – ForkliftRepair (листинг 2).

Листинг 1

Динамическое событие ForkliftCrash

```
Random rnd = new Random();
if (forkliftsNumber > 0) {
    set_forkliftsNumber(forkliftsNumber - 1);
    create_ForkliftRepair(rnd.nextInt(19) + 1);
}
create_ForkliftCrash(rnd.nextInt(50) + 1);
```

Листинг 2

Динамическое событие ForkliftRepair

```
set_forkliftsNumber(forkliftsNumber + 1);
```

В динамическом событии ForkliftCrash происходит уменьшение количества погрузчиков через случайный промежуток времени от 0 до 50 минут модельного времени, затем создается событие ForkliftRepair, и, через промежуток времени от 0 до 19 минут модельного времени, происходит увеличение количества погрузчиков.

На рис. 1 видно, что увеличение числа поломок приводит к загрузенности приемочной зоны.

На рис. 2 представлен график загрузенности зоны приемки до добавления динамического события ForkliftCrash (поломка вилочного погрузчика) в модель. На графике видно, что зона приемки загрузена в среднем на 60-80%.

На рис. 3 изображен график загрузенности зоны приемки после внесения добавлений в модель работы оптового склада. Не трудно заметить, что после добавления нового события загрузенность зоны приемки заметно возросла.

Заключение

Как видно из полученных результатов, влияние подобных стохастических событий на модель может быть довольно велико. Добавление всего лишь одного риска в модель увеличило загрузенность зоны приемки примерно на 20%, что может быть критичным для работы всего склада.

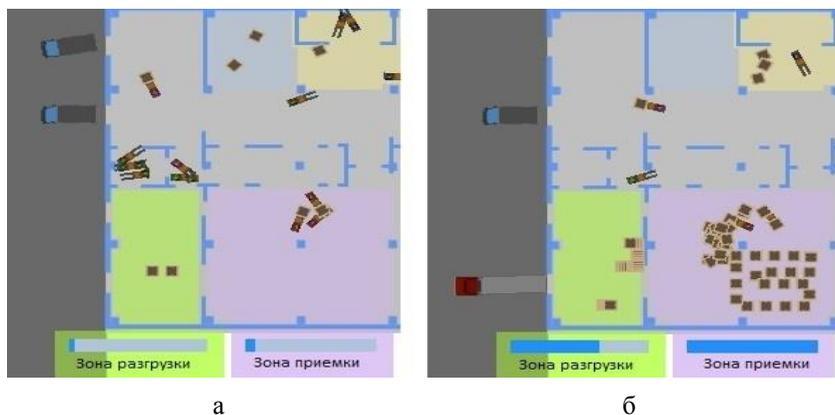


Рис. 1. Увеличение загруженности приемочной зоны: *а* – до внесения изменений в модель, *б* – после внесения изменений в модель

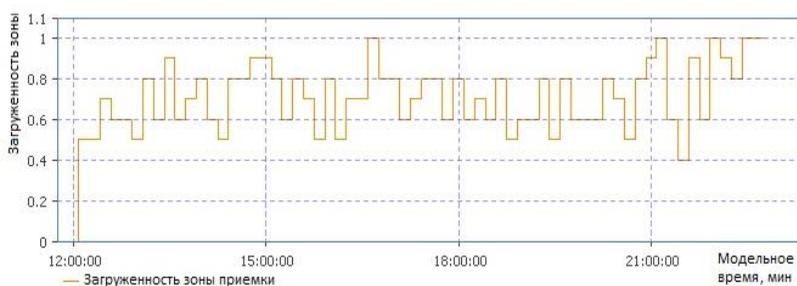


Рис. 2. График загруженности приемочной зоны до внесения изменения в модель

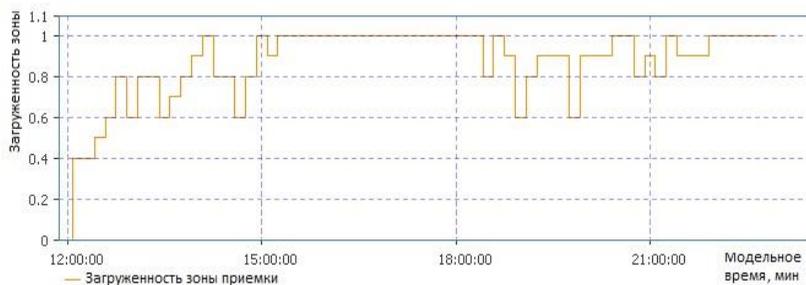


Рис. 3. График загруженности зоны приемки после внесения изменения в модель

Очевидно, что адекватность модели, построенной без учёта рисков, будет заметно ниже, чем у модели, в которой учтены, по крайней мере, самые вероятные риски. При этом, эффективность такой модели напрямую зависит от правильного расчета вероятности появления входящих в нее факторов рисков.

Таким образом, возможность учитывать большинство рисков складского хозяйства позволит не только заранее увидеть поведение модели в случае возникновения внештатных ситуаций и рассчитать влияние тех или иных рисков на работу склада, но также позволит снизить затраты при принятии управленческих решений.

Необходимо помнить, что добавление новых параметров усложняет модель, поэтому стоит отсеивать те риски, влияние и вероятность которых достаточно низки.

Список литературы

1. Куприяшкин А.Г. Основы моделирования систем: учеб. пособие / А.Г. Куприяшкин; Норильский индустр. ин-т. – Норильск : НИИ, 2015. – 135 с.
2. Бродецкий Г. М. Моделирование логистических систем. Оптимальные решения в условиях риска / Г. М. Бродецкий. – М. : Издательский дом «Вершина», 2006 – 376 с.
3. Каталевский, Д.Ю. Основы имитационного моделирования и системного анализа в управлении: учебное пособие; 2-е изд., перераб. и доп. / Д.Ю. Каталевский. – М. : Издательский дом «Дело» РАНХиГС, 2015. – 496 с.
4. Левина Т. В. Актуальные вопросы управления логистическими рисками / Т. В. Левина // Логистика и управление цепями поставок. – 2014. – Вып. 4. – С. 36-42.
5. Стерлигов К. А. Механизмы управления рисками в логистике/ К. А. Стерлигов // Логистика&система. – 2006. – Вып. 4. – С. 49-55.
6. Ляпина С. Ю. Управление рисками в инновационной деятельности: учеб. пособие / С. Ю. Ляпина, М. В. Грачева. – М. : Юнити-Дана, 2012. – 352 с.

РЕАЛИЗАЦИЯ АЛГОРИТМОВ ОБРАБОТКИ ИЗОБРАЖЕНИЙ В ИНФОРМАЦИОННОЙ СИСТЕМЕ ДЛЯ СОЗДАНИЯ ПАНОРАМ

Г. В. Измайлов

Студент

С. В. Борзунов

Доцент

Введение

На сегодняшний день обработка изображений является важным направлением применения современной вычислительной техники. Известны такие задачи обработки, как фильтрация и восстановление изображений, сегментация изображений в качестве средства сжатия информации. Но не менее важной задачей является построение двумерной модели на основе массива входных данных, представленных в виде фотографий, другими словами – создание панорам. Под панорамой понимается фотография, полученная путём технологии сборки изображения из отдельных кадров. Такие изображения применяются в различных научно-технических сферах: в медицине, промышленности, в социальных сетях, картографии и многих других [1].

В социальных сетях основной акцент делается не только на возможности общения, но и на совместном доступе пользователей к фото- и видеоматериалам. Однако не существует такого популярного приложения, которое бы строило для отдельно взятой фотографии панораму на основе снимков других пользователей, сделанных непосредственно в том же месте.

Основная трудность, заключающейся в решении данной задачи, является значительный объем анализируемых данных. Для реализации возможности создания панорам на основе множества снимков необходимо рассмотреть имеющиеся алгоритмы обработки изображений и оптимизировать их для достижения поставленной цели [2].

В силу этого представляется актуальной задача реализации алгоритмов обработки изображений в информационной системе для создания панорам.

1. Требования к прототипу информационной системы

Цель работы заключалась в разработке прототипа информационной системы для создания панорам на основе фотографий пользователей.

К разрабатываемому прототипу были выдвинуты следующие требования:

- использование базы данных, в которой хранятся все необходимые данные о пользователе, его фотографиях и созданных панорамах, а также наиболее популярных местах (где сделано наибольшее количество снимков различными пользователями);
- оптимизация существующих алгоритмов для создания двумерных изображений с возможностью быстрой обработки большого объема данных;
- наличие удобного и понятного пользовательского интерфейса.

Требования к составу и параметрам технических средств обусловлены тем, что задача создание прототипа – проверка возможности обработки большого объема данных программными средствами для дальнейшего создания полноценного приложения. Состав технических средств определен следующим образом:

- интернет-сервер для хранения базы данных;
- средство для создания базы данных – MySQL Workbench 6.0;
- система управления базой данных MySQL;
- язык программирования C#.

2. Реализация информационной системы по созданию панорам

На главной форме приложения необходимо пройти авторизацию или зарегистрироваться. Для авторизации необходимо ввести логин учетной записи и пароль, если они совпадут с имеющимися в базе данных, авторизация пройдет успешно. Если учетные данные не найдены, необходимо указать адрес электронный адрес, придумать имя учетной записи и пароль, если логин, придуманный вами, свободен, вы успешно пройдете регистрацию, будет прислано соответствующее письмо на почтовый ящик.

Когда вход в учетную запись осуществлен, пользователь имеет возможность указать дополнительную информацию о себе: место жительства, имя, увлечения.

Пользователь может загрузить свою фотографию, указать ее географические координаты и попытаться создать панораму, совместив ее со снимками других пользователей, сделанных в том же месте.

Отличительной особенностью задачи является то, что у снимков будут иметь различную яркость, контрастность, резкость и другие параметры. Для решения этой проблемы осуществляется предварительная обработка входных данных: снимки обрабатываются таким образом, чтобы у них примерно совпадали наиболее важные критерии для определения особых точек и дальнейшего создания панорамы.

Алгоритм работы программы представлен на рис. 1.



Рис. 1. Блок-схема алгоритма создания панорамы

Поиск особых точек на снимках осуществляется с помощью модернизированного алгоритма SURF: нам необходимо выделять больше особых точек, чем в стандартных задачах обработки изображений. Далее происходит сопоставление дескрипторов между двумя изображениями. При достижении порогового значения количества совпавших дескрипторов, происходит попытка сшивания изображений, в нашем случае это пороговое значение меньше стандартного из-за особенности поставленной задачи [3].

Если найдется хотя бы один снимок, который удовлетворяет условия создания панорамы, то откроется форма «Создание панорамы». На ней будут миниатюры пользовательских фотографий, информация о пользователях, с помощью которых строится двумерная модель и непосредственно панорама.

Затем строится матрица гомографии с использованием согласованных векторов признаков с помощью алгоритма RANSAC. В конце алгоритма применяется преобразование деформации на основе составленной ранее матрицы гомографии и стандартных методов объединения двух

изображений. Есть возможность сохранить полученную панораму на файловую систему компьютера или добавить в свою галерею в базу данных.

База данных была создана в приложении MS Workbench. Её особенность заключается в том, что с её помощью можно компактно хранить всё информацию о пользователях, их фотографиях, панорамах и любимых местах [4]. База данных приложения представлена на рис. 2. Из которого видно, что определены четыре сущности: пользователь, фотографии, места и панорамы.

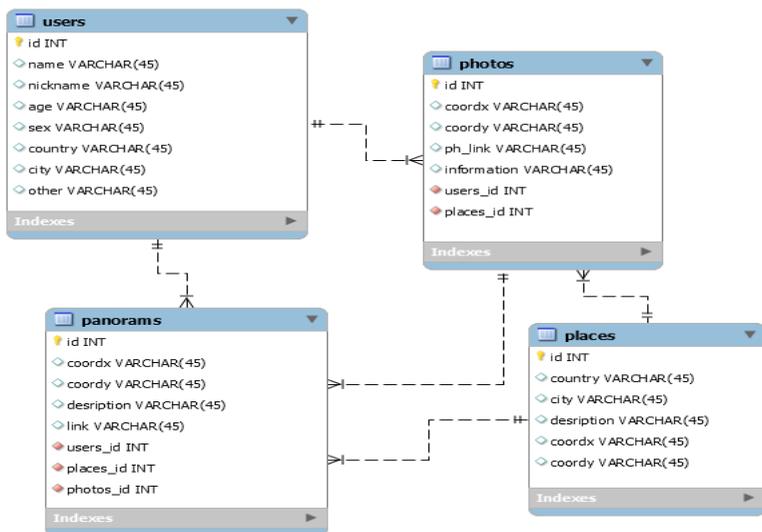


Рис. 2. Схема базы данных прототипа информационной системы для создания панорам

3. Результаты работы программного комплекса

Прототип информационной системы успешно строит панорамы на определенном наборе данных, которые подаются на вход. Данный способ обработки и анализа изображений демонстрирует довольно быструю скорость работы на большом объеме данных.

Результат работы приложения представлен на рис. 3 и рис. 4. Видно, что интерфейс прототипа информационной системы является развитым и, безусловно, позволяет пользователи решить задачи по созданию панорам на основе своих фотографий. Однако для создания полноценного приложения будут необходимы некоторые изменения в интерфейсе программы.

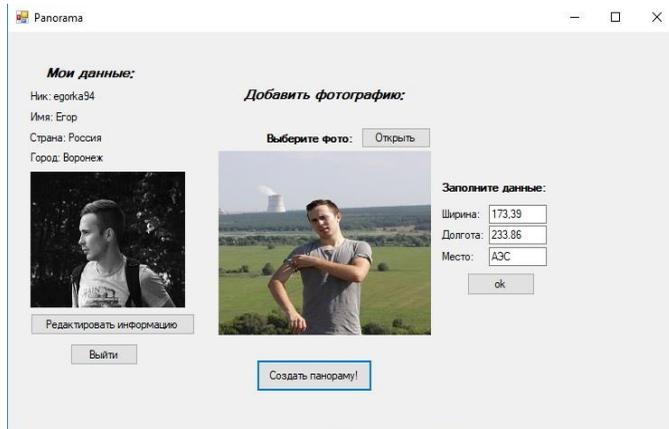


Рис. 3. Главная форма приложения

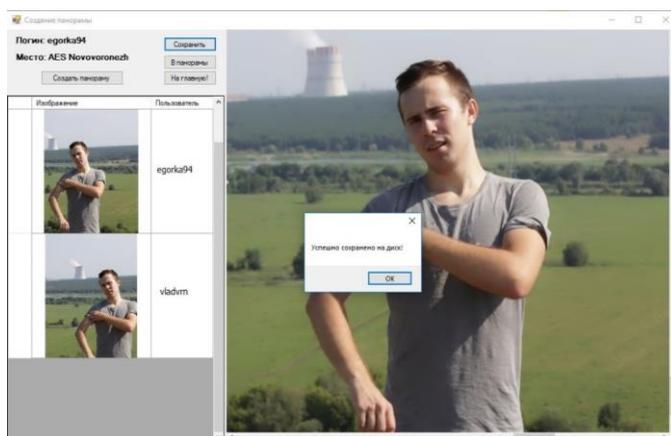


Рис. 4. Модуль по созданию панорам

Заключение

В рамках данной работы был создан прототип информационной системы для создания панорам. В ходе реализации были решены следующие задачи:

- реализована оптимизация алгоритмов обработки изображений, направленных на создание двумерных панорам;
- создана база данных, позволяющая обеспечить функциональность системы и хранение необходимой информации.

Список литературы

1. Bradski G. Learning OpenCV. Computer Vision with the OpenCV Library / G. Bradski, A. Kaehler. — Boston : O'Reilly Media, 2008. — 580 p.
2. Szeliski R. Image Alignment and Stitching / Szeliski R. — Redmond, WA : Microsoft Research, 2006. — 104 p.
3. Анисимов Б.В. Распознавание и цифровая обработка изображений / Б.В. Анисимов, В.Д. Курганов, В.К. Злобин. — М. : Высшая школа, 1983. — 294 с.
4. Справочное руководство по MySQL [Электронный ресурс] : интернет – справочник. – Режим доступа : <http://www.mysql.ru/docs/man/Reference.html>

РАСПРЕДЕЛЕНИЕ ПРОГРАММНЫХ ЗАДАЧ ПО ФИЗИЧЕСКИМ МОДУЛЯМ В АВИОНИКЕ

Д. А. Капустина

Студент

М. Г. Матвеев

Профессор

Введение

В развитии комплексов бортового оборудования (КБО) летательных аппаратов (ЛА) перспективным направлением является интегрированная модульная авионики (ИМА). Данный принцип используется в построении бортовых комплексов современных летательных аппаратов. ИМА базируется на концепции единой вычислительной платформы и сетевой архитектуры. Под ИМА понимается объединение различных ресурсов, таких как: источники ввода и вывода, питания, коммуникационные шины. Все эти составляющие предназначены для единой задачи – управления и контроля ЛА. ИМА позволила перейти от концепции «один модуль – одна функция» к – «один модуль – много функций» [1].

1. Постановка задачи

Бортовое оборудование вычислительной системы авионики представляет собой многомодульный вычислительный комплекс. В данной работе рассматривается многомодульный комплекс с общей операционной системой и памятью, имеющий M физических модулей одинаковой производительности.

Введем следующие ограничения и допущения:

- функции управления и контроля ЛА разделены на множество задач F и упорядочены или частично упорядочены;
- производительность всех физических модулей M одинакова и постоянна на протяжении выполнения задач F ;
- каждая задача обслуживается одновременно только на одном модуле на любом из них, доступном в момент времени t ;
- прерывания выполнения задач запрещены;
- после завершения выполнения задачи, освобожденные модули могут быть назначены на следующие задачи.

Систему ИМА можно представить в виде соответствия:

$$G: F \rightarrow M, \quad (1)$$

где F – это множество задач, а M – множество модулей.

В ряде основополагающих работ по данной тематике показано, что множество функций, представляющих собой программные задачи F удобнее всего изучать, представляя их сетевой структурой – графом, узлы которого являются задачами, а ребра показывают связи между ними.

Построение соответствия (1), представляет собой достаточно сложную техническую задачу. В некоторых простейших случаях данное соответствие можно определить «вручную», однако в общем случае этот процесс оказывается достаточно объемным, так что возникает естественная необходимость в компьютерных вычислениях.

Основной задачей данной работы является разработка и реализация компьютерного алгоритма, который позволит найти соответствие (1).

2. Алгоритм распределения программных задач по физическим модулям

Зададим множество задач $F = \{1, \dots, n\}$ в виде стрелочного графа. Фиктивные задачи с нулевой продолжительностью выполнения обозначаются штриховыми дугами. Так же заданы продолжительности обслуживания задач $p_i \geq 0$ для каждой задачи $i = 1, \dots, n$.

Задача минимизации времени выполнения всех элементов множества F представим в виде:

$$\begin{aligned} C_{\max} &\rightarrow \min \\ C_i &\leq C_j, \forall (i, j) \end{aligned} \quad (2)$$

где C_{\max} – время завершения всех задач F , и не нарушается условие следования; C_i – время завершения выполнения i -задачи.

Рассмотрим основные шаги алгоритма.

Шаг 1. Построение диаграммы Ганта.

Перед тем как начать распределять задачи, следует определиться с минимальным необходимым количеством модулей. Построенная диаграмма по исходному стрелочному графу позволяет определить необходимое количество модулей как $\max\{S(t)\}$, где $S(t)$ – количество параллельно выполняющихся задач на диаграмме в момент времени t . Построение диаграммы рассмотрено в [2].

Шаг 2. Построение допустимого распределения задач методом диспетчеризации List Scheduling (LS).

Теоретическая часть данного метода представлена в [3].

Данный метод требует соблюдения следующих ограничений:

1. В каждый момент времени $t \in [0, C_{\max})$ должно выполняться неравенство:

$$\sum_i^M \varphi_i(t) \leq M, \quad (3)$$

где $\varphi_i(t) = 1$, если требование i обслуживается в момент времени t и $\varphi_i(t) = 0$, в противном случае.

2. Условие следования:

$$S_i + p_i \leq S_j, \quad (4)$$

где $i \rightarrow j$ для $i, j \in F$, а S_i – время начала выполнения i -задачи.

Таким образом, выполняемые задачи в момент обслуживания должны быть полностью обеспечены ресурсами и не нарушены отношения предшествования между задачами.

Рассмотрим основные этапы алгоритма LS:

1. Составляем лист задач без предшественников EL ;

2. Выбираем i -задачу из листа и проверяем наличие свободного модуля для ее выполнения. Если есть свободный модуль, то задача может быть выполнена на отрезке: $[t, t + p_i]$, если нет – то ожидаем, когда один из модулей освободится в отрезок времени $[t_k, t_k + p_i]$, где $t_k > t$;

3. Назначаем выполнение задачи на модуль и резервируем ресурсы модуля: $M = M - 1, \tau \in [t_k, t_k + p_i]$;

4. Из списка исключаем i -задачу: $EL = EL / \{i\}$;

5. Добавляем в список EL задачи, для которых предшественники уже распределены на модули;

6. Если $EL \neq \emptyset$, то переходим к этапу 2, иначе - завершаем распределение.

Шаг 3. Построение дерева поиска оптимального распределения.

Очевидно, что построенное распределение на шаге 2 алгоритма не единственное, так как на этапе 2 не однозначно определяется i -задача из

списка EL . Следовательно, может быть построено множество расписаний, среди которых содержится расписание, являющееся решением (1).

Построенное дерево поиска оптимального расписания дает множество последовательностей задач. Полученные последовательности применяются для построения распределений с помощью метода LS.

3. Пример построения оптимального распределения программных задач по физическим модулям

В данном разделе работы рассмотрим пример построения оптимального распределения программных задач по физическим модулям.

Исходный граф F имеет вид, представленный на рис. 1. Вершины графа – задачи, значения над вершинами – время, необходимое на выполнения указанной задачи.

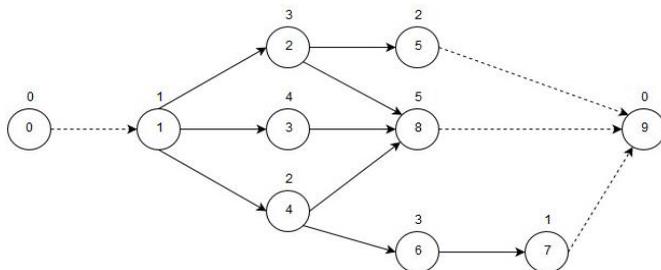


Рис. 1. Пример графа F

На рис. 2 представлена диаграмма Ганта, где видно, что для выполнения всех задач из F требуется не более трех физических модулей и 12 единиц времени.

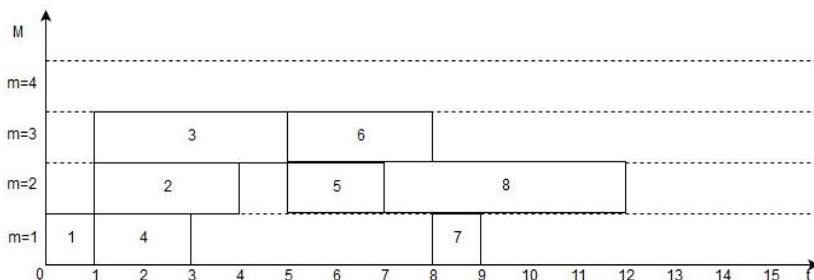


Рис. 2. Диаграмма Ганта: начальное распределение

Прежде чем распределять задачи методом LS, необходимо построить дерево поиска оптимального распределения.

Построенное дерево дало 505 последовательностей, не нарушающих условия следования. Фрагмент дерева представлен на рис. 3.

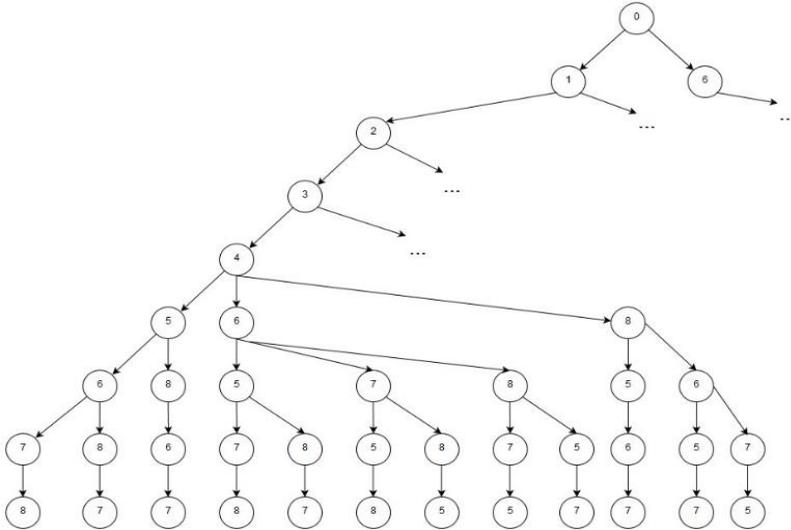


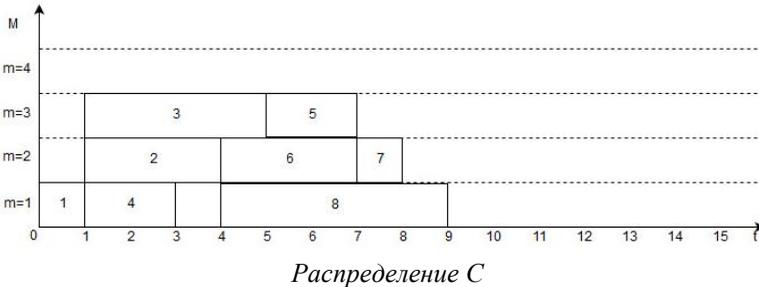
Рис. 3. Фрагмент дерева поиска оптимального распределения

Таким образом, все полученные последовательности применяются для построения распределений задач по модулям методом LS, среди которых выбирается распределение, удовлетворяющее начальному условию (1) путем поиска минимального C_{\max} .

$$C = \min\{C_{\max}\}, \tag{5}$$

где C – решение начального условия (1).

Распределение, которое удовлетворяет C , представлено на рис. 4.



Распределение C
Рис. 4. Диаграмма Ганта

Как видно из диаграммы, полученное распределение задач по модулям требует 9 единиц времени, что на 25% быстрее, чем при исходном распределении.

Заключение

Основным результатом данной работы стала разработка и компьютерная реализация алгоритма распределения программных задач по физическим модулям. С использованием разработанного алгоритма были проведены расчет минимального необходимого количества модулей и построение оптимального распределения задач.

Список литературы

1. Чуянов, Г. А. Перспективы развития комплексов бортового оборудования на базе интегрированной модульной авионики / Г.А. Чуянов, В.В. Косьянчук, Н.И. Сильвесюк // Известия ЮФУ. Технические науки. – 2013. – С. 55-62.
2. Матвеев, М. Г. Распределение функциональных задач в оптимизируемой многопроцессорной системе авионики / М.Г. Матвеев [и др.] // Вестник ВГУ. Серия : Системный анализ и информационные технологии. – Воронеж, 2017. – С. 26-34.
3. Лазарев, А. А. Теория расписаний. Задачи и алгоритмы / А.А. Лазарев, Е.Р. Гафаров. – М. : Издательство МГУ, 2011. – 222 с.

РАЗРАБОТКА ПРОГРАММНОГО МОДУЛЯ ДЛЯ ИССЛЕДОВАНИЯ И АНАЛИЗА ПРОЦЕССА ОРНШТЕЙНА- УЛЕНБЕКА

А. В. Караблин
Студент
Н. П. Стадная
Ассистент

Введение

Задача моделирования и прогнозирования актуальна всегда и наше время не является исключением.

Актуальность рассматриваемой тематики связана с широким применением в современных задачах процесса Орнштейна-Уленбека. Исторически данный процесс впервые был представлен в статье 1930 года в качестве модели скоростей частиц в процессе соударения с окружающими их частицами непосредственно в области физики [1].

На данный момент процесс Орнштейна-Уленбека широко используется не только в физике стохастических процессов, но и в анализе нестационарных рядов в экономических дисциплинах. В частности, его использование распространено в задачах финансовой математики, в таких, как ценообразовании опционов, оптимизации портфеля, теории рисков и др. [1-3]. Также процесс Орнштейна-Уленбека интересен тем, что является единственным стационарным гауссовским марковским процессом, поэтому изучение его различных модификаций и обобщений представляет интерес в математике [1].

1. Постановка задачи

В рамках данной работы предлагалось подготовить программный модуль, реализующий анализ процесса Орнштейна-Уленбека. В основе численных расчётов лежит уравнение Фоккера-Планка. Для достижения поставленной цели, необходимо было решить следующие задачи:

- построение теоретической модели уравнения Фоккера-Планка (УФП) и метода собственных функций при решении УФП;
- изучение метода собственных функций при решении УФП;
- разработка и представление численного решения УФП с зависящим от времени коэффициентом диффузии;

2. Построение математической модели

В основе анализа процесса Орнштейна-Уленбека лежит соответствующее уравнение Фоккера-Планка [1].

$$\frac{\partial p}{\partial t} = \frac{\partial}{\partial x}(kxp) + \frac{D(t)}{2} \frac{\partial^2 p}{\partial x^2}. \quad (1)$$

В (1) k – параметр, зависящий от свойств исследуемой системы, $D(t)$ – коэффициент диффузии. Представляет интерес исследовать решение этого уравнения при различных зависимостях $D(t)$, для того чтобы в дальнейшем можно было прогнозировать временную зависимость данного коэффициента для подобных процессов.

Для решения (1) можно использовать метод собственных функций. Он основан на знании решения уравнения (1) с $D(t) = D_0$, которое, в свою очередь, строится как разложение по системе собственных функций стационарного уравнения [1] и имеет вид (2):

$$p_b(x, t) = \sum_{n=0}^{\infty} A_n p_s(x) e^{-nkt} Q_n(x) \quad (2)$$

$p_b(x, t)$ – решение уравнения (1) с $D(t) = D_0$, $D(t) = D_0$, $Q_n(x)$ – система собственных функций стационарного уравнения, ортогональная с весом $p_s(x)$, A_n – коэффициенты, определяемые исходя из начального условия

$p(x,0)$, $p_s(x)$ – решение стационарного уравнения, описывающего процесс Орнштейна-Уленбека. Формулы (3-5), указанные ниже, определяют данные функции и коэффициенты:

$$Q_n(x) = \sqrt{\frac{1}{2^n \cdot n!}} \cdot H_n \left(x \sqrt{\frac{k}{D_0}} \right), \quad (3)$$

$$p_s(x) = \sqrt{\frac{\pi}{kD_0}} e^{-\frac{kx^2}{D_0}}, \quad (4)$$

$$A_n = \int_{-\infty}^{\infty} p(x,0) Q_n(x) dx. \quad (5)$$

В формуле (3) H_n – полиномы Эрмита. Условием для начального профиля $p(x,0)$ является стремление данной функции к 0 на бесконечности.

Так же, параметры в системе подбираются таким образом, чтобы основные изменения происходили в промежутке $(-1, 1)$ по x . Граничные условия для уравнения Фоккера-Планка задаются поглощающие, в связи с чем, мы не выходим за рамки $(-1, 1)$ по координате x .

При решении уравнения (1) теперь представляем $D(t)$ как $D_0 + D_1(t)$. Решение (1) ищем в виде суммы решений (2) и добавки, которая представляет собой разложение по системе собственных функций $Q_n(x)$:

$$p(x,t) = p_b(x,t) + \sum_{n=0}^{\infty} a_n(t) p_s(x) Q_n(x). \quad (6)$$

Подставляя (6) в (1) и учитывая ортогональность системы собственных функций, можно получить систему дифференциальных уравнений на коэффициенты $a_n(t)$. Учитывая тот факт, что все получающиеся интегралы вычисляются аналитически, удаётся получить данную систему, которая представляет собой бесконечную систему уравнений с зацепляющимися коэффициентами. Для реализации её численного решения необходимо оставлять конечное число уравнений. В разрабатываемом модуле число оставляемых уравнений равно 10 (как показали вычисления, большее количество уравнений не оказывают существенного влияния на результат). Далее, так как в начальный момент времени профиль остаётся тем же, то необходимо положить $a_n(0) = 0$ для любого n .

3. Анализ решения нестационарной задачи

В разрабатываемом программном модуле осуществляется задание параметров системы, функции $D(t)$, начального профиля; получение коэффициентов A_n , решение системы на $a_n(t)$, построение решения $p(x,t)$, визуализация полученных решений.

Реализация программного модуля осуществлялась в математическом пакете «WOLFRAM MATHEMATICA». Представим результаты работы модуля для начального профиля $p(x,t) = xe^{\frac{kx^2}{D_0}}$ (рис.1).

Справа на рис. 1 указан вид начального профиля на всей оси x , слева – в промежутке $(-1, 1)$.

Для данного начального профиля решение $p_b(x,t)$ будет иметь вид, представленный на рис. 2.

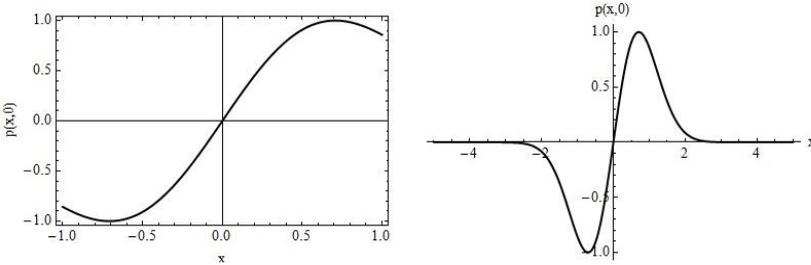
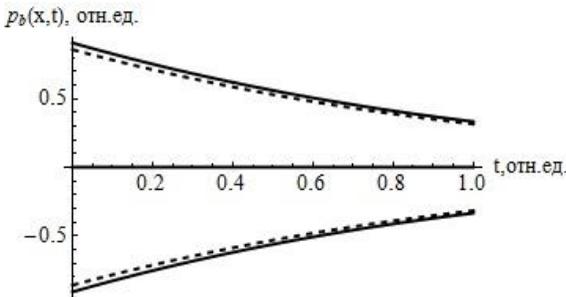


График слева построен в промежутке от -1 до 1, справа – на всей оси

Рис. 1. Начальный профиль



Значения указаны в относительных единицах. Жирными линиями обозначены значения при $x=-1$ и $x=1$, пунктирными – при $x=-0,5$, $x=0,5$

Рис. 2. Решение уравнения (1) с заданным начальным профилем и $D(t) = D_0 = 1$

Отметим, что наблюдается экспоненциальный спад со временем.

Далее, в модели рассматривается изменение данного решения для трёх различных зависимостей $D_1(t)$: экспоненциальной, гармонической и их произведения:

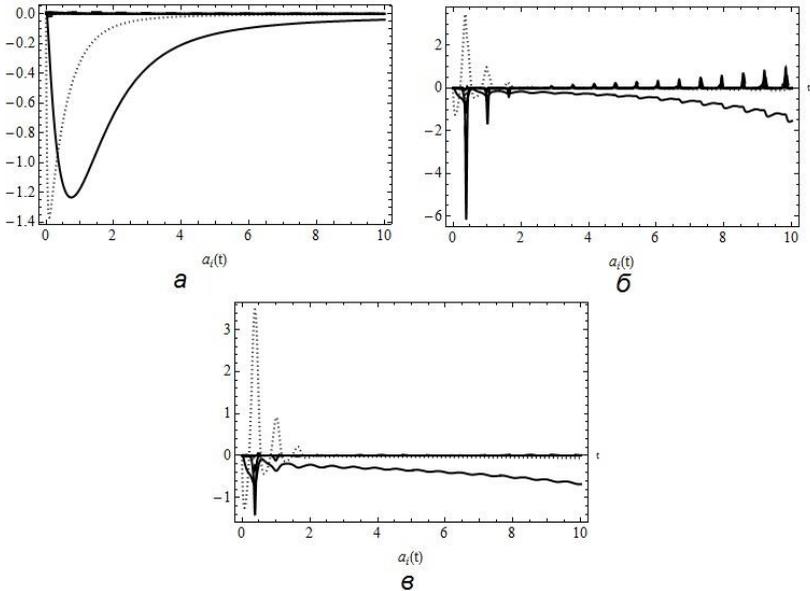
$$D_1(t) = \exp(-0,01t);$$

$$D_1(t) = \cos(10t)$$

$$D_1(t) = \exp(-0,01)\cos(10t)$$

Значение множителя в экспоненте и косинусе могут быть выбраны различными. Данные зависимости приведены для примера.

Решение системы уравнений даёт набор коэффициентов $a_n(t)$, поведение которых зависит от вида функции $D(t)$. Вид зависимостей представлен на рис. 3.



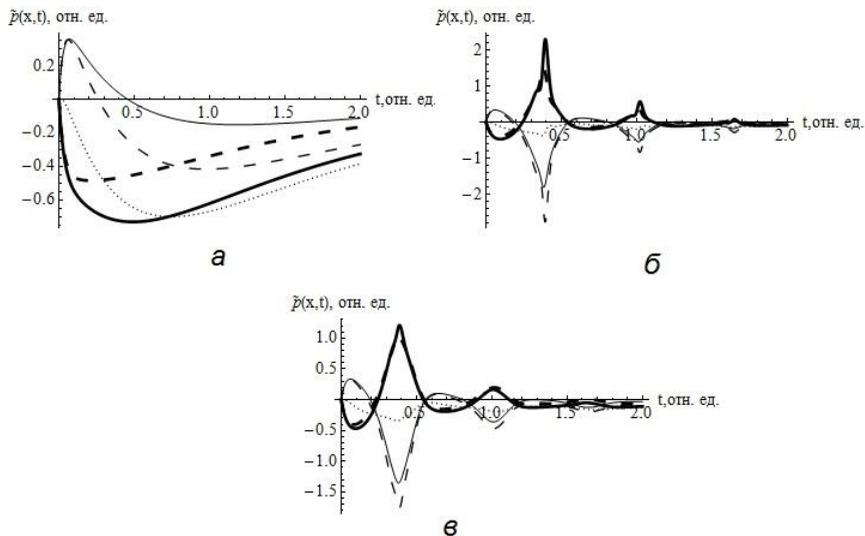
Показаны первые три коэффициента (жирная, полужирная и пунктирная линии соответственно)

Рис. 3. Временные зависимости коэффициентов $a_n(t)$ при различных видах функции $D_1(t)$: а – $\exp(-0,01t)$, б – $\cos(10t)$, в – $\exp(-0,01)\cos(10t)$

Из рис. 3 видим, что в случае экспоненциальной зависимости коэффициенты со временем затухают, а в случае гармонической зависимости наблюдается рост, при этом характер периодичности

сохраняется. Интересно, что в случае экспоненциальной функции ненулевые коэффициенты спадают в отрицательную область, образуя характерную «яму». Случай *в* периодичен за счёт косинуса, но амплитуда изменения не такая большая ввиду имеющейся экспоненциальной зависимости, дающей затухание.

На рис. 4 представлена добавка к решению $p_b(x,t)$, связанная с временной зависимостью $D_1(t)$.



$x=-1$ – нежирная сплошная линия, $x=-0,5$ – нежирная пунктирная линия,
 $x=0$ – тонкая пунктирная линия, $x=0,5$ – жирная пунктирная линия,
 $x=1$ – жирная сплошная линия

Рис. 4. Вид добавки к решению в уравнении (1) при различных видах функции $D_1(t)$: *а* – $\exp(-0,01t)$, *б* – $\cos(10t)$, *в* – $\exp(-0,01)\cos(10t)$

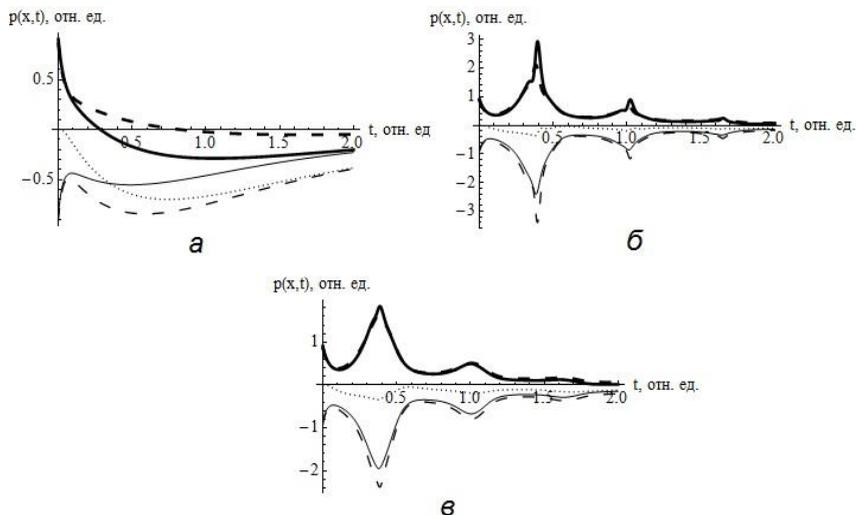
Тенденция стремления к 0 со временем сохраняется для всех трёх зависимостей, при этом необходимо обратить внимание на то, что наличие экспоненты в $D_1(t)$ в случае *в* сглаживает получаемые зависимости (по сравнению со случаем *б*).

Общее решение уравнения (1) будет иметь вид, представленный на рис. 5.

Заметим, что со временем функция стремится к 0, что соответствует ожиданиям. В случае, когда в $D_1(t)$ входит косинус, наблюдается

периодичность. Добавление экспоненты к косинусу сглаживает зависимость $p(x,t)$.

Подбор параметров позволяет проводить сравнение с реальными данными, добиваясь их схожести. Это важно для решения задач прогнозирования. Верный подбор временной зависимости позволяет анализировать и предсказывать результаты для важных физических, экономических, биологических процессов.



$x=-1$ – нежирная сплошная линия, $x=-0,5$ – нежирная пунктирная линия,
 $x=0$ – тонкая пунктирная линия, $x=0,5$ – жирная пунктирная линия,
 $x=1$ – жирная сплошная линия

Рис. 5. Решение уравнения (1) при различных видах функции $D_1(t)$:
 $a - \exp(-0,01t)$, $б - \cos(10t)$, $в - \exp(-0,01)\cos(10t)$

Заключение

Разработан программный продукт, реализующий численное решение нестационарного уравнения Фоккера-Планка с использованием метода собственных функций, который включает визуализацию полученных результатов. Варьируя параметры исследуемой системы и временную зависимость коэффициента диффузии $D(t)$, можно наблюдать зависимости, схожие с реальными зависимостями в различных диффузионных процессах, процессах в экономике, биологии и медицине. Данный продукт, таким образом, может быть полезен для специалистов вышеуказанных областей.

Список литературы

1. Гардинер К. В. Стохастические методы в естественных науках / К.В. Гардинер. – М. : Отдельное издание, 1986. – 528 с.
2. Ширяев А.Н. Теория случайных процессов / А.Н. Ширяев. – М. : Физматлит, 2005. – 402 с.
3. Кирьянова Л.В. Теория случайных процессов : курс лекций / Л.В. Кирьянова. – М. : Издательство МГСУ, 2016. – 96 с.

МОДИФИЦИРОВАННАЯ СИСТЕМА ЛОРЕНЦА

Е. А. Карпов

Студент

М. Е. Семенов

Профессор

Введение

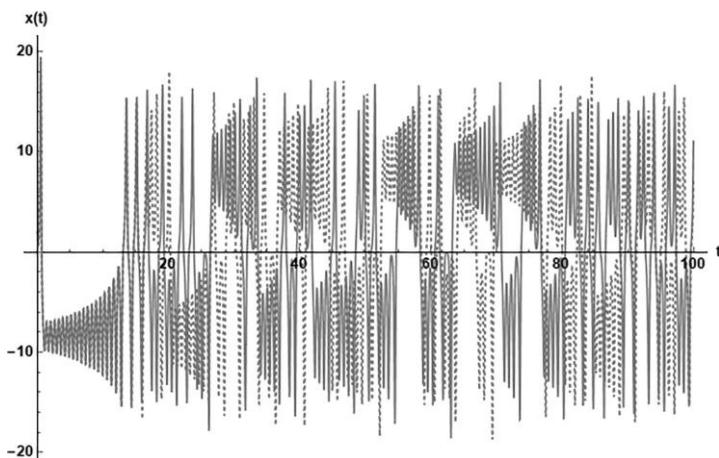
Работа, опубликованная в 1963 году американским ученым Эдвардом Лоренцом стала основополагающей для изучения теории хаоса. В этой работе моделировался процесс конвекции в подогреваемом снизу слое жидкости. Исходная система термогидродинамики рассматривалась в приближении Бусинесска (жидкость предполагалась слабо сжимаемой). Используя метод гармонического баланса и предполагая, что существенный вклад в решение вносят лишь первые гармоники Лоренца, получил следующую систему:

$$\begin{cases} \dot{x} = \sigma(y - x) \\ \dot{y} = x(r - z) - y, \\ \dot{z} = xy - bz \end{cases} \quad (1)$$

в которой x , y , z определяют скорость конвективных движений, распределение температуры по горизонтали и вертикали, соответственно. Параметры σ , r , b отвечают числу Прандтля (отношение коэффициента кинематической вязкости к коэффициенту теплопроводности), числу Релея (характеризует поведение жидкости под действием температуры) и геометрии конвекционных процессов. В ходе исследования ученый заметил отличительную особенность системы (1), а именно, если следить за эволюцией траекторий, с близкими начальными условиями расходятся с экспоненциальной скоростью (рис. 1) при этом, оставаясь в ограниченной области фазового пространства. Более того динамика изображающей точки

на фиксированной траектории очень «похожа» на реализацию случайного процесса.

Позднее выяснилось, что при помощи системы уравнений (1) можно описать и другие физические модели, например, одномодовую модель лазера, осциллятор с инерционным возбуждением, а также многие другие. На сегодняшний день существуют десятки тысяч работ, посвященных системе Лоренца.



Траектории $x(t)$ с близкими начальными условиями $(0.211, -0.2, 20)$ и $(0.212, -0.2, 20)$

Рис. 1. Две траектории $x(t)$

В работе [1] рассматривается система Лоренца дробного порядка. Было установлено, что хаос может существовать в системах дробного порядка менее 3. В работе [2] ставится задача о нахождении аттракторов комплексной модели Лоренца, которая получается путем расширения пространства действительных чисел в пространство комплексных. Производится доказательство наличия 2 аттракторов, не связанных с аттрактором Лоренца. Проводится анализ модифицированной системы при помощи Ляпуновских показателей, которые помогают определить тип наблюдаемой динамики. Также немаловажным результатом является то, что эти аттракторы имеют квазипериодический характер. Интересна работа [3], в которой приводятся алгоритмы для генерации псевдослучайных чисел, на основе модифицированной особым образом системе Лоренца. Был произведен анализ полученных алгоритмов с точки зрения безопасности и эффективности, и сделан вывод, что данный подход может быть использован в некоторых сферах деятельности, связанных с

передачей данных на расстояние. Но не обязательно рассматривать какие-то сложные модификация системы, можно проводить анализ и упрощенной модели как в работе [4], в которой с помощью теории устойчивости Ляпунова исследуются глобальные притягивающие множества. Доказывается, что построение функций Ляпунова применимо в данной модели, но только лишь с некоторыми поправками, а именно добавление к функции Ляпунова особого члена. Однако, не стоит забывать работу [5] которая была опубликована в далеком 1985 году, в которой приводится алгоритм для расчета спектра ляпуновских показателей, ведь этот алгоритм используется до сих пор. В [6] проводится интересное исследование и приводится алгоритм для шифрования изображения, его статистический анализ и заключение о безопасности предложенного. Отличные результаты также в работе [7], где сравнивалась классическая система Лоренца с системой, в которой переменные были комплексными. Сравниваются бифуркационные диаграммы, спектр ляпуновских показателей и сечения Пуанкаре. Результат показывает, что при комплексных переменных может быть достигнуто гиперхаотическое и квазипериодическое поведение.

1. Модель водяного колеса

В настоящей работе производился анализ модели водяного колеса. Данная модель представляет собой обычное колесо, на ободе которого закреплены чаши, с отверстием на дне, через которое выливается вода. Сверху на чаши льется поток воды и в зависимости от скорости этого потока колесо будет иметь различную динамику. При малом потоке будет наблюдаться неподвижное состояние колеса, при большем оно начнет двигаться в одну из сторон, однако, если поток будет значительным водяное колесо начнет демонстрировать хаотическую динамику (рис. 2).

Его учет модифицирует классическую систему Лоренца путем добавления в нее слагаемого, отвечающему моменту сухого трения, и тогда она принимает следующий вид:

$$\begin{cases} \dot{x} = \sigma(y - x) - M \cdot \text{sign}(x) \\ \dot{y} = x(r - z) - y \\ \dot{z} = xy - bz \end{cases}, \quad (2)$$

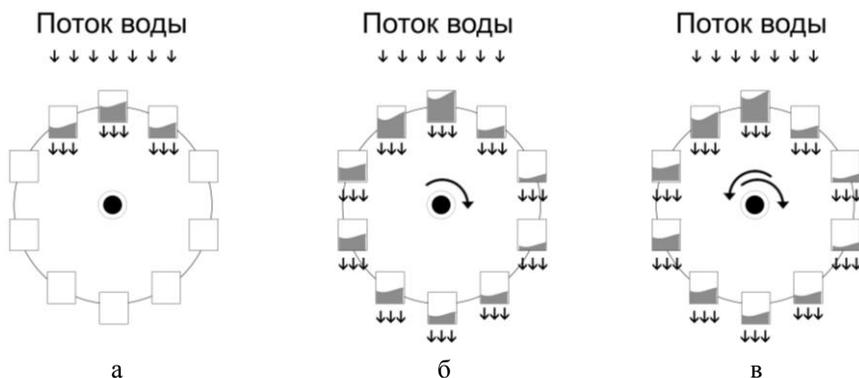


Рис. 2. Модель водяного колеса: *а* – при малом потоке воды колесо остается в покое, *б* – при большем потоке вращается с постоянной скоростью, *в* – при значительном потоке вращается хаотически

2. Результаты

Результатом настоящей работы является следующий факт: оказывается, что в независимости от начальных данных изображающая точка за конечное время оказывается в сколь угодно малой окрестности нуля. Рис. 3 иллюстрирует типичное поведение системы (2).

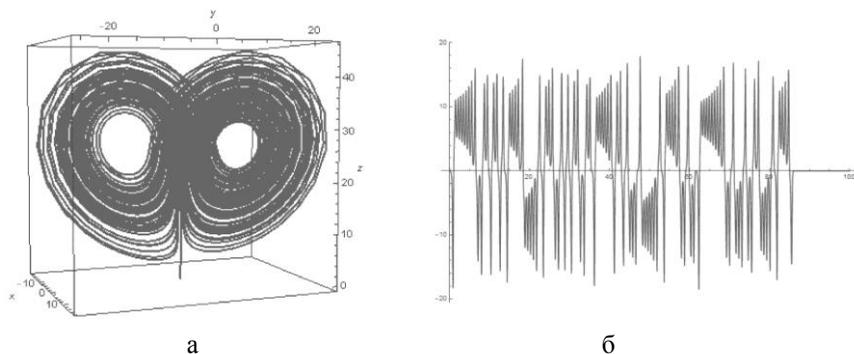


Рис. 3. Поведение системы (2): *а* – фазовый портрет модифицированной системы Лоренца при классических значениях параметров, *б* – зависимость первой координаты от времени

Как показывают численные эксперименты динамика системы соответствует хаотическому поведению в течении некоторого (зависящего от начальных условий) времени. При этом наличие хаотической динамики диагностируется не только посредством «наблюдения» фазового портрета,

но и спектр Ляпуновских показателей. Иными словами, в течение начального промежутка времени модифицированная система «не замечает» слагаемого обусловленного моментом сухого трения. Время хаотического режима также имеет стохастический характер и при равномерном распределении начальных условий имеет экспоненциальный характер.

Заключение

В рамках данной работы исследовалась модифицированная система Лоренца, полученная при рассмотрении модели водяного колеса.

Был проведен ее анализ, позволивший установить, что ноль – является абсолютно устойчивым, а попадание в ноль проявляет стохастические свойства и описывается экспоненциальным законом распределения.

Список использованной литературы

1. Xiang-Jun W. Chaos in the fractional-order Lorenz system / W. Xiang-Jun, S. Shi-Lei // *International Journal of Computer Mathematics*. – 2009. – Vol. 86. – Issue 7. – P.1274-1282.
2. Xavier Gómez-Mont. The Attractors in the Complex Lorenz Model. / Xavier Gómez-Mont, José-Job Flores-Godoy, Guillermo Fernández-Anaya // *3rd IFAC Conference on Analysis and Control of Chaotic Systems*. – 2012. – Vol. 45. – Issue 12. – P. 87-92.
3. Lynnyk V. Pseudo random number generator based on the generalized Lorenz chaotic system. / V. Lynnyk, N. Sakamoto, S. Čelikovský // *4th IFAC Conference on Analysis and Control of Chaotic Systems*. – 2015. – Vol. 48. – Issue 18. – P. 257-261.
4. Fuchen Zhang. Global dynamics for the simplified Lorenz system model / Fuchen Zhang, Yonglu Shu // *Applied Mathematics and Computation*. – 2015. – Vol. 259. – 2015. – P. 53-60.
5. Wolf A. Determining Lyapunov exponents from a time-series. / A. Wolf, J. B. Swift, H. L. Swinney, J. A. Vastano // *Physica D : Nonlinear Phenomena*. – 1985. – Vol. 16. – Issue 3. – P. 285-317.
6. Iqtadar Hussain. Construction of chaotic quantum magnets and matrix Lorenz systems S-boxes and their applications / Iqtadar Hussain, Amir Anees, Ali Hussain AlKhalidi, Abdulmohsen Algarni, Muhammad Aslam // *Chinese Journal of Physics*. – 2018. – Issue 3. – P. 216-242.
7. Moghtadaei M. Complex dynamic behaviors of the complex Lorenz system / M. Moghtadaei, M.R. Hashemi Golpayegani // *Scientia Iranica*. – 2012. – Vol. 19. – Issue 3. – P. 733-738.

Работа выполнена при финансовой поддержке РФФИ (проекты No. 16-08-00312, No. 17-01-00251).

ЗАДАЧА О ДВИЖЕНИЯХ ОДНОГО КЛАССА ПЛОСКИХ ФИГУР

Д. В. Козориз
Студент
А. В. Лобода
Профессор

Введение

«Задача о перемещении дивана» [1-3] была сформулирована в 1966 году. Её постановка: необходимо найти наибольшую площадь S жёсткого тела, которое может быть перемещено через Г-образный коридор, образованный тоннелями единичной ширины. Целью данной работы является получение оценок для площадей одного класса фигур (равнобедренных трапеций), удовлетворяющих условиям задачи о диване. Эта работа является обобщением более простого случая равнобедренного треугольника, рассмотренного ранее Грищенко С.И.

1. Простейшие свойства допустимых трапеций

Рассмотрим движение через угол равнобедренной трапеции (рис. 1). Будем обсуждать трапецию, большее основание которой превосходит 1, а значит, при движении фигура совершает поворот из начального горизонтального положения в конечное вертикальное.

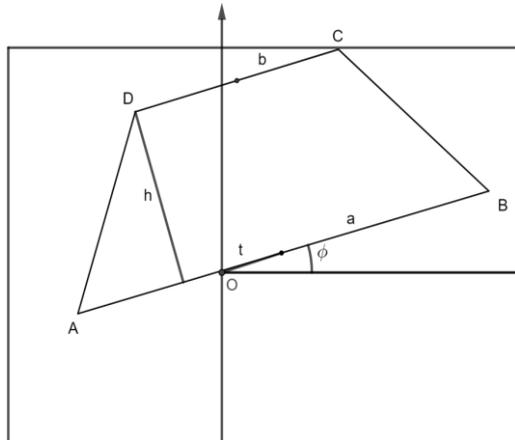


Рис. 1. Трапеция при прохождении через угол

Введем обозначения: a – половина большого основания, b – половина меньшего основания, h – высота; также введем параметры для описания движения: φ – угол наклона основания относительно горизонтали, $\varphi \in [0, \pi/2]$, t – смещение центра большого основания трапеции относительно центра координат (t положительно в том случае, если центр основания находится правее центра координат). Обсуждаемая трапеция движется таким образом, что в любой момент времени её нижнее основание проходит через центр координат.

Уравнения движения вершин трапеции:

$$\begin{cases} x_A = (t-a) \cdot \cos \varphi \\ y_A = (t-a) \cdot \sin \varphi \end{cases} \quad \begin{cases} x_C = (b+t) \cdot \cos \varphi - h \cdot \sin \varphi \\ y_C = (b+t) \cdot \sin \varphi + h \cdot \cos \varphi \end{cases}$$

$$\begin{cases} x_B = (a+t) \cdot \cos \varphi \\ y_B = (a+t) \cdot \sin \varphi \end{cases} \quad \begin{cases} x_D = (t-b) \cdot \cos \varphi - h \cdot \sin \varphi \\ y_D = (t-b) \cdot \sin \varphi + h \cdot \cos \varphi \end{cases}$$

Исходя из рис. 1, следует ввести следующие ограничения на движение точек A, B, C, D : $x_A \geq -1, y_B \leq 1, y_C \leq 1, x_D \geq -1$. Таким образом:

$$(a-t) \cdot \cos \varphi \leq 1 \tag{1}$$

$$(a+t) \cdot \sin \varphi \leq 1 \tag{2}$$

$$(b+t) \cdot \sin \varphi + h \cdot \cos \varphi \leq 1 \tag{3}$$

$$(b-t) \cdot \cos \varphi + h \cdot \sin \varphi \leq 1 \tag{4}$$

Понятно, что $a \leq \sqrt{2}$ (отрезок длины больше $2\sqrt{2}$ не может пройти через коридор), $h \leq 1$ (в силу начального расположения трапеции). Рассматривая неравенства (3) и (4) при $\varphi = \pi/4$, получим $b+h \leq \sqrt{2}$.

Назовем тройку a, b, h допустимой, если выполняется набор ограничений:

$$a \leq \sqrt{2}, h \leq 1, b+h \leq \sqrt{2}. \tag{5}$$

Рассмотрим теперь ограничения, связанные с параметром t . Получим оценки для t из неравенств (1)-(4):

$$\begin{aligned} t &\geq a - \frac{1}{\cos \varphi} & t &\geq b + \frac{h \cdot \sin \varphi - 1}{\cos \varphi} \\ t &\leq \frac{1}{\sin \varphi} - a & t &\leq \frac{1 - h \cdot \cos \varphi}{\sin \varphi} - b. \end{aligned}$$

Обозначим: $f_1(\varphi) = a - \frac{1}{\cos \varphi}$, $f_2(\varphi) = \frac{1}{\sin \varphi} - a$, $f_3(\varphi) = b + \frac{h \cdot \sin \varphi - 1}{\cos \varphi}$, $f_4(\varphi) = \frac{1 - h \cdot \cos \varphi}{\sin \varphi} - b$. Будем называть далее $f_1(\varphi)$, $f_3(\varphi)$ нижними, а $f_2(\varphi)$, $f_4(\varphi)$ - верхними оценочными функциями. Рассмотрим, как эти функции могут располагаться друг относительно друга при разных параметрах a, b, h , удовлетворяющих (5). Для каждой пары, состоящей из нижней и верхней функций, необходимым условием существования трапеции, проходящей через коридор и имеющей параметры a, b, h , является доминирование верхней функции при любом φ .

Лемма. Если параметры a, b, h допустимы, то при $\varphi \in (0, \pi/2)$ неравенства $f_1(\varphi) \leq f_2(\varphi)$, $f_3(\varphi) \leq f_4(\varphi)$ выполняются (рис. 2).

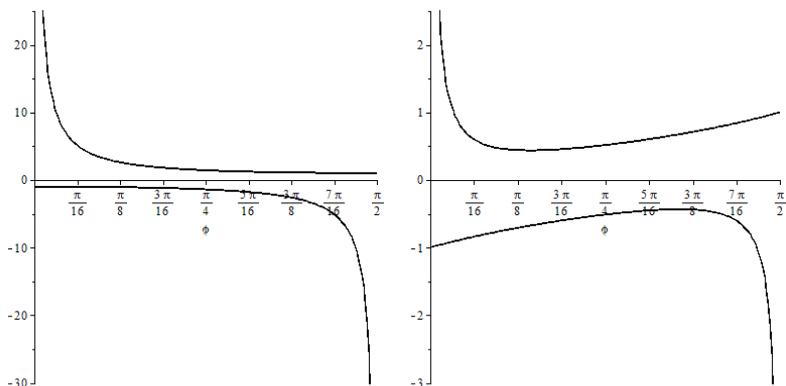


Рис. 2. Расположение пар функций $(f_1(\varphi), f_2(\varphi))$ и $(f_3(\varphi), f_4(\varphi))$

Доказательство леммы. Для пары $(f_1(\varphi), f_2(\varphi))$ проверим неравенство:

$$a - \frac{1}{\cos \varphi} \leq \frac{1}{\sin \varphi} - a.$$

Заметим, что оно равносильно следующему неравенству для a :

$$a \leq \frac{\sin \varphi + \cos \varphi}{\sin 2\varphi}.$$

Минимум функции в правой части этого неравенства равен $\sqrt{2}$, а значит, первая часть леммы верна.

Аналогично, для пары $(f_3(\varphi), f_4(\varphi))$ нужно проверить свое неравенство

$$b + \frac{h \cdot \sin \varphi - 1}{\cos \varphi} \leq \frac{1 - h \cdot \cos \varphi}{\cos \varphi} - b$$

или, что равносильно, неравенство:

$$b \cdot \sin 2\varphi + h \leq \cos \varphi + \sin \varphi.$$

После замены $\theta = \varphi - \frac{\pi}{4}$ ($\theta \in [-\frac{\pi}{4}, \frac{\pi}{4}]$) оно переписывается в форме

$$2 \cdot b \cdot \cos^2 \theta - \sqrt{2} \cos \theta + h - b \leq 0. \quad (6)$$

Обозначая $\cos \theta$ через μ , заметим, что график функции $2 \cdot b \cdot \mu^2 - \sqrt{2} \mu + h - b$ представляет собой параболу. На концах промежутка $[1/\sqrt{2}, 1]$ эта функция принимает неположительные значения и является выпуклой вниз. Следовательно, условие ее неположительности выполняется на всём промежутке, а левая часть проверяемого неравенства (6) не превышает нуля при всех $\theta \in [-\frac{\pi}{4}, \frac{\pi}{4}]$. Лемма доказана.

2. Нарушение необходимых условий

Ситуация с парами $(f_3(\varphi), f_2(\varphi))$ и $(f_1(\varphi), f_4(\varphi))$ сложнее.

Например, при выполнении условий $a = \sqrt{2}, b + h = \sqrt{2}$ графики функций $f_3(\varphi)$ и $f_2(\varphi)$ имеют разные наклоны в точке $\varphi = \pi/4$ (в силу различия их производных). Значит, они пересекаются, и на некотором промежутке $f_3(\varphi)$ доминирует над $f_2(\varphi)$ (рис. 3).

Полное исследование этой ситуации пока не реализовано.

Обсудим некоторые ограничения на наборы a, b, h , при которых нарушается необходимое условие прохождения трапеции через угол

$$b + \frac{h \cdot \sin \varphi - 1}{\cos \varphi} \leq \frac{1}{\sin \varphi} - a. \quad (7)$$

Освободимся в (7) от знаменателей

$$h - 2 \cos \varphi - 2 \sin \varphi + h \cdot \cos 2\varphi + (a + b) \cdot \sin 2\varphi \leq 0 \quad (8)$$

и обозначим через $R(\varphi)$ тригонометрический многочлен из левой части (8). Заметим, что на краях отрезка $\varphi \in [0, \pi/2]$ многочлен $R(\varphi)$ принимает неположительные значения при любых допустимых a, b, h , а при малых значениях этих параметров $R(\varphi)$ отрицателен на всем отрезке.

При некоторых критических наборах параметров график $R(\varphi)$ касается горизонтальной оси своим максимумом. При таком наборе параметров трапеция еще проходит через коридор, но дальнейший подъем графика $R(\varphi)$ (если его можно обеспечить соответствующим изменением параметров) запретит такое прохождение.

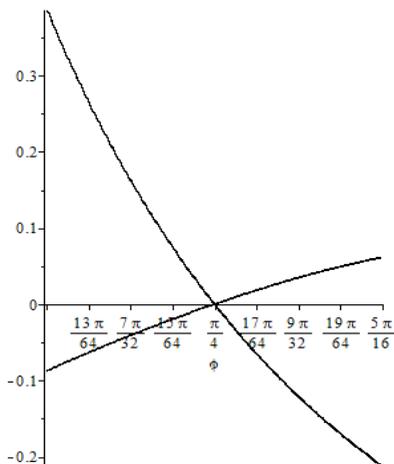


Рис. 3. Расположение пары функций ($f_3(\varphi)$, $f_2(\varphi)$)

Для интересующих нас критических наборов выполняются 2 ограничения:

$$\begin{cases} R(\varphi) = 0 \\ R'(\varphi) = 0 \end{cases}$$

Решив эту систему относительно $\sin 2\varphi$, $\cos 2\varphi$, получим:

$$\cos 2\varphi = \frac{-a \cdot (\sin \varphi + \cos \varphi) + b \cdot (\sin \varphi - \cos \varphi) - 2h \cdot (\sin \varphi + \cos \varphi) + h^2}{(a+b)^2 + h^2};$$

$$\sin 2\varphi = \frac{2a \cdot (\sin \varphi + \cos \varphi) + 2b \cdot (\sin \varphi - \cos \varphi) + h \cdot (\sin \varphi - \cos \varphi) - h(a+b)}{(a+b)^2 + h^2}.$$

Для полученных значений $\sin 2\varphi$, $\cos 2\varphi$ должно выполняться основное тригонометрическое тождество, отсюда:

$$b \sin \varphi \cos \varphi - 4 \sin \varphi \cdot h - 4 \cos \varphi \cdot h - (a+b)^2 + 5 = 0,$$

$$3 \sin 2\varphi - 4h(\sin \varphi + \cos \varphi) - (a+b)^2 + 5 = 0,$$

Введем замены $\theta = \varphi - \frac{\pi}{4}$, $\theta \in [-\frac{\pi}{4}, \frac{\pi}{4}]$; $A = a + b$:

$$3\cos 2\theta - 4\sqrt{2}h \cdot \cos \theta - A^2 + 5 = 0,$$

$$6\cos^2 \theta - 4\sqrt{2}h \cdot \cos \theta - A^2 + 2 = 0. \quad (9)$$

При отсутствии (при некотором наборе параметров) решений у полученного квадратного уравнения относительно $\cos \theta$ имеем трапецию, проходящую через угол.

Первый случай отсутствия решений – отрицательный дискриминант у квадратного уравнения

$$6v^2 - 4\sqrt{2}h \cdot v - A^2 + 2 = 0, \quad (10)$$

отвечающего (9), т. е.

$$8h^2 + 6A^2 - 12 < 0 \quad \text{или} \quad \frac{2}{3}h^2 + \frac{1}{2}A^2 < 1. \quad (11)$$

Уравнение $\frac{h^2}{1.5} + \frac{A^2}{2} = 1$ является уравнением эллипса; допустимая область изображена на рис. 4. Оценка площади в этом случае составляет $S \leq \sqrt{3}/2$.

Другой возможный случай – уравнение (10) имеет корни v_1, v_2 , но при некоторых A, h , значения v_1, v_2 выходят за промежуток $[1/\sqrt{2}, 1]$.

$$v_1 = \frac{\sqrt{2}}{3}h - \frac{1}{6}\sqrt{8h^2 + 6A^2 - 12}, \quad v_2 = \frac{\sqrt{2}}{3}h + \frac{1}{6}\sqrt{8h^2 + 6A^2 - 12}.$$

Для первого корня неравенство $v_1 < 1/\sqrt{2}$ выполняется всегда – такой корень не даёт информации. Возможны также случаи $v_2 > 1$ и $v_2 < 1/\sqrt{2}$. Во втором из них получается ограничение

$$A^2 + 4h - 5 < 0 \quad (12)$$

и несколько лучшая оценка $S < 1.07582877$. Ясно, однако, что максимальная площадь трапеции, проходящей через угол, не может быть меньше аналогичной площади треугольника (достигающей величины ≈ 1.319), являющегося частным случаем трапеции.

Численные эксперименты с неравенством (8) приводят к оценке ≈ 1.402 для площади трапеции, проходящей через угол. Однако точное формульное описание этой оценки пока не получено.

Такие же оценки можно получить при рассмотрении пары функций ($f_1(\varphi)$, $f_4(\varphi)$).

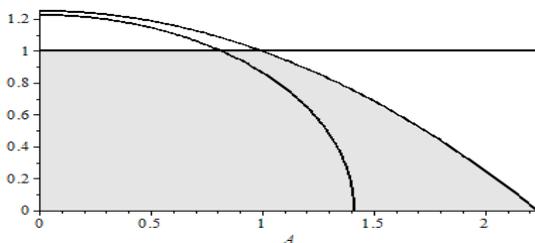


Рис. 4. Области, удовлетворяющие неравенствам (11), (12)

Заключение

Описан подход к исследованию известной оптимизационной задачи, связанный с движением многоугольных фигур. Выписаны и частично исследованы необходимые условия прохождения равнобоких трапеций через угол. Получены некоторые оценки для площадей таких фигур.

Список литературы

1. Gerver, J. L. On Moving a Sofa Around a Corner / J. L. Gerver // *Geometriae Dedicata*. – 1992. – V.42. – №3. – P. 267-283.
2. Kallus, Y. Improved upper bounds in the moving sofa problem / Y. Kallus, D. Romik // arXiv:1706.06630 [math.MG]
3. Шклярский, Д. О. Геометрические оценки и задачи из комбинаторной геометрии / Д. О. Шклярский, Н. Н. Ченцов, И.М. Яглом. – М. : Наука, 1974. – 384с.

ОПТИМИЗАЦИЯ МЕТОДА ОБРАТНОЙ ТРАССИРОВКИ ЛУЧЕЙ С ПОМОЩЬЮ VNH-ДЕРЕВА

Д. А. Королёв

Студент

Д. И. Соломатин

Старший преподаватель

Введение

Трассировка лучей (Ray tracing; рейтрейсинг) – технология построения изображения трёхмерных моделей в компьютерных программах, при которых отслеживается обратная траектория распространения луча (от экрана к источнику) [1].

Визуализация, выполненная с помощью трассировки лучей, имеет ряд преимуществ по сравнению с растеризацией:

- физически более корректное изображение, особенно в части теней и бликов, отражений предметов, преломлений света и других оптических эффектов (зависит от реализации);
- возможность получить максимально фотореалистичное изображение;
- элегантность и сравнительная простота реализации (по отношению к ухищрениям, применяемым в растеризации, чтобы получить имитацию оптических эффектов).

Однако при множестве несомненных преимуществ метод обратной трассировки лучей имеет и существенный недостаток – существенно более низкую по сравнению с растеризацией производительность. В данной статье рассматривается индексация объектов в пространстве с помощью BVH-дерева в качестве одно из классических приемов, применяемых в реализациях метода обратной трассировки лучей для увеличения производительности.

1. Суть метода обратной трассировки лучей

При построении изображений методом обратной трассировки лучей виртуальный луч испускается из начальной точки камеры через каждый пиксель изображения и осуществляется поиск пересечения луча с объектом. Если луч попадает в объект то, из точки пересечения луча с объектом испускается новый луч в сторону источника света. В зависимости от того сможет ли луч добраться до источника света зависит интенсивность цвета пикселя. Иллюстрация метода приведена на рис. 1.

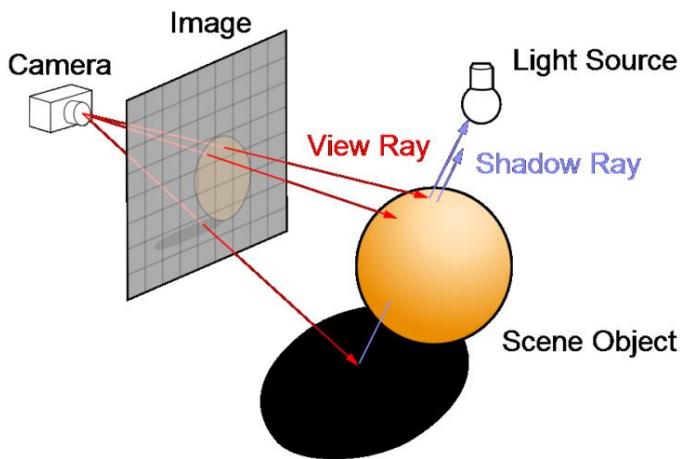


Рис. 1. Метод обратной трассировки лучей

С учетом материалов объектов, которые при отражении света могут последний рассеивать, приходится обрабатывать не один, а множество лучей (могут применяться различные стратегии). Отображаемая сцена также может быть сложной и состоять из миллионов треугольников. Очевидно, что в такой ситуации простейший перебор треугольников в поисках пересечения с каждым лучом будет занимать огромное количество времени.

2. Индексация объектов с помощью BVH-дерева

В простейшей реализации метода трассировки лучей основные затраты производительности будут уходить на поиск пересечения луча с одним из примитивов сцены, как правило, треугольников. При этом возможна ситуация, когда очередной луч вообще ни один объект сцены не пересекает. Для того, чтобы при поиске пересечения не перебирать все примитивы сцены, можно их индексировать с помощью BVH-дерева.

Исторически BVH (Bounding Volume Hierarchy – иерархия ограничивающих объемов) деревья используются для расчета столкновений в физике. Однако в последнее время BVH активно стараются задействовать в рейтрейсинге в связи с тем, что в анимированных сценах BVH можно быстрее перестраивать, и, как правило, можно перестраивать не все дерево [2].

В простейшем случае в качестве ограничивающих объемов в BVH-деревьях используются параллелепипеды, все стороны которых параллельны осям координат. Такое BVH-дерево классифицируется как AABB (Axis Aligned Bounding Box).

При построении двоичного AABB-дерева на первом шаге берется параллелепипед, вмещающий в себя все объекты сцены. Первоначальный параллелепипед делится на два таким образом, чтобы каждый из новых параллелепипедом включал примерно одинаковое кол-во примитивов, содержащихся в родительском параллелепипеде. Данный алгоритм повторяется рекурсивно для созданных параллелепипедов до тех пор, пока на нижнем уровне каждый параллелепипед не будет содержать только один примитив сцены.

Однако не все так просто: типичной является ситуация, когда невозможно равномерно разделить примитивы сцены между параллелепипедами так, чтобы все примитивы попали только в один из новых параллелепипедов. В этом случае возможны две стратегии:

- разделение пространства (Spatial Split), когда происходит расширение одного из параллелепипедов так, чтобы объекты на границах полностью включить в него;

- разделение объектов (Object Split), когда «спорные» объекты включаются в каждый из двух новых параллелепипедов (Spatial Split).

Принцип построения двоичного BVH(AABB)-дерева для двумерного случая показан на рис. 2.

На рис. 3 показаны несколько первых уровней BVH-дерева, построенного для 3D-сцены, содержащей фигуру быка и стеклянный шар. В данной сцене фигура быка состоит множества полигонов, а шар представляет собой один примитив, поэтому вокруг шара описан куб, который дальше на параллелепипеда не делится.

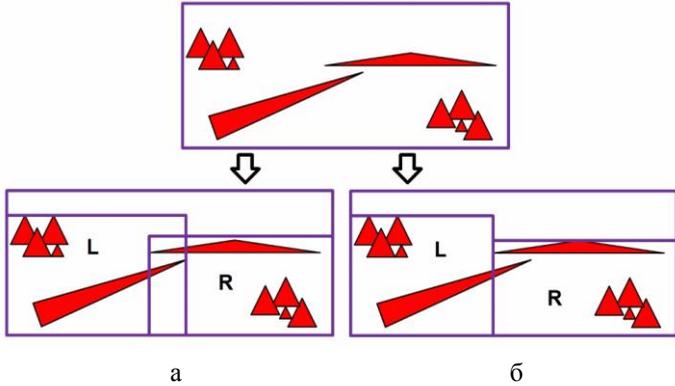


Рис. 2. Построение BVH-дерева: а – разделение пространства, б – разделение объектов

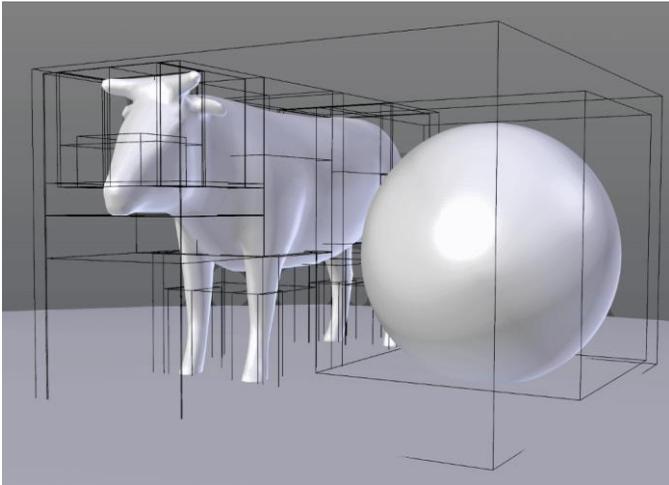


Рис. 3. BVH-дерево для 3D-сцены (несколько первых уровней)

В методе трассировки лучей при поиске пересечения луча с примитивом построенное BVH-дерево позволяет существенно сократить перебор объектов, т. к. очевидно, что для каждого параллелепипеда, через который луч не проходит, перебирать содержащиеся в нем примитивы не нужно. В большинстве случаев BVH-дерево в качестве индекса позволяет искать пересечения с объектами за логарифмическое от количества примитивов сцены время.

3. Реализация и тестирование

В рамках данной работы на языке C++ был реализован прототип алгоритма обратной трассировки лучей с применением BVH-дерева для ускорения поиска пересечений.

На рис. 4 приведен результат визуализации сцены из рис. 3, полученный с помощью данного прототипа. Фигура быка на данной сцене состоит из 471508 полигонов, материал – металл с небольшой шероховатостью (0.1). Шар – один примитив, материал – стекло. Видно, что через шар просматривается фигура быка, но шар вносит преломление в распространение лучей. Сцена освещена диффузным светом, а также точечным источником света, из-за которого появляется тень от объектов на полу.

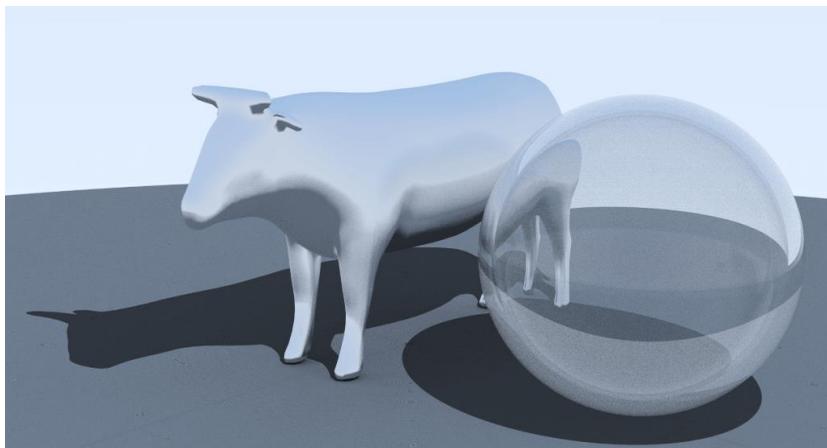


Рис. 4. Результат визуализации

Данная сцена в разрешении 1920 x 1080 при 100 лучах на пиксель на процессоре i7 4790k (4 ядра, 8 потоков, в коде используется OpenMP и флаг оптимизации для процессоров поколения haswell) просчитывается за 40 секунд. Без использования BVH-дерева визуализация занимает около 8 часов.

Заключение

Реализованный прототип алгоритма обратной трассировки лучей в настоящий момент достаточно примитивен, по качеству получаемых результатов несопоставим с применяемыми на практике системами визуализации. Однако в силу простоты данный прототип может быть легко доработан и использован в дальнейших экспериментах и исследованиях.

Список литературы

1. Трассировка лучей [Электронный ресурс]. – Режим доступа : https://ru.wikipedia.org/wiki/Трассировка_лучей
2. BVH [Электронный ресурс]. – Режим доступа : <http://ray-tracing.ru/articles184.html>

РАЗРАБОТКА АЛГОРИТМОВ ОЦЕНКИ ХАРАКТЕРА СОДЕРЖИМОГО РАСТРОВЫХ ИЗОБРАЖЕНИЙ

В. С. Лыгин

Студент

М. А. Дрюченко

Доцент

Введение

В данной работе рассматривается задача быстрой оценки и распознавания типа содержимого тематических изображений различных классов. Данная задача может найти свое применение в качестве одного из этапов в схеме обработки изображений, предусматривающей автоматическую высокоуровневую обработку с точным распознаванием содержимого данных.

Для решения данной цели были поставлены следующие задачи:

- анализ различных алгоритмов обработки изображения.
- анализ и сравнение современных подходов к распознаванию типа содержимого растровых изображений.
- исследование показателей качества работы представленных в данной работе алгоритмов и анализ их эффективности.

1. Анализ типа содержимого изображений с использованием данных морфологической гранулометрии

Гранулометрия часто применяется в медицинских и промышленных приложениях, связанных с анализом содержимого изображений, например, позволяет автоматически выделять различные объекты и оценивать их

процентный состав в зависимости от площади [1]. В данной работе анализировался характер содержимого изображений микроэлектроники (рис. 1), содержащих проводники и площадные объекты различной топологии.

Метод морфологической гранулометрии заключается в применении к исходному изображению I или его сглаженной копии операции морфологического размыкания M_{open} с увеличивающимся размером структурообразующего элемента (по умолчанию квадрат). Для анализа изображений помимо размыкания использовалась операция морфологического градиента M_{grad} . На каждой итерации размыкания и градиента вычислялись математическое ожидание и среднее квадратичное отклонение:

$$m_i^{(o,g)} = (WH)^{-1} \sum_{x,y=1}^{W,H} I_i^{(o,g)}(x,y), \quad \sigma_i^{(o,g)} = \sqrt{(WH)^{-1} \sum_{x,y=1}^{W,H} (I_i^{(o,g)}(x,y) - m_i^{(o,g)})^2},$$

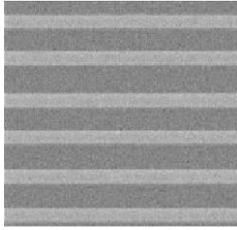
где $I_i^{(o)} = M_{open}(I, r_i)$, $I_i^{(g)} = M_{grad}(I, r_i)$; $r_i = 3 + 2(i - 1)$, $i = \overline{1, N}$ – размер структурообразующего элемента, (по умолчанию количество шагов $N = 21$).

Далее для полученных последовательностей значений математических ожиданий и СКО, описывающих кривые распределения объектов по ширине (площади) и градиенту яркости, применялся дискретный аналог производной первого порядка:

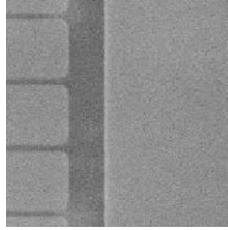
$$m_i^{(o,g)} = \left| m_{i+1}^{(o,g)} - m_i^{(o,g)} \right|, \quad \sigma_i^{(o,g)} = \left| \sigma_{i+1}^{(o,g)} - \sigma_i^{(o,g)} \right|, \quad i = \overline{1, N-1},$$

в результате чего формировались последовательности значений, характеризующие уменьшение ширины (площади) и градиента яркости между последовательными морфологическими преобразованиями. Обладая хорошей информативностью относительно характера содержимого анализируемых изображений кривые $m^{(o)}, \sigma^{(o)}, m^{(g)}, \sigma^{(g)}$ достаточно успешно применялись в подзадачах, связанных с автоматизацией настройки сценариев предобработки и настройки параметров алгоритма классификации [2, 3]. На рис. 2, 3 представлены примеры гранулометрических кривых, полученных на типовых изображениях микроэлектроники, содержащих объекты двух типов. Кривые для изображений с проводниками обозначены пунктиром, кривые для площадных – сплошной линией.

График СКО по операции градиента показывает изменения количества перепадов яркости или количество границ, переходов между текстурами. Для проводников характерно большое число перепадов яркости при малых размерах структурного элемента, для площадных объектов же при больших размерах.



а



б

а – содержит проводники, б – содержит площадные объекты

Рис. 1. Примеры изображений микроэлектроники

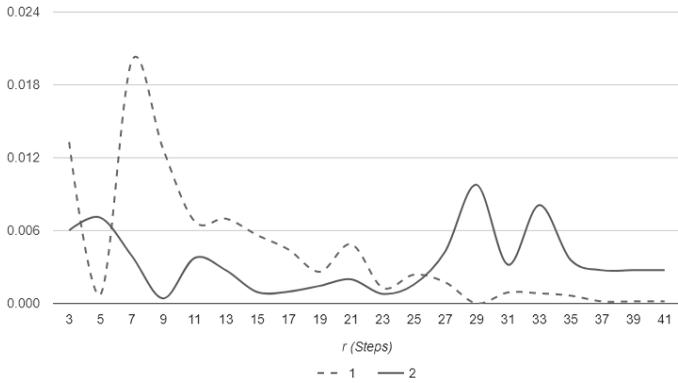


Рис. 2. SKO по операции градиента

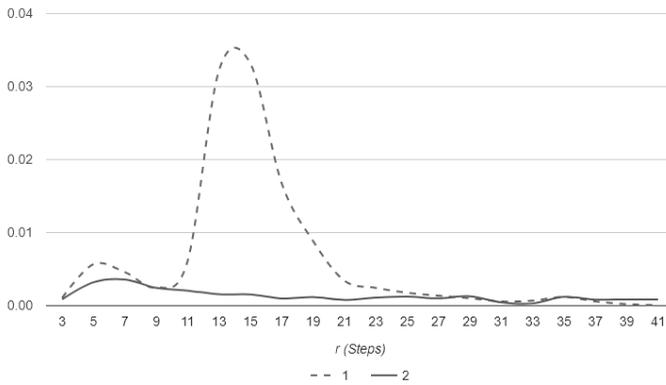


Рис. 3. Математическое ожидание по операции размывания

График математического ожидания по размыканию показывает относительную площадь объектов (относительно всего кадра) с шириной больше чем размер структурного элемента.

В качестве классификатора, оценивающего тип содержимого кадров по результатам гранулометрической обработки, рассматривалась нейронная сеть прямого распространения – многослойный перцептрон [4]. Архитектура такой сети представлена на рис. 4. Обучающие данные НС-классификатора определяются в виде $Z = \{z_1, z_2, \dots, z_k\}$, где $z_i = (m_1^{(o)}, \dots, m_n^{(o)}, \sigma_1^{(o)}, \dots, \sigma_n^{(o)}, m_1^{(g)}, \dots, m_n^{(g)}, \sigma_1^{(g)}, \dots, \sigma_n^{(g)})^T$ – вектор, составленный из данных морфологической гранулометрии, вычисленных для тестового изображения I или его фрагмента; n – количество различных значений радиуса структурообразующего элемента в морфологических операциях (значения по оси X на рис. 2, 3); k – объем обучающей выборки. Ожидаемые на выходе сети значения определяются целевой матрицей P , где каждый компонент представляет из себя бинарный вектор $(p_1, p_2, \dots, p_M)^T$ из M элементов; $M-1$ элемента вектора принимают значение 0, а один из элементов, индекс которого характеризует заданный тип содержимого кадра, значение 1.

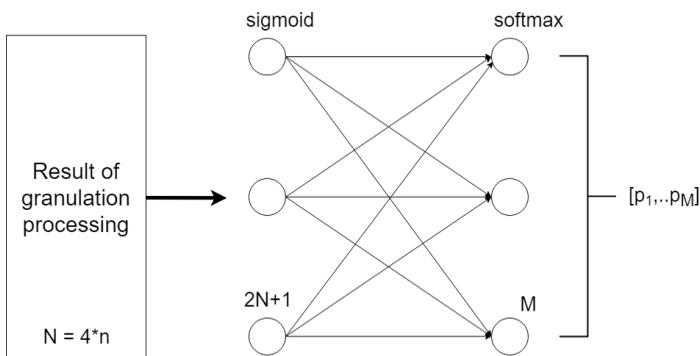


Рис. 4. Архитектура многослойного перцептрона

2. Анализ типа содержимого изображений с использованием методов глубокого обучения

Второй подход к оценке типа содержимого тематических растровых изображений основан на использовании методов глубокого обучения и не предполагает специальной подготовки исходных данных, заключающийся в формировании векторов признаковых описателей. В качестве исходных данных в алгоритме классификации напрямую используются фрагменты

растра фиксированного размера, которые подаются на вход сверточной сети [5]. По аналогии с сетями прямого распространения на выходе сверточной сети ожидаются бинарные вектора $(p_1, p_2, \dots, p_M)^T$, характеризующие принадлежность фрагмента к одному из M классов.

В данной работе рассматривалась сеть из трех сверточных слоев (рис. 5), на вход которой подавались фрагменты изображения размером 200x200 пикселей, которые далее сворачивались до вектора признаков размером 1x1x256 элементов. На выходе сети формировался вектор вероятностей классов.

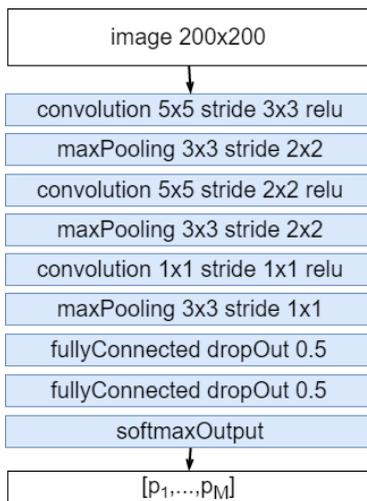


Рис. 5. Архитектура сверточной сети

3. Сравнение реализованных алгоритмов

Сравнение алгоритмов проводилось с использованием реальных изображений фрагментов кристаллов микроэлектроники разрешением 4000x4000 пикселей. Обучающая выборка включала 50 изображений, тестовая – 100 кадров. Для обучения сетей использовались 15000 фрагментов размера 200x200 пикселей, для тестирования – 30000 фрагментов аналогичного размера. Отметим, что при формировании выборок растровые фрагменты подвергались дополнительной аугментации. Тестирование производилось кросс-валидацией и на тестовых данных не входящих в обучающую выборку. Результаты тестирования представлены в табл. 1 и 2. Наилучшая точность классификации типа содержимого кадров получается при использовании сверточной нейронной сети. Такой результат достигается за счет того, что

в результате работы сверточных слоев получается более информативная карта признаков, чем вектор из данных морфологической гранулометрии.

Таблица 1

Результаты кросс-валидации

	MLP	CNN
AUC	0.994	0.987
Ошибки первого рода	0,043	0.079
Ошибки второго рода	0,0066	0.0027

Таблица 2

Результаты тестирования на новых изображениях

	MLP	CNN
AUC	0.964	0.978
Ошибки первого рода	0.0143	0.0121
Ошибки второго рода	0.153	0.137

Заключение

В рамках данной работы были реализованы два варианта обучаемых алгоритмов оценки содержимого растровых изображений – алгоритм, использующий сети прямого распространения (MLP) и данные морфологической гранулометрии, а также алгоритм, использующий глубокие нейронные сети сверточного типа (DL) и фрагменты растров в качестве обучающих данных.

На языке Java разработан библиотечный модуль, реализующий алгоритмы классификации изображений по типу их содержимого. При создании компонента использовался фреймворк DL4J [6]. С использованием тестовой выборки изображений микроэлектроники проведено исследование эффективности предложенных алгоритмов в части оценки точности классификации типа содержимого. Наилучшие результаты были получены для сверточных сетей.

Список литературы

1. Гонсалес Р. Цифровая обработка изображений / Р. Гонсалес, Р. Вудс. – М. : Техносфера, 2005. – 1072 с.
2. Желтов, С.Ю. Разработка теории, методов и алгоритмов машинного зрения в задачах обнаружения объектов: диссертация доктора технических наук: 05.13.01 / Желтов Сергей Юрьевич. – М., 2002. – 338 с.
3. Sera J. Image Analysis and Mathematical Morphology / J. Sera. – Orlando, FL, USA : Academic Press, 1982. – 621 p.

4. Fausett Laurene V. Fundamentals of neural networks: architectures, algorithms, and applications / V. Laurene Fausett. – Englewood Cliffs, NJ : Prentice-Hall, 1994. – 461 p.

5. Гудфеллоу Я. Глубокое обучение / Я. Гудфеллоу, И. Бенджио, А. Курвилль. – 2-е изд., испр. – М. : ДМК Пресс, 2018. – 652 с.

6. DeepLearning4j [Электронный ресурс]. – Режим доступа : <https://github.com/deeplearning4j/deeplearning4j>

БИОМЕТРИЧЕСКАЯ ИДЕНТИФИКАЦИЯ ПОЛЬЗОВАТЕЛЯ НА ОСНОВЕ РАСПОЗНАВАНИЯ ИЗОБРАЖЕНИЙ РАДУЖНОЙ ОБОЛОЧКИ ГЛАЗА С ИСПОЛЬЗОВАНИЕМ ГЛУБОКИХ НЕЙРОННЫХ СЕТЕЙ

А. И. Милованова

Студент

А. А. Сирота

Профессор

Введение

Во всем мире на настоящий момент активно развиваются методы идентификации и распознавания личности. Решить задачу распознавания личности можно с помощью биометрических систем. В биометрических системах используется совокупность характеристик, основанных на биологических особенностях человеческого тела. Основные преимущества биометрических технологий очевидны: это их эффективность, надежность, безопасность. В отличие от других технологий, биометрическая защита более эффективна. Традиционные методы защиты не исключают возможности потери или кражи информации. Уникальный биометрический идентификатор служит ключом, который невозможно потерять [1].

Важность автоматической обработки и анализа биометрической информации в последнее время заметно повысилась из-за повышения требований к функциональным возможностям автоматических систем безопасности, расположенных в общественных местах. Как и во всех системах распознавания актуальным является пресечение попыток подделки биометрического признака. В связи с началом широкого практического применения этому вопросу уделяется большое внимание. Указанные обстоятельства свидетельствуют об актуальности данной темы.

Данная работа посвящена изучению идентификации человека по такому биометрическому параметру, как радужная оболочка глаза. Рассматриваются методы защиты от подделки данного параметра. Целью работы является разработка и исследование программного средства, позволяющего распознавать человека по радужной оболочке.

1. Методика эксперимента

Для распознавания человека по радужной оболочке была построена биометрическая система с использованием сверточной нейронной сети [2]. В качестве языка программирования был выбран Python. Он хорошо подходит для проведения экспериментов и определения гиперпараметров обучения, необходимых для решения данной задачи. Система работает в двух режимах: обучения и классификации. В режиме обучения можно задавать гиперпараметры обучения, а также предоставляется возможность применить фильтры предобработки (фильтр Габора, фильтр Собеля). Обучающая выборка содержит в себе 300 изображений. Часть из них - фотографии радужной оболочки определенного пользователя (Milovanova), другая часть – фотографии радужной оболочки людей из Iris Database (Other). Изображения предварительно нормализуются. Для изображений была реализована аугментация – дополнение выборки искусственными примерами, созданными из фактически присутствующих в самой выборке образцов [3].

Была создана нейронная сеть, схема которой представлена на рис. 1.

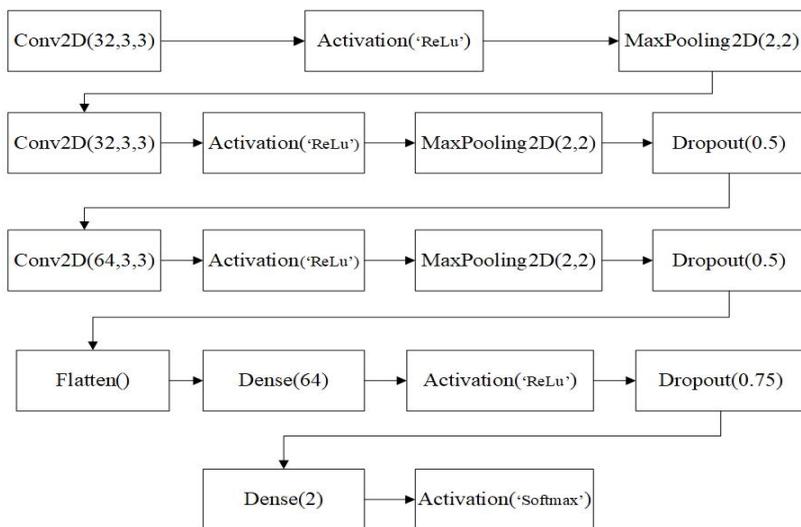


Рис. 1. Схема сверточной нейронной сети

Сеть включает 3 каскада слоев свертки и подвыборки. Размер сверточных ядер 3×3 , размер подвыборки 2×2 . Сверточная часть сети предназначена для выделения характерных признаков в изображении.

После сверточной части идет полносвязная часть нейронной сети, которая отвечает за классификацию. Применяется слой Dropout, который используется для уменьшения переобучения. Выходной полносвязный слой включает всего два нейрона для классификации изображений по двум классам: Milovanova и Other.

В ходе работы был реализован перенос обучения. Идея переноса состоит в следующем. Сверточная часть сети во время обучения учится выделять характерные признаки на изображениях. Если признаки получились достаточно общими, то можно применить их для решения другой задачи классификации.

Для реализации переноса обучения был заменен классификатор в предварительно обученной нейронной сети VGG16. Новая сеть включает сверточную часть VGG16 и новый классификатор для распознавания двух заявленных классов. При этом обучались только веса добавленного в сеть классификатора, предварительно обученная сверточная часть была «заморожена».

В режиме классификации производилось присвоение изображений радужной оболочки глаза к определенному классу, а также построена гистограмма яркости, с целью определения подделки. Была смоделирована ситуация простейшей подделки глаза, а именно, печать его цифровой фотографии на принтере с высоким разрешением. Если глаз напечатан в натуральную величину, то его изображение достаточно похоже на изображение непосредственно регистрируемого «живого» глаза [4].

2. Результаты эксперимента

В ходе эксперимента проводилось обучение и тестирование сетей с различной архитектурой.

В приведенной ниже таблице показаны результаты экспериментов при разном количестве эпох обучения нейронной сети. Обучение и тестирование проводилось на данных без аугментации. В качестве анализируемого показателя используются точность распознавания, а также ошибки первого и второго рода. Можно заметить, что при увеличении числа эпох точность распознавания увеличивается, что приводит к уменьшению значения ошибок.

Результаты экспериментов для данных без аугментации

	Точность распознавания на тестовых данных (val_acc)	Ошибка первого рода (FAR)	Ошибка второго рода (FRR)
25	0.6	0.0	0.02
75	0.78	0.03	0.0
100	0.81	0.02	0.0

Для улучшения анализируемых показателей распознавания была произведена аугментация данных и перенос обучения. Пример аугментированных данных представлен на рис. 2.

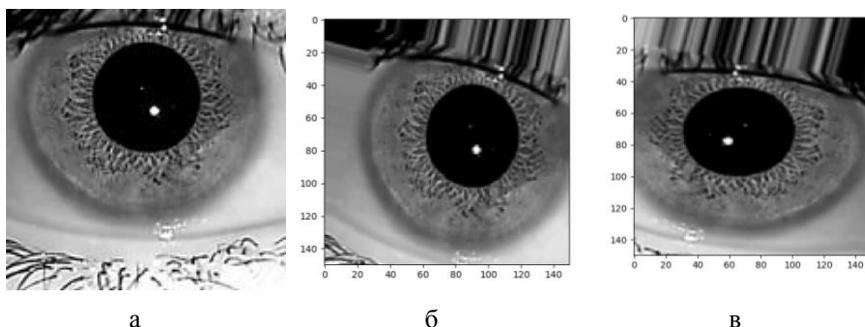


Рис. 2. Пример аугментации данных: *а* – исходное изображение, *б* – поворот, *в* – смещение и увеличение

Точность распознавания увеличилась до 100%, как можно увидеть на рис. 3, а ошибки первого и второго рода составили 0%. При этом функция потерь после 100 эпох обучения значительно уменьшилась, как показано на рис. 4.

В ходе проверки истинности изображения анализ гистограмм показал, что на поддельном изображении большое количество пикселей с высокой яркостью (из-за сетки печати) и гистограмма смещена в сторону высоких значений, что можно увидеть на рис. 5.

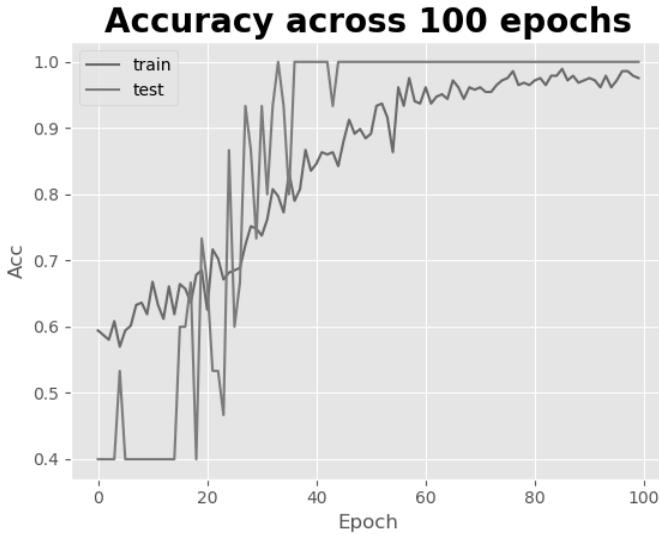


Рис. 3. Точность распознавания после 100 эпох обучения для режимов обучения и тестирования после аугментации данных и переноса обучения

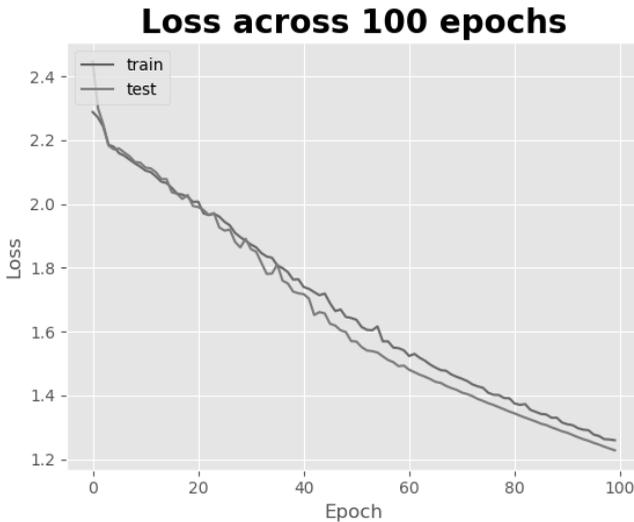


Рис. 4. Функция потерь после 100 эпох обучения для режимов обучения и тестирования после аугментации данных и переноса обучения



серым – гистограмма поддельного, черным – истинного изображения

Рис. 5. Гистограммы изображений

Заключение

В рамках данной работы была создана система распознавания пользователя по изображению радужной оболочки глаза с использованием сверточной нейронной сети. Был проведен анализ достоверности биометрической идентификации пользователя для системы. По результатам выполненной работы можно сделать следующие выводы:

- важную роль при создании обучающей выборки имеет ее объем и предобработка. Это в свою очередь влияет на точность распознавания пользователя нейронной сетью. С данной проблемой позволяет справиться техника аугментации, а также применение различных фильтров на этапе предобработки изображений.

- метод переноса обучения позволяет применить уже обученную сеть для новых задач классификации. При этом она улучшает точность распознавания до 100%, а также уменьшает значение функции потерь.

- поддельное изображение радужной оболочки глаза можно определить с помощью анализа гистограмм изображений.

Список литературы

1. Болл Руд М. Руководство по биометрии / М. Болл Руд, Х. Конел Джонатан. – М. : Техносфера, 2007. – 368 с.
2. Гонсалес Р. Цифровая обработка изображений / Р. Гонсалес, Р. Вудс. – М. : Техносфера, 2005. – 1072 с.
3. Николенко С. Глубокое обучение / С. Николенко, А. Кадури, Е. Архангельская. – СПб. : Питер, 2018. – 480 с.
4. Матвеев И. А. Методы и алгоритмы автоматической обработки изображений радужной оболочки глаза: автореферат дис. ... доктора технических наук : 05.13.11 / Матвеев Иван Алексеевич. – М., 2014. – 41 с.

МОДЕЛИРОВАНИЕ ЭКВИДИСТАНТНОЙ ТРЕХЭЛЕМЕНТНОЙ РЕШЕТКИ В УСЛОВИЯХ ВЛИЯНИЯ ПОДСТИЛАЮЩЕЙ ПОВЕРХНОСТИ

Е. Ю. Морозов

Студент

Л. А. Минин

Доцент

Введение

При моделировании реальных задач важную роль играет возможность аналитического решения проблемы в целом или какой-то ее существенной части. Это позволяет сравнительно просто оценить влияние входных параметров на решение, разрешающую способность предлагаемого алгоритма и границы его применимости. В данной статье приводится пример такой модели, развивающей предложенный в [1] метод пеленгации электромагнитного сигнала по разности фаз на решетке из трех датчиков.

1. Постановка задачи

Рассматривается трехэлементная вертикальная система датчиков с расстоянием Δh между соседними элементами. В данной работе предполагается, что принимаемый сигнал формируется как суперпозиция двух плоских электромагнитных волн одинаковой частоты. Первая из них, приходящая от регистрируемого источника, имеет амплитуду $|E_0|$ и фазу на среднем элементе φ_0 . Вторая представляет собой отраженную от поверхности Земли волну с амплитудой $|E_1|$ и фазой на среднем элементе φ_1 . Соответствующие комплексные компоненты задаются формулами

$$E_0 = |E_0|e^{i\varphi_0}, \quad E_1 = |E_1|e^{i\varphi_1}.$$

Пусть β_0 – угол отклонения волны источника от горизонта (считаем, что источник находится выше датчиков и на большом расстоянии), β_1 – угол отклонения отраженной волны от горизонта (отражение происходит ниже датчиков), λ – длина волны приходящего сигнала. Введем следующие обозначения:

$$\omega_0 = \frac{2\pi}{\lambda} \Delta h \sin \beta_0, \quad \omega_1 = \frac{2\pi}{\lambda} \Delta h \sin \beta_1. \quad (1)$$

Тогда для комплексных сигналов z_0 , z_B и z_H , регистрируемых на среднем, верхнем и нижнем датчиках соответственно, справедливы следующие соотношения:

$$\begin{cases} z_0 = E_0 + E_1 \\ z_H = E_0 e^{-i\omega_0} + E_1 e^{i\omega_1} \\ z_B = E_0 e^{i\omega_0} + E_1 e^{-i\omega_1} \end{cases} \quad (2)$$

Требуется по заданным величинам z_0 , z_B и z_H определить угол β_0 (рис. 1).

Система (2) содержит три комплексных уравнения относительно шести неизвестных вещественных величин β_0 , β_1 , $|E_0|$, $|E_1|$, φ_0 , φ_1 и является нелинейной. Выбор данной модели обусловлен тем, что удастся аналитически исключить из этой системы все переменные, кроме ω_0 , которая, согласно (1), включает в себя интересующую нас величину β_0 .

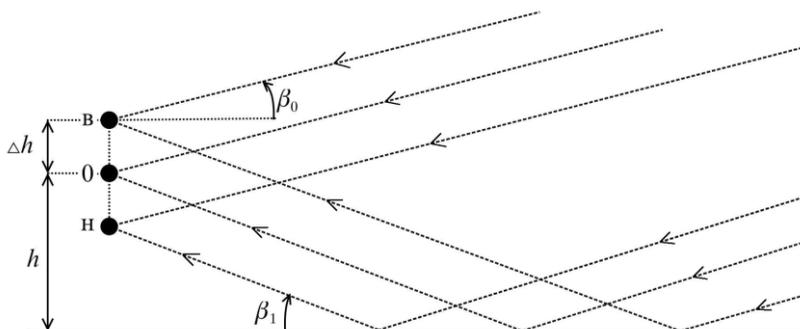


Рис. 1. Геометрия расчётной модели эквидистантной трехэлементной антенной решеткой из соосных вертикальных вибраторных антенн

2. Решение системы

Для исключения E_0 умножим второе уравнение (2) на $e^{i\omega_0}$, а третье уравнение – на $e^{-i\omega_0}$. Тогда:

$$\begin{cases} z_0 = E_0 + E_1 \\ z_H e^{i\omega_0} = E_0 + E_1 e^{i(\omega_0+\omega_1)} \\ z_B e^{-i\omega_0} = E_0 + E_1 e^{-i(\omega_0+\omega_1)} \end{cases}$$

Вычитая первое уравнение полученной системы из второго и третьего уравнений, приходим к системе двух комплексных уравнений

$$\begin{cases} z_H e^{i\omega_0} - z_0 = E_1(e^{i(\omega_0+\omega_1)} - 1), \\ z_H e^{-i\omega_0} - z_0 = E_1(e^{-i(\omega_0+\omega_1)} - 1). \end{cases}$$

Для исключения E_1 , умножим второе уравнение на $e^{i(\omega_0+\omega_1)}$ и сложим с первым:

$$z_H e^{i\omega_0} - z_0 + (z_B e^{-i\omega_0} - z_0)e^{i(\omega_0+\omega_1)} = 0.$$

Таким образом,

$$z_H e^{i\omega_0} - z_0 = -(z_B e^{-i\omega_0} - z_0)e^{i(\omega_0+\omega_1)}.$$

Осталось исключить ω_1 , для чего достаточно взять модуль от левой и правой частей последнего равенства:

$$|z_H e^{i\omega_0} - z_0| = |z_B e^{-i\omega_0} - z_0|. \quad (3)$$

Аналогичное соотношение получается и для ω_1 .

$$|z_H e^{-i\omega_1} - z_0| = |z_B e^{i\omega_1} - z_0|. \quad (4)$$

3. Анализ разрешимости уравнения относительно ω_0

Запишем уравнение (3) в следующем виде:

$$|z_H e^{i\omega_0} - z_0|^2 = |z_B e^{-i\omega_0} - z_0|^2.$$

Воспользовавшись тем, что $|z|^2 = z z^*$, где z^* означает комплексное сопряжение, приходим к равенству

$$(z_H e^{i\omega_0} - z_0)(z_H e^{i\omega_0} - z_0)^* = (z_B e^{-i\omega_0} - z_0)(z_B e^{-i\omega_0} - z_0)^*.$$

Раскрывая скобки, получим

$$|z_H|^2 - z_H z_0^* e^{i\omega_0} - z_H^* z_0 e^{-i\omega_0} + |z_0|^2 = |z_B|^2 - z_B z_0^* e^{-i\omega_0} - z_B^* z_0 e^{i\omega_0} + |z_0|^2.$$

Перегруппируем слагаемые:

$$(z_B^* z_0 - z_H z_0^*) e^{i\omega_0} + (z_B z_0^* - z_H^* z_0) e^{-i\omega_0} = |z_B|^2 - |z_H|^2.$$

Обозначим $z_B^* z_0 - z_H z_0^* = u - iv$. Тогда $z_B z_0^* - z_H^* z_0 = u + iv$,

$$(u + iv)(\cos \omega_0 + i \sin \omega_0) + (u + iv)(\cos \omega_0 - i \sin \omega_0) = |z_B|^2 - |z_H|^2.$$

Соответственно,

$$u \cos \omega_0 + v \sin \omega_0 = \frac{|z_B|^2 - |z_H|^2}{2} = c. \quad (5)$$

Условие разрешимости уравнения (5) может быть записано в следующем виде [2]:

$$u^2 + v^2 \geq c^2. \quad (6)$$

Для решения (5), перейдем к половинному углу

$$u \left(\cos^2 \frac{\omega_0}{2} - \sin^2 \frac{\omega_0}{2} \right) + 2v \sin \frac{\omega_0}{2} \cos \frac{\omega_0}{2} = c \left(\cos^2 \frac{\omega_0}{2} + \sin^2 \frac{\omega_0}{2} \right)$$

и перегруппируем слагаемые

$$(u - c) \cos^2 \frac{\omega_0}{2} + 2v \sin \frac{\omega_0}{2} \cos \frac{\omega_0}{2} - (u + c) \sin^2 \frac{\omega_0}{2} = 0.$$

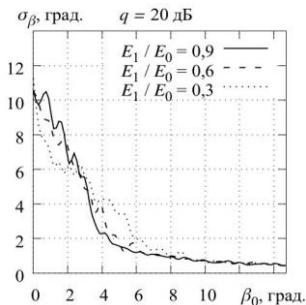
Если $|u - c| > |u + c|$, то делим данное уравнение на $\sin^2 \omega_0/2$, в противном случае делим на $\cos^2 \omega_0/2$. Получается квадратное уравнение относительно $\operatorname{ctg} \omega_0/2$ в первом случае или $\operatorname{tg} \omega_0/2$ во втором случае с неотрицательным в силу (6) дискриминантом.

4. Статистический анализ погрешности

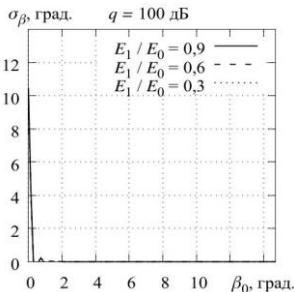
Наличие явного метода решения системы (2) существенно облегчает анализ погрешности. Стандартным образом возмущаются входные данные z_0 , z_n и z_b с заданным отношением сигнал/шум (ОСШ) q , решается полученная система и вычисляется среднеквадратичная ошибка (СКО) σ_β определения угла β_0 .

Возмущения исходных значений сигналов на каждой из антенн производилось путем добавления к ним некоторой случайной величины, распределенной по нормальному закону с нулевым математическим ожиданием и дисперсией, соответствующей заданному ОСШ. Устойчивость предложенного метода определения угла β_0 оценивалась при значении ОСШ $q = 20$ дБ. Также при оценке рассматривался идеальный случай, в котором влияние шума пренебрежимо мало, т.е. ОСШ $q = 100$ дБ. Решение уравнения (5) производилось численно, с использованием метода простой итерации. На рис. 2 и 3 приведены графики зависимостей СКО от определяемого угла.

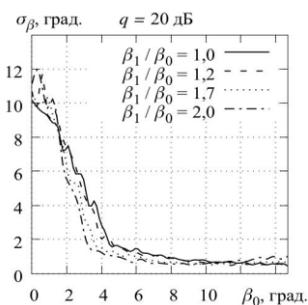
Из этих графиков можно сделать следующий вывод: если β_0 имеет величину порядка $1^\circ - 3^\circ$, то погрешность становится больше самой определяемой величины, т.е. вычисления с практической точки зрения бесполезны. Существенным оказывается и соотношение углов β_0 и β_1 . С увеличением разности между ними растет и погрешность. Большая разность между этими углами возможна в том случае, когда объект пеленгации находится слишком близко к датчикам.



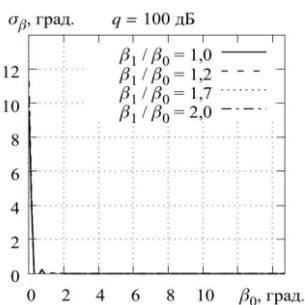
a)



b)

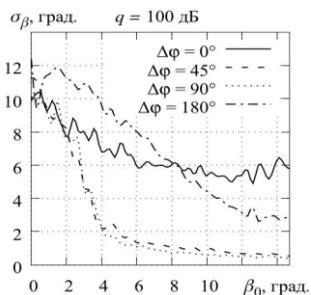


v)

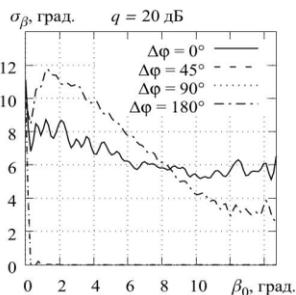


z)

Рис. 2. Графики частотных зависимостей случайных СКО определения угла при различных значениях параметров принятого сигнала:
a, б – амплитуды, в, з – углы приема отраженной волны



a)



b)

Рис. 3. Графики частотных зависимостей случайных СКО определения угла при различных значениях разности фаз принятых сигналов:
a – q = 100 дБ, б – q = 20 дБ

Достаточно важным параметром является разность фаз – если пришедшие сигналы приходят в фазе или противофазе, то предложенная модель оказывается неэффективной. Это объясняется тем, что величины z_n и z_b становятся слишком близкими друг к другу и критерий разрешимости (6) нарушается для большинства значений β_0 . Сложности определения угла β_0 , как и следовало ожидать, начинаются при стремлении его к нулю. В этом случае все три коэффициента u , v и s уравнения (5) становятся малыми, и преобразования, связанные с делением на эти величины, являются неустойчивыми.

Заключение

Отметим, что помимо предложенной модели, можно построить и другие, как, например, в [1]. Однако система уравнений, описывающих ту модель, содержала не шесть, а пять вещественных параметров и являлась переопределенной, что отражалось в наличии областей изменения параметров, при которых система становилась несовместной. Предложенная модель лишена данного недостатка. Вопрос о распространении ее на большее число датчиков или на другую геометрию их взаимного расположения требует дополнительного исследования.

Список литературы

1. Способ и характеристики угломестного радиопеленгования с использованием эквидистантной трехэлементной антенной решетки из соосных вертикальных вибраторных антенн / А.Д. Виноградов, В.В. Грибанов [и др.] // Антенны. – Вып. 5 (237). – 2017. – С. 13-18.
2. Справочник по специальным функциям с формулами, графиками и математическими таблицами / под ред. М. Абрамовица и И. Стиган; науч. ред. Н. П. Беляцкий. – М. : Наука, 1979. – 207 с.

ПРИМЕНЕНИЕ ГИБРИДНЫХ И НЕЙРОСЕТЕВЫХ МОДЕЛЕЙ ДЛЯ ПОСТРОЕНИЯ РЕКОМЕНДАТЕЛЬНЫХ СИСТЕМ

Д. Э. Мубаракшин

Студент

А. А. Сирота

Профессор

Введение

Исторически развитие рекомендательных систем шло по двум основным направлениям. В хронологическом порядке, вначале появилась так называемая *коллаборативная фильтрация*. Со временем это направление достигло пика своего развития, когда для дальнейшего увеличения точности рекомендаций требовались огромные денежные и временные затраты.

Вследствие этого начали активно развиваться и другие подходы к решению задачи. Многие компании осознали, что качественная и подробная информация о рекомендуемых объектах может быть большим подспорьем при учете интересов пользователей для формирования рекомендаций. Так появилась *фильтрация на основе содержания*. Однако оказалось, что и этот подход имеет ряд проблем, решить которые простым путем так и не удалось.

Таким образом, как ни странно, ни один рекомендательный алгоритм не мог удовлетворить потребностям пользователя. Тогда появились алгоритмы *гибридной* фильтрации [1], которые сочетали в себе преимущества упомянутых выше алгоритмов.

В рамках данной статьи будет предложен новый подход к решению задачи, основой которого будет именно гибридная фильтрация, и будет продемонстрирована эффективность этого подхода при применении к реальным данным.

1. Постановка задачи

Имеются входные данные, предоставленные одним из крупнейших онлайн-сервисов по продаже книг. Они содержат чуть меньше миллиона записей, в которых указаны id пользователя, id книги и оценка, которую этот пользователь поставил книге по шкале от 1 до 10. Также имеется дополнительная информация о книгах: помимо автора и названия, это жанр произведения, количество страниц, цена книги и т.д.

Цель работы – для каждого пользователя сформировать персонализированный ранжированный список рекомендуемых книг фиксированной длины.

Зачем это может быть полезно? Рассмотрим два примера. Если новая книга только что появилась в системе, при этом ни один пользователь еще не успел ее оценить, но при добавлении книги в систему были указаны все ее дополнительные параметры, мы сразу сможем рекомендовать эту книгу пользователям за счет этих дополнительных данных. Другой пример: если пользователь регулярно покупал одну-две книги в месяц, но в какой-то момент не совершал покупок уже три месяца, мы можем подобрать оптимальный набор книг и предоставить пользователю скидку на эти книги. Вероятность того, что он приобретет одну из этих книг и будет совершать покупки и в дальнейшем, довольно высока.

2. Формальное описание гибридной модели

Пусть U — это множество пользователей, зарегистрированных в системе, I — множество книг, которые пользователи могут оценивать, F^U — множество характеристик пользователей и F^I — множество характеристик книг. Каждый пользователь может оценивать некоторое количество книг, при этом оценка книги пользователем может быть как позитивной (например, 10 по десятибалльной шкале), так и негативной (1 по десятибалльной шкале). Множество всех оценок пользователями книг является набором пар $(u, i) \in U \times I$. Это множество является объединением множеств с только позитивными оценками S^+ и только негативными оценками S^- .

Пользователи и книги полностью описываются своими признаками [2]. Каждый пользователь u характеризуется своим множеством признаков $f_u \subset F^U$. То же самое применимо и к любой книге i : $f_i \subset F^I$. Все признаки известны заранее и используются в качестве метаданных пользователей и книг.

В модели для каждого признака f рассчитываются представления e_f^U и e_f^I , представляющие из себя векторы длиной d . Кроме того, каждому признаку соответствует скалярная величина смещения (b_f^U для признаков пользователей и b_f^I для признаков книг).

Скрытое представление пользователя u представляется в виде суммы представлений его признаков:

$$q_u = \sum_{j \in f_u} e_j^U . \quad (1)$$

То же самое справедливо и для книги i :

$$p_i = \sum_{j \in f_i} e_j^I . \quad (2)$$

Величина смещения для пользователя u рассчитывается как сумма смещений его признаков [3]:

$$b_u = \sum_{j \in f_u} b_j^U. \quad (3)$$

Соответственно, для книги i :

$$b_i = \sum_{j \in f_i} b_j^I. \quad (4)$$

Тогда предсказание модели относительно оценки объекта i пользователем u рассчитывается как скалярное произведение представлений пользователя и объекта с учетом смещений пользователя и объекта [4]. Обычно при этих расчетах также добавляют функцию активации f :

$$r_{ui} = f(q_u \cdot p_i + b_u + b_i). \quad (5)$$

В качестве функции активации f могут применяться различные функции. Тожественное отображение неплохо подходит для предсказания оценок, но так как нас в первую очередь интересует то, какие объекты лучше подходят для конкретного пользователя, разумнее всего использовать сигмоиду:

$$f(x) = \frac{1}{1 + \exp(-x)}. \quad (6)$$

Целью оптимизации для модели будет являться минимизация функции потерь для всего объема данных. Функцию потерь можно представить как

$$L(e^U, e^I, b^U, b^I) = - \prod_{(u,i) \in S^+} r \times \prod_{(u,i) \in S^-} (1 - r_{ui}) \quad (7)$$

Для обучения модели используются алгоритмы асинхронного стохастического градиентного спуска. В частности, был использован алгоритм AdaGrad, основная суть которого заключается в том, что скорость обучения не является постоянной, а регулируется самим алгоритмом в зависимости от параметров.

3. Архитектура нейронной сети

Несмотря на все достоинства алгоритма гибридной фильтрации, даже при длительном обучении соответствующей модели результаты получаются несовершенными. В связи с этим была выдвинута гипотеза: нужно решить задачу альтернативными методами, а затем составить ансамбль двух моделей и сравнить результаты новой модели с предыдущими показателями. В качестве такого альтернативного метода были выбраны нейронные сети.

На рисунке представлена финальная архитектура нейронной сети, которая была использована в качестве одной из моделей ансамбля:

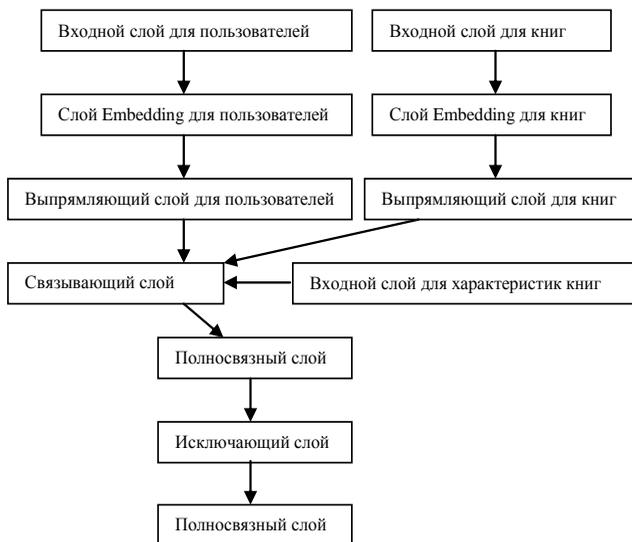


Рисунок. Архитектура нейронной сети

4. Сравнение с другими моделями и результаты экспериментов

ROC AUC и MAE – метрики, широко используемые при анализе рекомендательных систем. Метрика ROC AUC может принимать значения от 0 до 1, где 1 – идеальный результат. MAE (Mean Absolute Error) – это средняя абсолютная ошибка, чем она меньше – тем лучше результат. Результаты работы представленного ансамбля моделей и других широко распространенных методов представлены в табл. 1.

Таблица 1

Сравнение моделей по метрикам ROC AUC и MAE

	ROC AUC	MAE
Коллаборативная фильтрация	0,52	0,49
Фильтрация на основе содержания	0,63	0,41
Гибридная фильтрация	0,67	0,34
Нейронная сеть	0,5	0,52
Ансамбль моделей	0,69	0,29

Пример результатов работы рекомендательной системы представлен в следующей табл. 2.

Таблица 2

Пример рекомендаций для одного из пользователей

Книги, которые пользователь оценил положительно			
Оценка	Автор	Название	Жанр
9	Х.К. Андерсен	Стойкий оловянный солдатик	Сказки зарубежных писателей
10	Р. Скарри	Книжка про машинки	Сказки зарубежных писателей
10	Л. Толстой	О людях больших и маленьких	Повести и рассказы о детях
10	А. Грин	Рассказы	Произведения школьной программы
Рекомендованные системой книги			
9,4	Х.К. Андерсен	Снежная королева	Сказки зарубежных писателей
9,32	А.С. Пушкин	У Лукоморья дуб зеленый	Отечественная поэзия для детей
9,32	Ф.Х. Бернетт	Таинственный сад	Повести и рассказы о детях
8,76	Д.П. Льюис	Старый дом	Зарубежная поэзия для детей

Как можно увидеть, с одной стороны, рекомендации, выдаваемые системой, довольно предсказуемые (например, появляются рекомендации того автора, который уже был оценен положительно – Х.К. Андерсен). Но с другой стороны, появляются новые авторы и даже жанры, которые были выявлены на основе информации об оценках других пользователей и характеристиках книг (например, в рекомендациях появилась поэзия).

Заключение

В рамках данной статьи был предложен новый подход к решению задачи построения рекомендательной системы, основой которого стал алгоритм гибридной фильтрации. Было представлено формальное описание модели, проведено сравнение с другими, широко распространенными моделями, и представлены результаты работы модели.

Список литературы

1. arXiv.org [Электронный ресурс] : M. Kula. Metadata Embeddings for User and Item Cold-start recommendations. – Режим доступа : <https://arxiv.org/pdf/1507.08439.pdf>

2. CiteSeer [Электронный ресурс] : I. Soboroff, C. Nicholas. Combining content and collaboration in text filtering. – Режим доступа : <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.39.8019>
3. MIT Media Lab [Электронный ресурс] : M. Saveski, A. Mantrach. Item cold-start recommendations. – Режим доступа : http://web.media.mit.edu/~msaveski/assets/publications/2014_item_cold_start/paper.pdf
4. The Stanford University InfoLab [Электронный ресурс] : J. McAuley, J. Leskovec. Hidden factors and hidden topics: understanding rating dimensions with review text. – Режим доступа : <http://infolab.stanford.edu/~julian/pdfs/recsys13.pdf>

РАЗРАБОТКА ТРАНСЛЯТОРА КОДА С ЯЗЫКА МАТЛАВ В КОД НА ЯЗЫКЕ PYTHON

В. А. Новиков

Студент

Д. И. Соломатин

Старший преподаватель

Введение

В настоящее время MATLAB является популярной платформой научных вычислений. При всех его многочисленных преимуществах есть существенный недостаток: высокая стоимость лицензии данного пакета программ. Ряд пользователей ограничен в возможности использования кода на языке MATLAB, поскольку не может себе позволить приобрести лицензионную копию данного ПО. В том числе по этой причине, ряд исследователей и разработчиков ищут подходящую замену MATLAB в своей работе. И на эту роль во многих случаях подходит язык программирования Python с богатой коллекцией библиотек для научных вычислений, обработки и визуализации данных и т.д. Однако, при переходе на Python возникает проблема использования уже существующих наработок на языке MATLAB. Одним из способов решения указанной проблемы является автоматическая трансляция MATLAB-кода в код на языке Python.

1. Постановка задачи

Необходимо разработать набор приложений и библиотек, который позволит осуществлять трансляцию кода, написанного на языке MATLAB, в код на языке Python.

К полученному коду предъявляются следующие требования:

- сохранение функциональных возможностей, исходной семантики операторов и операций, корректность получаемых результатов;
- сохранение исходной структуры кода.

Полученный после трансляции код должен осуществлять полнофункциональную поддержку таких возможностей языка MATLAB, как:

- типы данных: скалярные типы данных, массивы, строки, с учетом особенностей их реализации в MATLAB [1];
- базовые математические векторные и поэлементные операции;
- поддержка процедур и функций с учетом их особенностей;
- поддержка основной функциональности стандартной библиотеки.

Трансляция кода должна осуществляться в формат, совместимый с актуальной версией Python [2].

Таким образом, можно выделить две основные программные части, которые требуется реализовать:

- библиотека совместимости, написанная на языке Python, которая предоставит отсутствующую в Python функциональность;
- собственно транслятор, который будет осуществлять преобразование кода на языке MATLAB в код на языке Python с применением указанной библиотеки совместимости.

2. Сравнение функциональных возможностей языков

В функциональности языков имеются существенные различия: в поведении базовых типов данных, в семантике определения и вызова процедур и функций, в наборе поддерживаемых операций. Для реализации транслятора необходимо проанализировать возможности двух языков. В данном разделе выделяются те особенности, которые должны быть учтены для корректной трансляции кода, написанного на языке MATLAB.

2.1. Матрицы

Под реализацией матриц в Python будем понимать функциональность, предоставляемую библиотекой `numpy` [3].

В реализациях матриц MATLAB и `numpy` имеется ряд основных различий:

- семантика операции индексации;
- возможность динамического расширения.

В языках MATLAB и Python имеется различие в порядке следования индексов при многомерной индексации. На примере N-мерной матрицы: в языке MATLAB индексы следует располагать в следующем порядке:

`(rowIndex, columnIndex, 3thIndex, ..., N-1thIndex, NthIndex),`

тогда как в `numpy` принят следующий порядок:

(N_{th}Index, N-1_{th}Index, ..., 3_{th}Index, rowIndex, columnIndex)

В языке MATLAB синтаксис индексации совпадает с синтаксисом вызова функций: индексы записываются в круглых скобках, тогда как в Python – в квадратных. Важно реализовать в Python возможность индексации с помощью метода `__call__`, это позволит поддерживать анонимные функции и дескрипторы функций.

Данные языки имеют разные начальные индексы. Индексация в MATLAB начинается с единицы, индексация в Python – с нуля.

MATLAB предлагает следующие способы индексации массивов:

Линейная индексация. Независимо от размерности массива, имеется возможность обращаться к его элементам по скалярному индексу. Также имеется возможность линейной индексации массивом скаляров: при этом результатом будет массив, состоящий из элементов с линейными индексами, находящимися в индексирующем массиве. Наглядно особенности линейной индексации представлены на рис. 1. В numpy также имеется возможность линейной индексации. Однако по умолчанию она осуществляется в порядке расположения элементов по строкам.

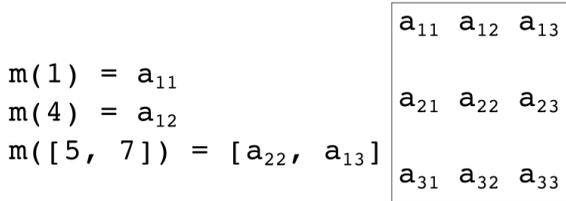


Рис. 3. Линейная индексация в MATLAB

Многомерная индексация. В MATLAB имеется возможность обращаться к элементу по его многомерному индексу. Каждый индекс может быть как скаляром, так и массивом скаляров. Результат подобной операции наглядно представлен на рис. 2, где объекты результирующего массива находятся на пересечении линий.

В numpy также имеется возможность многомерной индексации, однако, у этой операции отличная от MATLAB семантика. Наглядно это отличие можно продемонстрировать на примере:

`m[[0, 1], [2, 3]] = [m01, m23]`

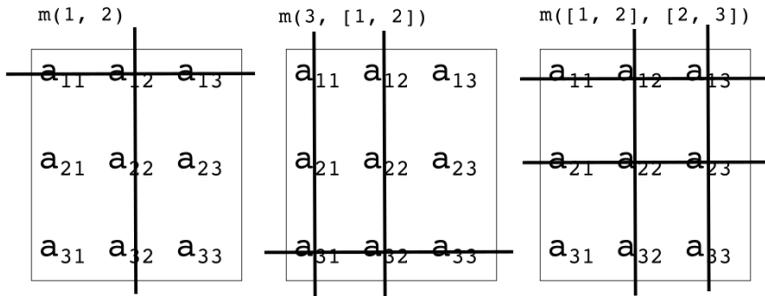


Рис. 4. Многомерная индексация в MATLAB

Для разрешения описанных выше различий требуется определить некоторый класс-обертку над массивом `numpy`, в котором:

- переопределены методы `__getitem__`, `__setitem__` для соответствия описанным выше возможностям индексации;
- переопределен метод `__call__` для доступа к элементам по индексу;
- поддерживается возможность динамического роста.

2.2. Операции

В языке MATLAB операции разделяются на поэлементные и матричные. В `numpy` все бинарные операции – поэлементные, а матричные представлены в виде функций.

В целом, любая операция MATLAB может быть записана и в Python – либо в инфиксной нотации, либо вызовом функции.

Особое внимание следует уделить операциям умножения и деления. В MATLAB значки `*` и `/` обозначают матричное умножение и деление соответственно, однако часто используются и для скалярных величин. В Python данные операции являются поэлементными, а матричные представлены в виде функций. Поскольку в MATLAB данные операции встречаются повсеместно, замена их на вызовы функций сильно повлияет на читаемость транслируемого кода. Поэтому в Python необходимо переопределить данные операции для соответствия их MATLAB семантике.

Таким образом, можно предложить следующий способ разрешения описанных выше различий в функциональности:

- для класса массива переопределить операции `*` и `/`;
- для операций, не представленных в Python в инфиксной нотации, реализовать функции, выполняющие соответствующие действия.

Результат трансляции небольшого математического выражения представлен в табл. 1.

Результат трансляции математического выражения

MATLAB	Python
<code>a = 5 * b' + c .* d</code>	<code>a = 5 * ctranspose(b) + dotmul(c, d)</code>

2.3. Операторы

Язык программирования MATLAB предлагает следующие операторы:

- операторы ветвления: IF, SWITCH;
- операторы цикла: FOR, WHILE, поддерживаются операторы управления циклами BREAK и CONTINUE;
- оператор присваивания

Поддержка такого набора операторов соответствует большинству из языков высокого уровня, в том числе и Python. Различие в MATLAB и Python версиях данных операторов заключается лишь в синтаксисе.

Оператор присваивания в MATLAB отличается от такового в Python: в MATLAB присваиваемое значение копируется, когда как в Python имя переменной связывается с присваиваемым значением, и изменение этого значения отразится на всех, ссылающихся на него, переменных.

2.4. Функции

В MATLAB при определении функции задается список имен входных и выходных переменных. Возможно определение функций, принимающих произвольное число параметров (при помощи ключевого слова `varargin`), и возвращающих произвольное число значений (при помощи ключевого слова `varargout`).

В MATLAB явный вызов оператора возврата из функции `return` не требуется. Значения, присвоенные выходным переменным, возвращаются из функции при достижении конца тела функции. Тем не менее, с помощью оператора `return` можно принудительно вернуть управление вызывающему коду в произвольном месте тела функции. Python требует явного вызова оператора возврата из функции со списком возвращаемых значений.

В MATLAB внутри тела функции определены следующие специальные функции:

- `nargin` – функция, возвращающая число переданных в функцию аргументов;
- `nargout` – функция, возвращающая число ожидаемых для возврата значений функции. В Python нет удобного синтаксиса для получения данных значений.

В MATLAB, если функция не принимает аргументов, вызов возможен без добавления круглых скобок. В Python вызов функции всегда выполняется с добавлением круглых скобок.

Особенностью вызова функций в MATLAB является то, что возможна передача любого количества аргументов, независимо от количества входных переменных (но при отсутствии varargin «лишние» параметры будут потеряны). В Python число переданных аргументов должно соответствовать числу входных переменных (если она не определена как функция со списком аргументов переменной длины).

В MATLAB возможно множественное присваивание значений, возвращенных из функции. При этом, если число переменных для присваивания меньше числа возвращаемых значений, то «лишние» значения теряются. В Python число возвращаемых значений должно точно соответствовать числу переменных.

Все описанные выше различия могут быть решены следующим образом:

- явно определить в сигнатуре транслированных функций аргументы nargin, nargsout;
- явно добавить в конец тела функции оператор возврата со списком всех выходных переменных;
- явно передавать в вызов функции число ожидаемых для возврата значений nargsout;
- реализовать декоратор @mlfunc, применяемый к каждой транслированной функции. В задачи данного декоратора входит подсчет реального числа переданных аргументов (nargin), передача аргументов функции корректным путем (выбрасывая лишние значения или дополняя недостающие значениями None).

Результат трансляции небольшой функции представлен в табл. 2:

Таблица 2

Результат трансляции функции

MATLAB	Python
<pre>function[a,b,c]=func(d, e) a = d * e(5) + e'; b = d ^ 3 + e .* 7; c = nargin; endfunction; [a, b, c] = func(d, e)</pre>	<pre>@mlfunc def func(d, e, nargin, nargsout=1): a = d * e(5) + ctranspose(e); b = d ** 3 + dotmul(e, 7); c = nargin; return (a, b, c); (a, b, c,) = func(d, e, nargsout=3)</pre>

3. Реализация

Исходя из анализа функциональных возможностей языков, выделены следующие задачи, которые решаются в рамках библиотеки совместимости:

- класс MATLAB-совместимого массива, расширяющий функциональность numpy-массивов. В частности: переопределение операций для соответствия семантики их MATLAB-версиям, реализация

возможности динамического расширения, переопределение операции индексации для поддержки всех возможных в MATLAB видов.

- набор функций для поддержки поэлементных операций;
- набор функций-заменителей стандартной библиотеки MATLAB;
- набор вспомогательных функций для замены стандартных синтаксических конструкций языка MATLAB. Пример: конструкции создания массивов и диапазонов.

Транслятор кода представлен как совокупность следующих модулей:

- модуль лексического и синтаксического анализа. В его задачу входит преобразование исходных текстов на языке MATLAB в абстрактное синтаксическое дерево (далее АСД). Данный модуль поддерживает основные синтаксические конструкции языка и формирует АСД, удобное для дальнейшей обработки;

- модуль преобразований АСД, на который возложены следующие обязанности:

- детектирование вызовов функций, выполненных без использования круглых скобок (такие выражения будут распознаны как имя переменной) и замена соответствующих узлов АСД на узлы вызова функций;

- распознавание выражений присваивания по индексу и замена данных узлов на узлы присваивания. Поскольку в MATLAB операции индексации и вызова функций имеют одинаковый синтаксис, то на уровне синтаксического анализа различить два данных случая невозможно. Необходимо по каким-либо критериям отслеживать данные случаи и уже на сформированном АСД заменять узлы на соответствующие.

- модуль кодогенерации, в задачи которого входит обработка АСД и генерация Python-кода;

- приложение транслятора – объединяет воедино все вышеописанные модули, предоставляя пользователю консольный интерфейс для осуществления трансляции кода.

В табл. 3 представлен пример трансляции скрипта, содержащий демонстрацию основных возможностей, поддерживаемых транслятором.

Заключение

В результате проделанной работы удалось реализовать сохраняющий структуру и функциональность исходного кода транслятор кода MATLAB в Python.

Поддерживаются следующие возможности языка MATLAB:

- типы данных: скалярные типы данных, массивы, строки, с учетом особенностей их реализации;

- математические векторные и поэлементные операции;
- операторы; if, while;
- поддержка процедур и функций с учетом описанных выше особенностей;
- поддержка наиболее часто встречающихся функций стандартной библиотеки (генерация массивов, тригонометрические функции, генерация случайных чисел).

Для проверки полученных результатов были написаны тестовые скрипты на языке MATLAB, отражающие вышеописанную функциональность. Полученный после трансляции код оставался работоспособным. Таким образом, в настоящий момент возможна корректная трансляция некоторая части MATLAB-кода.

В качестве направлений для дальнейшего развития можно выделить:

- расширение списка поддерживаемых функций стандартной библиотеки, в особенности функций визуализации данных;
- поддержка структур и массивов ячеек.

В перспективе возможна полноценная трансляция функций, представленных в библиотеке Octave (открытый проект, использующий язык MATLAB), в частности, модулей general и elfun.

Таблица 3

Пример трансляции скрипта

MATLAB	Python
<pre> K = 1000; n = 12; m = 1; s = 3; r = rand(n, K); x = m + s * (sum(r) - 6); ms = zeros(1, K); for k = 1 : K ms(k) = mean(x(1: k)); end; a = [1,2,3] m = [1,2 3;4, 5, 6;7 8 9]; e = ~5^-2*2 d = a + m(1, 1:3) * 5 a(1) = 10; m(1,2) = 11; m(1:2, 2:3) = [77,88;99,11]; </pre>	<pre> from ampt.core import * import ampt.amptlib as mpl K = 1000 n = 12 m = 1 s = 3 r = mpl.rand(n, K) x = m + s * (mpl.sum(r) - 6) ms = mpl.zeros((1,K)) for k in rg(1, K): ms[k] = mpl.mean(x(slice(1, k))) a = mr([[1, 2, 3]]) m = mr([[1, 2, 3], [4, 5, 6], [7, 8, 9]]) e = not 5 ** (-2) * 2 d = a + m(1,slice(1, 3)) * 5 a[1] = 10 m[1,2] = 11 m[slice(1, 2),slice(2, 3)] = mr([[77, 88], [99, 11]]) </pre>

Список литературы

1. Language Fundamentals [Электронный ресурс]. – Режим доступа : <https://www.mathworks.com/help/matlab/language-fundamentals.html>
2. Python 3.5.6 documentation [Электронный ресурс]. – Режим доступа : <https://docs.python.org/3/>
3. NumPy Reference [Электронный ресурс]. – Режим доступа : <https://docs.scipy.org/doc/numpy-1.14.0/reference/index.html>

ЦИФРОВОЙ АККРЕДИТИВ С ИСПОЛЬЗОВАНИЕМ ТЕХНОЛОГИИ БЛОКЧЕЙН

А. В. Обручников

Студент

Д. И. Соломатин

Старший преподаватель

Введение

Блокчейн – технология, стремительно ворвавшаяся в IT мир и изменившая представление о цифровых денежных средствах. Она впервые была применена на практике в криптовалюте «Bitcoin».

Сегодня можно увидеть множество вариантов применения этой технологии: финансы, право владения, интернет вещей, электронное голосование и другие [1]. В данной статье предлагается рассмотреть её возможности в качестве основы для проведения аккредитивной сделки.

Идея применения технологии распределенного реестра для проведения аккредитивов далеко не нова. Примером этому служат ряд осуществленных сделок, таких как:

– Контракт между Альфа-банк и S7 Airlines, в ходе которого специалисты со стороны банка построили систему умных контрактов на базе блокчейна Ethereum. По их словам, эта система повышает автоматизацию документооборота и прозрачность сделки.

– Договор о поставке молочной продукции на \$100,000 между ирландской Ornuа и Seychelles Trading Company. Он был реализован банком Barclays при помощи блокчейн платформы Waves. Главное достигнутое преимущество использования цепочки блоков в этом случае – сокращение времени проведения операции со свойственных 10 дней до 4 часов.

- Решение IBM и его тестовая сделка с Maersk – более глобальное решение, покрывающее всю цепочку поставок, включая государственные структуры.

- А также ряд других сделок: Сбербанк и Северсталь, BBVA, CITIC и Minsheng, и др.

Таким образом, целью данной работы является:

- анализ банковской аккредитивной сферы;
- проектирование системы на основе идей, полученных в ходе анализа.

1. Блокчейн и аккредитив

Для того чтобы понять как доверительную сеть блокчейна можно связать с финансовой сферой аккредитива, надо дать им четкие определения в контексте их взаимодействия друг с другом.

Аккредитив (letter of credit) – условное денежное обязательство, принимаемое банком по поручению приказодателя. Довольно просто можно понять его суть можно на следующем примере: фирма «А» хочет приобрести у фирмы «Б» крупную партию угля. Каким образом они должны провести сделку? Должна ли вначале фирма «А» осуществить перевод? Или необходимо чтобы фирма «Б» первой отгрузила весь товар? Подобные элементарные вопросы и более сложные призвана решить система аккредитива. В случае её использования, фирма «А» передает необходимую сумму банку. Тот ожидает, пока фирма «Б» документально подтвердит передачу товара, и затем осуществляет перевод в пользу фирмы «Б». Таким образом, банк выступает гарантом проведения сделки.

Устоявшееся в мировом сообществе понятие блокчейна сильно отличается от его настоящего «программного» определения. Однако эта условность довольно удобна, так как емко описывает существующие системы. Для целей данной работы рассмотрим блокчейн как базу данных, обладающую рядом отличительных свойств:

- Децентрализованность. Распределенная структура в сочетании с алгоритмами консенсуса гарантирует, что запись информации в такую систему всегда будет корректной.

- Защищенность. Криптографические протоколы обеспечивают неизменность записанной информации.

- Открытость. Это свойство не всегда является преимуществом. Так и в нашем случае – данные о совершенных сделках являются приватной информацией, которая защищена законами РФ. Однако на данный момент существует ряд приемов, позволяющих скрыть приватные части системы, оставив ее полную функциональность.

– Смарт-контракты. Еще одно крайне важное свойство блокчейн систем – это возможность создания и поддержки функционирования смарт-контрактов – алгоритмов, чье поведение можно жестко зафиксировать на уровне сети.

По данному описанию можно сделать вывод, что такое хранилище является отличной основой для проведения аккредитивной сделки.

Однако теперь появляется очень важный вопрос, который зачастую встает при проектировании блокчейн систем. Этот вопрос – «зачем?». Зачем нам необходимо применять блокчейн, если существующие банковские системы уже обеспечивают стабильную работу? Для ответа на этот вопрос необходимо рассмотреть проблемы современного банковского аккредитива.

На основании открытых источников [2] можно выявить следующие недостатки аккредитивов:

– Бюрократия. В каждой стране свои правила оформления документов, у каждого банка свои информационные системы. Для компаний, пользующихся аккредитивными сделками, это выливается во множественные затраты на юристов и бумажную работу. В связи с популярностью блокчейна, общее решение на его основе вполне могли бы принять и страны, и банки.

– Длительность. Проблема, которая следует из предыдущей. Для крупных аккредитивных сделок проверка подлинности документов может затянуться на несколько дней.

– Мошенничество. Острая проблема, от которой страдают все участники аккредитива – покупатели, продавцы и даже банки. Существует великое множество разнообразных способов мошенничества, и все они сильно подрывают экономику любого бизнеса.

2. Проектирование системы

Необходимо спроектировать систему, которая бы решала обозначенные ранее проблемы.

Почти все проведенные с помощью блокчейна аккредитивные контракты на данный момент являются тестовыми. То есть они рассчитаны не на массовое использование, а на разовое проведение сделки для демонстрации возможностей и поиска преимуществ. Подробности технической реализации таких контрактов остаются неизвестными, поэтому провести их анализ и использовать результаты для построения нашей системы не получится.

В распределенных системах типа блокчейн постоянно необходимо приходиться к общему решению в вопросе записи новой информации. Для этого применяются различные алгоритмы консенсуса. Ключевая разница в этих алгоритмах находится в способе добавления новых записей, а также в

выборе тех, кто эти записи может добавлять (валидаторов). Чтобы понять, какой алгоритм подходит нам, рассмотрим возможные варианты:

- Proof of work. Самый известный и распространенный алгоритм. Для своей работы он использует систему майнинга. Это означает, что валидатором может стать кто угодно. Все, что для этого нужно – подключиться к сети и затрачивать свои вычислительные мощности, взамен получая вознаграждение в виде цифровых активов сети.

- Proof of stake. Довольно известный алгоритм, которым стремятся заменить энергозатратный PoW. Его особенность в том, что пишущими узлами могут становиться только те, кто имеет достаточное количество цифровых активов.

- PBFT (Practical Byzantine Fault Tolerance) и его модификации. Алгоритмы, решающие задачу византийских генералов [3], которые обеспечивают корректное поведение участников в асинхронной среде. Его главное отличие – в том, что для функционирования ему нужен набор заранее известных валидаторов, и отсутствует необходимость в существовании цифровых активов.

По ряду причин последний вариант является наиболее подходящим для нашей задачи:

- Для работы протокола нет необходимости во встроенной криптовалюте. В контексте организации банковской операции на данный момент невозможно совместить криптовалюту, которая сложно поддается законодательной регуляции.

- К проектируемой системе должен быть обеспечен стабильный доступ. И учитывая особенности архитектуры PBFT, валидаторами могут стать банки, тем самым гарантируя надежность. Они будут затрачивать свои ресурсы для обеспечения проведения сделки, взамен предоставляя услуги по ее проведению.

- По заявлениям разработчиков блокчейн-платформ, основанных на PBFT консенсусе, пропускная способность транзакций их систем составляет до 3500 транзакций в секунду, против 15 Ethereum [4], который использует proof of work. Однако, стоит отметить, что пока ни одна из этих платформ не может сравниться в количестве пользователей с Ethereum.

Сформировав «backend» для нашей системы нам также будет необходимо средство, с помощью которого пользователь сможет получить доступ к ней. В рамках данной работы планируется реализация веб-приложение для таких нужд, которое может быть размещено на серверах какого-либо регулирующего органа. Однако учитывается предположение, что для оптимизации бизнес процессов организации будут создавать собственные приложения для работы с распределенным реестром.

На основании вышесказанного мы можем спроектировать диаграмму развертывания для нашей системы, которую можно видеть на рис. 1.

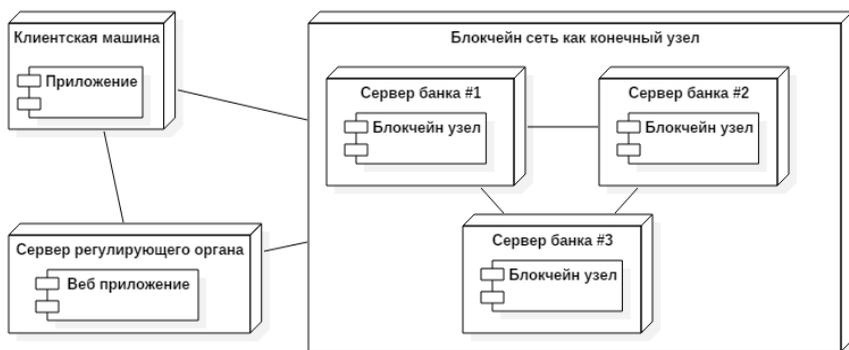


Рис. 1. Архитектура спроектированной системы

На данный момент такая система теоретически способна решить первую и вторую ранее описанные проблемы аккредитивной области.

Остается еще одна проблема – мошенничество. Эта проблема возникает, когда договор был заключен с непроверенной компанией, цель которой была не завершение контракта, а обман других участников и извлечение из этого прибыли. Можно было бы предложить множество способов для борьбы с разными видами мошенничества, однако есть один, с помощью которого можно избежать любого обмана – не заключать контракты с непроверенными компаниями. В этом могла бы помочь рейтинговая система, которая отображала бы насколько хорошо та или иная компания зарекомендовала себя.

Однако при создании такой системы встает вопрос – кто будет хранить и контролировать рейтинговую информацию? В наши дни мы крайне часто сталкиваемся с тем, что централизованные сервисы подвергаются хакерским атакам или намеренно обманывают своих пользователей. Справиться с этим поможет система, которая будет функционировать автоматически, исключая человеческий фактор. Как раз для таких систем и пригодно окружение блокчейн. Смарт-контракты позволят создать независимую рейтинговую систему, вычисляющую рейтинг на основе проведенных сделок по заранее заданным алгоритмам.

3. Реализация

Под наши архитектурные нужды подходит существующая блокчейн платформа Hyperledger Fabric. Это проект от IBM и Digital Asset с открытым исходным кодом, предоставляющий модульную архитектуру

для построения блокчейн решений. Его гибкость позволит нам создать необходимые условия для функционирования аккредитивной системы.

В данной платформе смарт-контракты называются чейнкод. Для его написания можно использовать компилируемый язык программирования Go. Программа запускается на сервере валидатора в защищенном Docker-контейнере. Обращаясь к серверу, мы можем вызывать нужные нам методы чейнкода.

В процессе проектирования для создания аккредитива в цифровом виде была создана система из нескольких чейнкодов. Кодовая часть одного из них описана ниже (см. листинг).

Листинг

Чейнкод для аккредитива

```
type SmartContract struct {
}

type loc_1 struct {
    buyer_pk string `json:"buyer_pk"`
    supplier_pk string `json:"supplier_pk"`
    bank_pk string `json:"bank_pk"`
    tc_pk string `json:"tc_pk"`
    stage int `json:"stage"`
    docs_hash []string `json:"docs_hash"`
}

func (s *SmartContract) Invoke(APIStub
shim.ChaincodeStubInterface) sc.Response {
    ...
}

func (s *SmartContract) create_loc_1(APIStub
shim.ChaincodeStubInterface, args []string) sc.Response {
    if len(args) != 5 {
        return shim.Error("Incorrect number of
arguments. Expecting 5")
    }
    var loc = loc_1{buyer_pk: args[1], supplier_pk: args[2],
bank_pk: args[3], tc_pk: args[4], stage: '0', docs_hash:
[]string}
    loc_as_bytes, _ := json.Marshal(loc)
    APIStub.PutState(args[0], loc_as_bytes)
    return shim.Success(nil)
}
```

В данном листинге описаны две структуры – SmartContract для смарт-контракта и loc_1 для аккредитива. При любом обращении к чейнкоду будет вызван метод Invoke. Метод create_loc_1 содержит бизнес логику по созданию аккредитива в сети.

Заключение

Для функционирования в реальных условиях, этой системе понадобится дополнительная проработка структуры с точки зрения юридических норм и экономической целесообразности. Однако на основании проведенного анализа и спроектированной архитектуры можно сказать, что технология блокчейн уверенно развивается, и уже сейчас способна решать задачи по поддержке enterprise решений.

Список литературы

1. Wirex, 20 областей применения Блокчейн вне финансовых сервисов, ч. 1 [Электронный ресурс]. – Режим доступа : <https://geektimes.ru/company/wirex/blog/281140/>
2. Расчеты и операционная работа в коммерческом банке, Аккредитивы — практика и перспективы [Электронный ресурс]. – Режим доступа : http://www.reglament.net/bank/raschet/2005_11_article_1.htm
3. Mike Ward, Transactions Per Second [Электронный ресурс]. – Режим доступа : <https://medium.com/corda/transactions-per-second-tps-de3fb55d60e3>
4. Задача византийских генералов [Электронный ресурс] : Википедия. – Режим доступа : https://ru.wikipedia.org/wiki/Задача_византийских_генералов

МОДЕЛЬ ПЕРЕДАЧИ ИНФОРМАЦИИ В КВАНТОВОЙ СЕТИ

Н. С. Орехов

Студент

А. Ф. Клиньских

Профессор

Введение

Актуальность работы связана с проблемой квантовых вычислений. На сегодняшний день существуют квантово-криптографические комплексы защиты информации, они успешно протестированы и готовятся к внедрению.

В данной работе проведен анализ известной в литературе модели квантовой передачи данных [1-3].

1. Квантовая сеть

Квантовая сеть – коммуникационная сеть, состоящая из пространственно-разнесенных узлов, в которые помещены индивидуально управляемые кубиты, и квантовых коммуникационных каналов, соединяющих эти узлы. Обмен информацией внутри сети выполняется путем пересылки кубитов по каналам. Физически такие сети могут состоять, например, из кластеров или захваченных в ловушки ионов, представляющие собой узлы, а также оптических волокон или каких-нибудь устройств, передающих фотоны, что обеспечивает реализацию квантовых каналов, как показано на рис. 1.

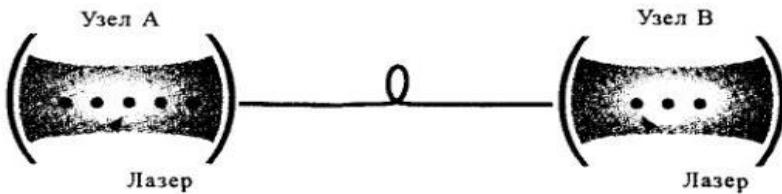


Рис. 1. Физическая модель квантовой сети

Цель работы – численное решение системы обыкновенных дифференциальных уравнений для амплитуд с целью обеспечить условие идеальной передачи данных. При этом система должна решаться при заданных начальных условия и с сохранением условия нормировки.

2. Модель передачи данных в квантовой сети

Рассмотрим простую теоретическую модель, предложенную Джейнсом (Jaynes) и Каммингсом (Cummings) на заре лазерной физики. В рамках такой модели атомы описываются как двухуровневые системы типа частицы со спином $\frac{1}{2}$, которые взаимодействуют с квантовым осциллятором. В экспериментах в Боулдере [1] внутренние степени свободы ионов (отвечающие за «спин») обмениваются энергией с внешним движением частиц в ловушке, представляющим собой механические колебания, квантами которого являются фотоны. Процесс обмена осуществляется в результате освещения ионов лазерными пучками с определёнными частотами, таким образом, вызываются переходы между внутренними состояниями иона и одновременное испускание или поглощение фотона [1]. В парижских экспериментах двухуровневые атомы взаимодействуют непосредственно с модой микроволнового поля в резонаторе, также представляющей собой квантовый осциллятор, квантами которого являются фотоны. В данном случае взаимодействие отвечает за

процессы поглощения и испускания фотонов атомами, пролетающими через резонатор.

На рис. 2а показано, как «расписывается» («signature») ион, осциллирующий в боулдеровской ловушке в суперпозиции состояний с различным числом квантов движения, а на рис. 2б показана соответствующая «подпись» атомов после резонансного взаимодействия в резонаторе со слабо осциллирующим электромагнитным полем, находящимся в суперпозиции состояний с разным числом фотонов.

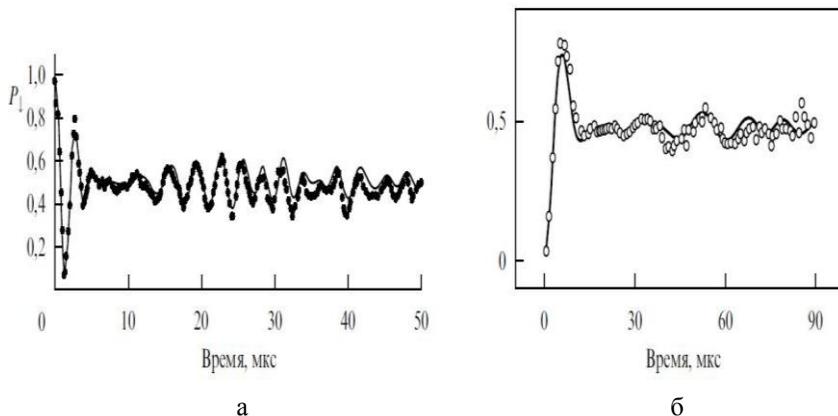


Рис. 2. «Подпись» иона в резонаторе: а – «расписание» иона, б – «подпись» иона

Такие сигналы, представляющие собой вероятность перехода между двумя состояниями иона или атома в зависимости от времени, называются «осцилляциями Раби» («Rabi oscillations»). Вид этих зависимостей соответствует биениям между двумя синусоидальными осцилляциями, отвечающими состоянием с разным числом квантов, и даёт информацию о разрежённости энергетического спектра механического или электромагнитного осциллятора, с которым связаны атома или ионы. Использование подобных осцилляций Раби играет важную роль в управлении состояниями атомов и полей для обработки квантовой информации.

3. Система обыкновенных дифференциальных уравнений для модели передачи данных в квантовой сети

В [2, Гл. 6] приведена система обыкновенных дифференциальных уравнений:

$$\left\{ \begin{array}{l} \alpha_1(t) = \frac{g_1(t) * \beta_a(t)}{\sqrt{2}}, \\ \alpha_2(t) = \frac{g_2(t) * \beta_a(t)}{\sqrt{2}}, \\ \beta_a(t) = -\frac{g_1(t) * \beta_a(t)}{\sqrt{2}} + \frac{g_2(t) * \beta_a(t)}{\sqrt{2}}; \end{array} \right. \quad (1)$$

где $\alpha_{1,2}(t)$ – амплитуды вероятностей нахождения системы в квантовом состоянии ($\langle 0 \rangle$ или $\langle 1 \rangle$); $\beta_a(t)$ – амплитуда вероятности нахождения системы в симметричном или антисимметричном состоянии; $g_{1,2}(t)$ – связи между состояниями системы ($\alpha_{1,2}$ и β_a).

Условия идеальной квантовой передачи данных. Идеальная квантовая передача произойдет при

$$\alpha_1(-\infty) = \alpha_2(+\infty) = 1. \quad (2)$$

Условие идеальной передачи данных (темного состояния) подразумевает, что $\beta_s(t) = 0$ при нормировке

$$|\alpha_1|^2 + |\alpha_2|^2 + |\beta_a|^2 = 1. \quad (3)$$

Исследования заселенности уровней. В качестве примера в программе, написанной в математическом пакете «Maple», была решена система обыкновенных дифференциальных уравнений со следующими параметрами:

$$g_1(t) = \ell^2 t, \quad g_2(t) = 2 * t, \quad l_1 = 0, \quad l_2 = 1, \quad (4)$$

где $l_{1,2}$ – границы временного отрезка.

В качестве примера работы программы на рис. 3 приведены графики заселенностей уровней при следующих начальных условиях: $\alpha_1(0) = \alpha_2(1) = 1$.

Заключение

Была проанализирована известная модель идеальной квантовой передачи информации в системе двух узлов, соединенных оптическим каналом связи (оптоволокно). Каждый узел представлял собой трехуровневый атом, два вырожденных основных состояния которого формировали кубит. Динамика переходов между ними определяется рамановскими переходами под действием лазерного излучения, огибающая которого меняется по заданному закону.

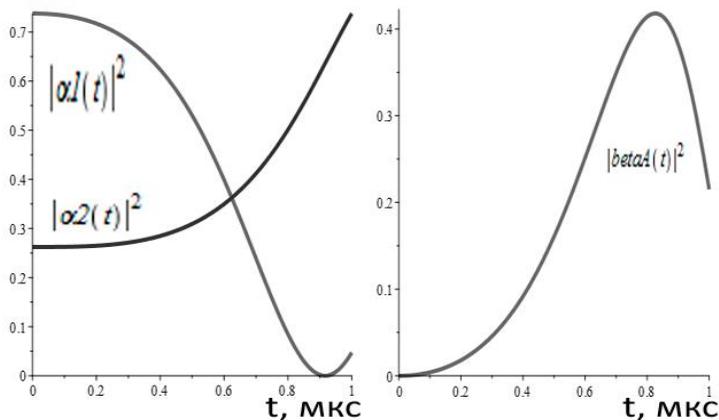


Рис. 3. Графики заселенностей уровней

Основной задачей являлось обеспечение идеальной квантовой передачи информации между узлами квантовой сети. Для решения этой задачи необходимо было найти форму управляющих импульсов в резонаторах. Форма этих импульсов определялась решением системы связанных дифференциальных уравнений. В работе проведен анализ этой задачи для разных форм импульсов. Был реализован алгоритм, проведено моделирование при заданных параметрах с целью иллюстрации процедуры идеальной передачи информации.

Список литературы

1. Арош С. Управление фотонами в ящике и изучение границы между квантовым и классическим / С. Арош // Успехи физических наук. – 2014. – Т. 184. – № 10. – С. 1068-1085.
2. Физика квантовой информации / Д. Бауместер, А. Экерт, А. Цайлингер. – М. : Постмаркет, 2002. – 376 с.
3. Квантовые вычисления и квантовая информация / М. Нильсен, И. Чанг. – М. : Мир, 2006. – 824 с.

ОПРЕДЕЛЕНИЕ МЕСТОПОЛОЖЕНИЯ В ПРОСТРАНСТВЕ С ИСПОЛЬЗОВАНИЕМ ТЕХНОЛОГИИ GPS

Н. В. Пауков

Студент

С. А. Зуев

Доцент

Введение

Научно-технический прогресс не обошёл стороной и такую важную для человечества сферу как навигацию в пространстве. В настоящее время существует несколько систем и технологий, которые позволяют большой точностью определять положение человека или какого-либо предмета с достаточно большой точностью [1].

Использование данных технологий значительно упрощает многие стороны жизни, а также существенно экономит такой ценный ресурс как время. Имея в своём распоряжении устройства для позиционирования (навигаторы), любой человек может с лёгкостью ориентироваться в незнакомых условиях и находить нужные для него места. Применение систем навигации даёт новый толчок для развития транспорта, делает перемещения и путешествия более безопасными, а также позволяет с лёгкостью заранее планировать и анализировать маршруты, по которым будут происходить эти перемещения [2, 3].

1. Цель работы

Цель данной работы — изучить принципы работы современных систем навигации и разработать устройство, позволяющее с достаточно большой точностью определять собственное местоположение в пространстве и получать координаты.

В качестве системы навигации для изучения была выбрана GPS, а разработка устройства проводилась на базе аппаратной платформы Arduino.

Требования к разрабатываемому устройству предъявлялись следующие:

- Определение местоположение в виде географической широты и долготы;
- Определение точного времени;
- Вывод полученных данных на LCD-дисплей (предусмотреть варианты подключения различных дисплеев);

– Возможность добавления нескольких режимов отображения данных, среди которых могут быть скорость перемещения и высота над уровнем моря.

2. GPS (Global Positioning System)

Очень важное место среди существующих на сегодняшний день систем навигации занимает система, носящая название GPS – глобальная система позиционирования, которая была в XX веке разработана военным ведомством США, но после стала доступна для свободного использования в гражданских целях.

Принципы работы GPS основываются на использовании искусственных спутников Земли и существовании так называемого эффекта Доплера, который заключается в изменении воспринимаемой наблюдателем частоты сигнала, если источник этого сигнала или сам наблюдатель находятся в движении.

В настоящий момент число запущенных для успешного функционирования GPS спутников достигает 31. Каждый из этих спутников имеет собственную траекторию движения и располагается на высоте около 20,000 км над поверхностью Земли, совершая каждые сутки два полных оборота вокруг планеты.

Предельная точность получения координат с помощью GPS для коммерческих целей в настоящее составляет порядка 5 метров, чего в большинстве случаев оказывается более чем достаточно.

Использование спутников, полностью покрывающих поверхность Земли, позволяет улавливать сигнал в любой точке поверхности (а также в околоземном пространстве), но имеет и существенные недостатки. Основными среди них являются прямая зависимость качества сигнала от погоды и различных помех (становится хуже в условиях облачности), а также высокая чувствительность к плотным препятствиям типа строительного кирпича и бетона, что делает невозможным использование системы в помещениях при недостаточной чувствительности приёмника.

Для определения текущего местоположения GPS-приёмнику требуется уловить сигнал от нескольких спутников (минимально от четырёх). По задержке распространения сигнала происходит определение расстояния от приёмника до каждого из доступных спутников. Далее можно определить примерную область пространства, в которой в данный момент времени располагается приёмник путем построения нескольких сфер с найденным радиусом и нахождения их пересечения. Так как вследствие атмосферных эффектов, а также движения самих спутников и приёмника найденное расстояние всегда имеет некоторую погрешность, то приёмник пытается добавить к построенной системе данные

дополнительных спутников. Чем больше спутников ‘видит’ приёмник GPS, тем точнее будет определено его положение.

3. Конструирование устройства для определения координат

Для сборки данного устройства и работы с системой позиционирования на практике необходимы следующие компоненты:

- Arduino UNO R3;
- Макетная плата;
- GPS-модуль Ublox NEO-6M V2;
- LCD-дисплей 16x2 (для отображения времени и координат);
- Тактовая кнопка (для переключения режимов отображения данных);
- Резисторы;
- Соединительные провода.

В целях обеспечения связи и обмена данными между Arduino и GPS-модулем используется протокол UART (Universal Asynchronous Receiver-Transmitter), который позволяет преобразовывать данные так, чтобы их можно было передавать по одной физической линии между двумя устройствами. Скорость передачи данных при использовании данного протокола принято измерять в бодах (число изменений состояния канала связи в секунду).

Для передачи и приёма данных у GPS-модуля будут использоваться выводы Rx (Receiver, получатель) и Tx (Transmitter, передатчик), которые будут соединяться с выводами 1 и 0 на Arduino, которые принимаем соответственно за Tx и Rx (обратный порядок нужен для того, чтобы данные передающего вывода всегда попадали на принимающий).

Исходя из требований, предусмотрим возможность подключения одного или нескольких типов дисплеев. Для этого будем резервировать несколько выводов на Arduino и использовать универсальную библиотеку LiquidCrystal [4]. В качестве основного дисплея используем LCD1602, который подключается по шине I2C для того, чтобы сократить количество занимаемых на Arduino выводов. I2C представляет собой последовательную асимметричную шину, которая довольно проста в использовании, но имеет относительно небольшую скорость передачи данных.

При подключении дисплея по I2C будут использоваться два аналоговых вывода (A4 и A5) [5]. Также, нужно установить на макетной плате тактовую кнопку, контакт которой соединяем с выводом под номером 2.

При запуске разрабатываемой системы, пока происходит старт GPS-модуля и координаты ещё не определены, будем выводить на дисплей сообщение с просьбой подождать. В случае, когда GPS-модуль не найден

или не передаёт корректных данных, то покажем сообщение об ошибке. Если данные с модуля получается успешно принять и обработать, то будем выводить нужную информацию в зависимости от режима вывода, который будет переключаться при нажатии кнопки. Данные на дисплее будут обновляться с заданной в исходном коде программы периодичностью.

Для приёма и обработки GPS данных будем использовать библиотеки SoftwareSerial [6] (позволяет реализовать последовательную передачу данных на любых цифровых выводах) и TinyGPS [7] (для извлечения требуемых нам данных с модуля).

Схема соединения всех используемых компонентов представлена на рис. 1.

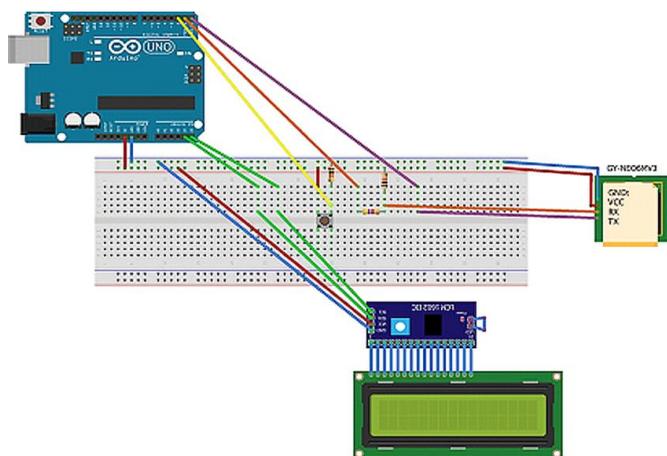


Рис. 1. Схема соединения компонентов для устройства определения координат

Полный исходный код проекта находится на GitHub [8]. Пример вывода полезной информации (координат и времени) показан на рис. 2 и 3.



Рис. 2. Географические координаты одной из аудиторий ФКН



Рис. 3. Точное московское время

Заключение

Данная статья посвящена изучению возможностей и основных принципов работы современных систем навигации в пространстве. Описана система GPS и её особенности. Разработано устройство, применяющие изученные принципы и технологии на практике и позволяющее ориентироваться в пространстве и определять собственное местоположение.

Список литературы

1. Козловский, Е. Искусство позиционирования / Е. Козловский // Вокруг света. – 2006. – № 12. – С. 204-280.
2. Принципы спутниковой навигации [Электронный ресурс] : Электрон. журн. – Режим доступа : <https://www.glonass-iac.ru/guide/navfaq.php>
3. Система GPS. Взгляд изнутри и снаружи [Электронный ресурс]. – Режим доступа : <https://www.ixbt.com/car/gps/gps.html>
4. Arduino – LiquidCrystal [Электронный ресурс]. – Режим доступа : <https://www.arduino.cc/en/Reference/LiquidCrystal>
5. Arduino – LiquidCrystal I2C [Электронный ресурс]. – Режим доступа : <https://github.com/fdebrabander/Arduino-LiquidCrystal-I2C-library>
6. Arduino – SoftwareSerial [Электронный ресурс]. – Режим доступа : <https://www.arduino.cc/en/Reference/SoftwareSerial>
7. Arduino – TinyGPS [Электронный ресурс]. – Режим доступа : <https://github.com/mikalhart/TinyGPS>
8. GPS_Navigator [Электронный ресурс]. – Режим доступа : https://github.com/n-paukov/CSF_GPS_Navigator

GPS-ТРЕКЕР С ИСПОЛЬЗОВАНИЕМ GSM И ARDUINO

Н. В. Пауков

Студент

С. А. Зуев

Доцент

Введение

Технологии мобильной связи прочно вошли в жизнь современного человека, упрощая и совершенствуя многие её аспекты. Используя такие технологии, два человека могут связаться и провести беседу в реальном времени, находясь на огромном расстоянии друг от друга. Помимо голосовой связи стали доступны услуги отправки мгновенных сообщений, а также подключения к всемирной сети Интернет.

Данная тема представляет огромный исследовательский интерес, а также, несмотря на большое количество уже существующих технических устройств и различных сервисов, всё еще находятся такие сферы общественной жизни, для которых применение технологий мобильной связи имеет широкие перспективы.

1. Цель работы

В качестве основной цели работы выбрано изучение систем мобильной связи и способов передачи различных данных на большие расстояния с помощью таких систем.

Для применения изученной информации на практике будет разработано устройство, позволяющее определить координаты пользователя в пространстве и предоставляющее возможность передать их на расстояние с использованием мобильной сети. Передача может осуществляться какому-либо абоненту данной сети или указанному Web-серверу.

Разработка такого устройства помимо учебных целей может также найти полезное применение в некоторых сферах общественной жизни. К таким сферам можно отнести:

- работу спасательных служб (устройство позволит человеку, который попал в бедствие, отправлять таким службам тревожный сигнал, содержащий время и местоположение, что значительно облегчит процесс поиска и оказания помощи);
- фитнес, оздоровительную физкультуру (к примеру, существует возможность создания системы, помогающей пользователю отслеживать

прогресс при занятии беговыми видами спорта, а также строить маршруты и планировать время).

Исходя из поставленных целей, можно определить следующие требования к разрабатываемому устройству:

- Возможность нахождения координат пользователя и текущего времени (для этого будет использована технология спутниковой навигации GPS);
- Возможность по запросу пользователя передавать полученные данные какому-либо абоненту мобильной связи или посылать их для обработки указанному Web-серверу;
- Предусмотренная возможность добавлять и передавать новые виды данных без перепроектирования устройства.

2. Системы мобильной связи и принципы их работы. GSM

Мобильная связь — способ связи между двумя объектами с использованием радиоканала. Данный способ в качестве одного из преимуществ имеет отсутствие необходимости проводного соединения между конечными абонентскими устройствами.

Существуют различные разновидности и классификации систем мобильной связи. Из их числа будем рассматривать те, которые находят наиболее активное применение в повседневной жизни:

- Пейджинговые системы с односторонней связью (клиентские устройства способны принимать сообщения, но лишены возможности передавать данные). Принцип работы таких устройств заключается в постоянном анализе определенной частоты радиоканала и реакции в случае обнаружения в сигнале своего идентификатора (номера).

- Пейджинговые системы с двусторонней связью;
- Системы мобильной сотовой связи.

Наиболее распространенными и известными в настоящее время на территории России являются системы сотовой связи GSM.

Основной принцип их работы заключается в разделении общей зоны предоставления услуг связи на некоторое количество ограниченных секторов, за каждый из которых отвечает собственная базовая станция. Затем, для координации работы, базовые станции подключаются к коммутатору через специализированный контроллер. При перемещении абонента из одного сектора в другой задача его обслуживания автоматически переходит к соответствующей базовой станции.

Таким образом, достигается возможность формировать обширные зоны покрытия, при нахождении в которых абоненты могут эффективно пользоваться возможностями мобильной связи.

За время, прошедшее с моментов первых испытаний таких систем связи, было разработано множество различных технологий и стандартов, которые принято группировать по поколениям систем сотовой связи.

Системы первого поколения (1G, first generation), будучи аналоговыми, отличались сравнительно высокой стоимостью обслуживания и имели низкую скорость передачи данных (не более 5-6 Кбайт/с). Сообщения, передаваемые по каналам связи, были подвержены действию шумов, искажений и не отличались высоким уровнем защищенности. Более того, на протяжении долгого времени отсутствовали общие международные стандарты, которые описывали бы работу таких систем [1].

Ощутимый вклад в борьбу с перечисленными недостатками внесло опубликование в 1991 году спецификации GSM (по названию группы специалистов, занимающихся разработкой, позже изменено на Global System for Mobile Communications). Новый стандарт предполагал использование цифровых сигналов вместо аналоговых. Вводилась возможность использовать множественный доступ по времени (разным абонентам выделяются короткие промежутки времени) и по частоте (разные абоненты используют разные частоты из выделенного набора). Также предполагалось использование одного из четырех вариантов диапазонов частот (890-915 МГц, 935-960 МГц, 1710-1785 МГц, 1805-1880 МГц). Вводилась возможность передавать короткие текстовые сообщения (SMS), для решения проблемы медленной и дорогой передачи данных позже была введена особая надстройка над GSM, которая получила название GPRS (General Packet Radio Service) [2].

GPRS предоставляет абонентам мобильных сетей инструмент для пакетной передачи данных другим абонентам и внешним узлам, не относящимся напрямую к мобильной сети. Учитывалась также возможность совместной работы GPRS и протоколов TCP/IP, что сделало возможным доступным использование всемирной сети — Интернета.

Несмотря на бурное развитие технологий и появление в дальнейшем третьего (3G) и четвертого (4G) поколений, стандарт GSM всё ещё остаётся одним из самых распространенных, доступных и активно используемых по всему миру. В силу обширной зоны покрытия у GSM, универсальности и возможности передавать самостоятельно сформированный набор данных остановимся именно на этом типе сетей. Так как конструируемое устройство предполагает использование вдали от крупных населенных пунктов и не требует высоких скоростей и большого объема трафика, возможности 3G и 4G сетей оказываются излишними, а возможное отсутствие достаточной зоны покрытия может быть критичным.

3. Разработка устройства определения пользовательских координат.

Передача и сохранение данных

С учетом указанных выше требований, устройство должно быть программируемым и доступным для дальнейшего расширения, а значит, в данном проекте оправдано использование микроконтроллера. В силу простоты, доступности, а также возможности быстрой сборки прототипов и тестовых макетов различных устройств, была выбрана в качестве основной системы на аппаратная платформа Arduino.

Помимо непосредственно Arduino и расходных материалов (резисторы, соединительные провода), для сборки будем использовать следующие компоненты:

- GSM/GPS модуль SIMCom SIM808 с поддержкой GPRS. Данный модуль поддерживает все диапазоны GSM-частот, а также отличается низким энергопотреблением;

- LCD-дисплей 16x2 (для отображения времени и координат);

- Тактовая кнопка (для выполнения отправки данных при клике).

Схема соединения всех используемых компонентов представлена на рис. 1.

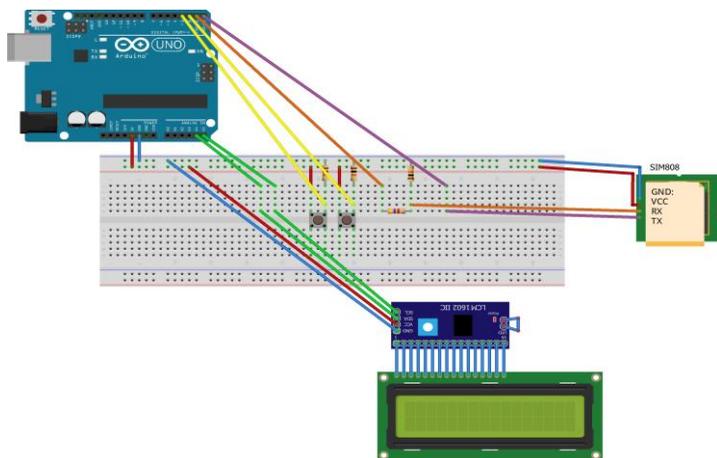


Рис. 1. Схема соединения компонентов для устройства передачи данных

Для управления модулем SIM808 используется стандарт AT-команд (от англ. “attention”) американской корпорации AT&T, который в силу своего удобства и универсальности был принят разработчиками GSM. Помимо стандартного набора команд, производители часто добавляют в устройства поддержку специфических инструкций.

AT-команды представляют собой строки в особом формате. Каждая такая строка должна начинаться с символов “AT”, за которыми следует описание команды. Для проверки корректного подключения модуля можно использовать команду, состоящую только из “AT”. В случае успешной настройки, модуль в качестве ответа должен вернуть строку “OK”.

Так как AT-команды задаются в текстовом формате, необходим механизм их кодирования и передачи в виде электрических сигналов. Для выполнения этой задачи будем использовать UART (Universal Asynchronous Receiver-Transmitter), который аппаратно поддерживается Arduino (на выводах 0 и 1) и позволяет последовательно организовать прием и передачу данных по цифровым каналам по стандарту RS-232. Прием данных может выполняться одновременно с их отправкой. Для записи и чтения данных с UART будем использовать библиотеку SoftwareSerial [3].

Рассмотрим использование AT-команд для получения GPS-данных с модуля SIM808. Во-первых, при инициализации программы нам потребуется активировать встроенный в SIM808 GPS-приемник. С этой целью выполняем команду «AT+CGNSPWR=1». Затем включаем передачу GPS-данных по стандарту NMEA с помощью команды: «AT+CGNSINF». После этого мы должны будем выполнять чтение сообщений модуля с Rx-вывода (receiver) Arduino, получать из них необходимую информацию и передавать какому-либо удаленному абоненту или серверу по команде пользователя (клик по тактильной кнопке). Для получения данных из NMEA-сообщений будем использовать библиотеку TinyGPS [4].

Для того чтобы отправить SMS с указанным текстом, нам потребуется выполнить несколько AT-команд: «AT+CMGF=1» для установки текстового режима отправки (проще PDU режима), «AT+CSCS="GSM"» для задания режима кодирования и «AT+CMGS="+number"\rText» для отправки SMS на указанный номер мобильного телефона (number).

В качестве второго режима отправки мы будем использовать отправку GET запроса на указанный в программе Web-сервер с помощью GPRS. Для этого будем также использовать AT-команды. Следует учесть, что при использовании метода GET параметры указываются непосредственно в URI, поэтому эту строку можно будет подготовить заранее.

Для передачи запроса с помощью GPRS следует выполнить несколько действий.

Во-первых, необходимо установить настройки для включения возможности выхода в Интернет. Для этого будем выполнять следующие команды в порядке их перечисления (листинг 1) (настройки доступа следует уточнять у используемого оператора сотовой связи).

Настройки для включения возможности выхода в Интернет

```

AT+SAPBR=3,1,"CONTYPE","GPRS" (указываем тип подключения)
AT+SAPBR=3,1,"APN","internet.mts.ru" (указываем APN - имя
точки доступа)
AT+SAPBR=3,1,"USER","mts" (указываем имя пользователя)
AT+SAPBR=3,1,"PWD","mts" (указываем пароль пользователя)
AT+SAPBR=1,1 (выполняем подключение)

```

После того, как мы выполнили подключение к точке доступа, можно заняться непосредственно составлением и отправкой запроса. Для этого выполним такие AT-команды, приведенные в листинге 2.

Составление и отправка запроса

```

AT+HTTPINIT
AT+HTTPPARA="CID",1 (необходимо для настройки сессии)
AT+HTTPPARA="URL","mydomain.ru/gps_handler.php?gps=1&lat=<latitude>&lon=<longitude>&time=<time>" (указываем URI, здесь lat, long, time - географические координаты и время соответственно)
AT+HTTPACTION=0 (устанавливаем метод запроса как GET)
AT+HTTPTERM (завершение работы HTTP, перед выполнением этой команды мы также можем считать и обработать ответ сервера, если это будет необходимо)

```

Часть настроек (включение GPS-приемника, установка параметров доступа в интернет) мы будем выполнять в момент запуска разрабатываемого устройства.

Далее, мы будем постоянно обновлять текущее время и координаты, а затем отображать их на дисплее. Для отправки данных на дисплей используем универсальную библиотеку LiquidCrystal [5], предназначенную для работы с LCD.

В любой момент времени у пользователя с помощью нажатия на одну из двух кнопок будет возможность выполнить отправку актуальных в текущий момент данных, по SMS (рис. 2) (абоненту с указанным номером), либо на Web-сервер (по указанному URI). В каждом случае мы вызываем соответствующую функцию отправки, а затем показываем пользователю сообщение со статусом отправки (успех или ошибка).

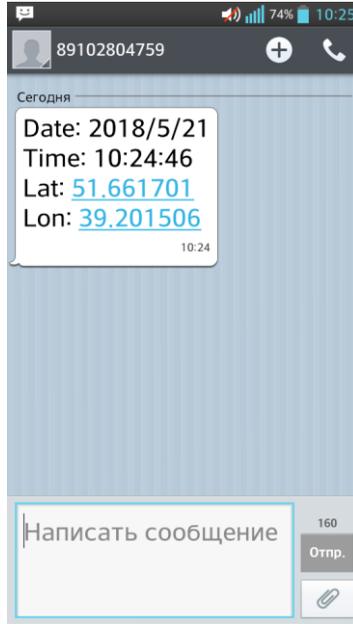


Рис. 2. Переданные с устройства координаты

Заключение

Данная статья посвящена изучению технологий мобильной связи и принципов установления связи на расстоянии. Описаны основные особенности GSM и GPRS. Разработано устройство, предоставляющее пользователю возможность получения своего положения и передачи его другому абоненту или Web-серверу. Определены сферы применимости и перспективы подобных разработок.

Список литературы

1. Ратынский М. Основы сотовой связи / М. Ратынский. – М. : Радио и связь, 2000. – 248 с.
2. Данилов В. Сотовые телефонные сети стандарта GSM / В. Данилов. – СПб. : СПбГУТ, 1996. – 100 с.
3. Arduino – SoftwareSerial [Электронный ресурс]. – Режим доступа : <https://www.arduino.cc/en/Reference/SoftwareSerial>
4. Arduino – TinyGPS [Электронный ресурс]. – Режим доступа : <https://github.com/mikalhart/TinyGPS>
5. Arduino – LiquidCrystal [Электронный ресурс]. – Режим доступа : <https://www.arduino.cc/en/Reference/LiquidCrystal>

ФОРМАЛИЗАЦИЯ И РЕШЕНИЕ ПРОИЗВОДСТВЕННЫХ ЗАДАЧ В КОМПАНИИ «АНАЛИТИКА. КОНСАЛТИНГ. БЕЗОПАСНОСТЬ»

Е. В. Погребецкая

Студент

И. В. Абрамов

Доцент

Введение

Информационные технологии активно вошли в современную жизнь, в том числе в организацию производственного процесса, деятельность которого невозможна без соответствующей системы управления. Динамичность современной корпоративной деятельности требует обдуманного подхода к организации системы управления, повышение эффективности которого можно достигнуть при уместном и рациональном использовании совокупности информационных технологий. Выбор оптимальной последовательности выполнения заказов является одним из критериев успешности компании. В реальности не всегда просто ранжировать заказы с учетом многих критериев и при этом делать это эффективно. Другой довольно частой проблемой управления, влияющей на эффективность работы компании, является назначение задач на исполнение сотрудникам с целью минимизации общего времени выполнения всех задач.

1. Постановка и решение задачи о ранжировании заказов в организации

Вычисление оптимальной последовательности выполнения заказов является задачей ранжирования. Решение задачи ранжирования заказов происходит в 2 этапа:

1. Оценка полезности каждого заказа;
2. Ранжирование заказов.

Проанализировав данные каждого из заказов, а также текущую ситуацию в компании, можно определить степень важности каждого, что позволит вычислить их приоритеты в рамках очереди выполнения заказов в компании и ранжировать заказы, сформировав оптимальную последовательность их выполнения. Для принятия правильного решения о возможности и целесообразности выполнения поступившего заказа и выбора оптимальной последовательности выполнения заказов необходимо проанализировать внутренние особенности предприятия.

В результате экспертного анализа производственного процесса и устройства рассматриваемого предприятия были выделены следующие критерии:

1. Степень срочности заказа
2. Прибыльность заказа для компании
3. Новый клиент

Следует пояснить, что желание компании показать новому клиенту себя с хорошей стороны довольно велико, поэтому необходимо учитывать и этот критерий.

Решение сводится к методу анализа иерархий (МАИ), включающему следующие этапы [1]:

Этап 1. Структурируем проблему выбора заказа в виде иерархии, показанной на рис. 1 [2].

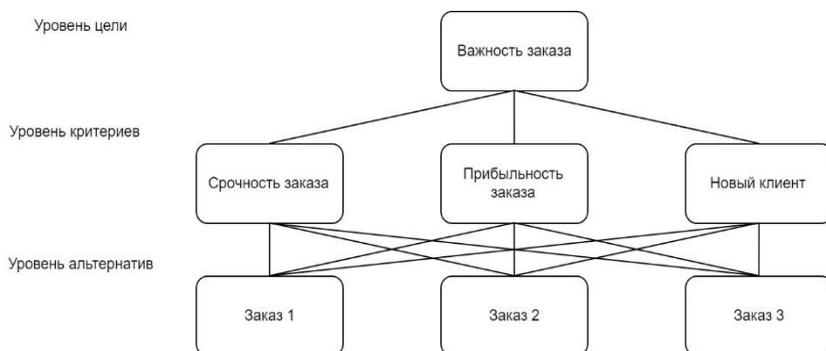


Рис. 1. Иерархическая структура МАИ

Этап 2. Оценим значимость каждого критерия.

Чтобы вычислить вес каждого из критериев необходимо составить матрицу A попарных сравнений критериев. Матрица A является вещественной матрицей $m \times m$, где m – количество критериев. Каждый элемент a_{jk} матрицы A отражает значимость j -го критерия относительно k -го критерия. Если $a_{jk} > 1$, то j -й критерий более важен, чем k -й критерий, а если $a_{jk} < 1$, то j -й критерий менее важен, чем k -й критерий. Если два критерия имеют одинаковую значимость, то элемент a_{jk} равен 1. Очевидно, что $a_{jj} = 1$ для всех j . Элементы a_{jk} и a_{kj} удовлетворяют следующему ограничению:

$$a_{jk} \cdot a_{kj} = 1.$$

Приоритетность оценивается по следующей шкале [1]: 1 – одинаковая значимость, 3 – некоторое преобладание значимости одного критерия перед другим (слабая значимость), 5 – существенная или сильная

значимость, 7 – очень сильная или очевидная значимость, 9 – абсолютная значимость, 2, 4, 6, 8 – промежуточные значения между соседними значениями шкалы. В результате экспертных оценок получаем матрицу A попарных сравнений рис. 2.

	Срочность заказа	Новый клиент	Прибыльность заказа
Срочность заказа	1	6	3
Новый клиент	1/6	1	1/5
Прибыльность заказа	1/3	5	1

Рис. 2. Матрица A попарных сравнений критериев

Эман 3. Определим весовой вектор критериев.

Существует несколько способов получить грубые оценки этого вектора-столбца, но мы воспользуемся способом, в котором необходимо умножить n элементов каждой строки матрицы A и извлечь корень n -й степени, а затем нормализовать полученные числа. Получается следующий вектор весов $w = [0.635 \ 0.078 \ 0.287]^T$.

Эман 4. Применим метод парных сравнений для оценки альтернатив по каждому критерию. Способ нахождения векторов весов альтернатив аналогичен способу, описанному на 3 этапе. Были получены следующие весовые векторы альтернатив по критериям: $s^{(1)} = [0.218 \ 0.091 \ 0.691]^T$, $s^{(2)} = [0.075 \ 0.183 \ 0.742]^T$, $s^{(3)} = [0.634 \ 0.192 \ 0.174]^T$.

Итоговая матрица оценок альтернатив по критериям имеет вид:

$$S = [s^{(1)} \ s^{(2)} \ s^{(3)}] = \begin{bmatrix} 0.218 & 0.075 & 0.634 \\ 0.091 & 0.183 & 0.192 \\ 0.691 & 0.742 & 0.174 \end{bmatrix}.$$

Эман 5. Найдём общую оценку каждой альтернативы. Формула нахождения оценок альтернатив имеет вид:

$$v = S \cdot w = [0.326 \ 0.127 \ 0.547].$$

Таким образом, самым приоритетным является третий заказ, в последовательности выполнения заказов он будет выполнен в первую очередь. Стоит обратить внимание, что третий заказ имеет самые худшие оценки по последнему критерию, однако вес этого критерия ниже, чем вес первого критерия, и поэтому не оказывает значительного влияния на результат.

2. Постановка и решение задачи о назначениях

В организации осуществляется планирование и распределение задач между сотрудниками. При этом необходимо минимизировать общее время, затраченное всеми сотрудниками на выполнение всех задач. Все сотрудники имеют разную квалификацию, то есть время, за которое будет сделана одна и та же задача разными сотрудниками, будет различаться.

Частный случай этой задачи для $m = n$, где m – количество сотрудников, а n – количество задач, решается Венгерским методом [3].

Математически задача формулируется следующим образом:

$$\begin{aligned} \min \sum_{i=1}^n \sum_{j=1}^n c_{ij} \cdot x_{ij}, \\ \sum_{i=1}^n x_{ij} = 1, \quad j = 1..n, \\ \sum_{j=1}^n x_{ij} = 1, \quad i = 1..n, \\ x_{ij} \in \{0,1\}, \end{aligned}$$

где $(c_{ij})_{n \times n}$ содержит время выполнения i -м сотрудником j -ой задачи, а $(x_{ij})_{n \times n}$ – факт назначения исполнителю i задачи j [4].

Однако случай $m = n$ в корпоративной среде встречается редко. В рассматриваемой организации необходимо каждому сотруднику назначить несколько задач, поэтому условие $\sum_{j=1}^n x_{ij} = 1$ (каждый сотрудник выполняет только одну задачу) для решения нашей задачи не подходит.

Задача о назначениях является частным случаем транспортной задачи, поэтому для ее решения можно пользоваться соответствующими алгоритмами.

Формальная постановка задачи в организации:

$$\min z = \sum_{i=1}^m \sum_{j=1}^n c_{ij} \cdot x_{ij},$$

при ограничениях

$$\sum_{j=1}^n x_{ij} \geq 1, \quad i = 1..m$$

(каждому сотруднику должно быть назначено минимум по одной задаче) и

$$\sum_{i=1}^m x_{ij} = 1, \quad j = 1..n$$

(каждая задача может быть назначена только одному сотруднику).

Рассмотрим решение задачи на примере используя симплекс-метод [5]. Пусть задана матрица стоимости

$$C = \begin{pmatrix} 4 & 3 & 3 & 2 & 4 & 1 & 4 & 1 \\ 2 & 1 & 1 & 4 & 1 & 2 & 2 & 4 \\ 4 & 4 & 1 & 5 & 1 & 1 & 3 & 2 \\ 3 & 4 & 2 & 4 & 2 & 2 & 1 & 3 \\ 2 & 1 & 4 & 4 & 1 & 4 & 4 & 2 \end{pmatrix},$$

где c_{ij} – время выполнения i -м сотрудником j -ой задачи. Применяя симплекс-метод получаем $z_{min} = 10$ и матрицу назначений

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

В результате анализа бизнес-процессов в организации, удалось оптимизировать и сделать более эффективным назначение задач сотрудникам, что позволит увеличить прибыль компании, так как каждый сотрудник будет выполнять задачи, соответствующие своей квалификации.

Заключение

Данная статья исследует проблемы, стоящие перед компанией, с целью повышения эффективности её работы. Были формализованы задачи, выполняемые в компании неоптимальным путём и решены с использованием методов оптимизации. Описанные решения позволяют улучшить бизнес-процессы в организации, такие как определение очерёдности выполнения заказов и назначение задач сотрудникам, с целью оптимизировать общее время выполнения всех работ. В итоге получились полностью рабочие модули, позволяющие компании оптимально распоряжаться своими ресурсами. Внедрение таких модулей позволит повысить общую прибыль организации, а также продуктивность выполнения задач, так как все сотрудники будут заняты на задачах, подходящих их опыту и квалификации.

Список литературы

1. Саати, Т. Л. Принятие решений. Метод анализа иерархий / Саати Л. Т. – М. : Радио и связь, 1989. – 316 с.
2. The Analytic Hierarchy Process [Электронный ресурс]. – Режим доступа : http://www.dii.unisi.it/~mocenni/Note_AHP.pdf
3. Свободная энциклопедия Википедия [Электронный ресурс]. – Режим доступа : https://ru.wikipedia.org/wiki/Венгерский_алгоритм
4. Свободная энциклопедия Википедия [Электронный ресурс]. – Режим доступа : https://ru.wikipedia.org/wiki/Задача_о_назначениях
5. Свободная энциклопедия Википедия [Электронный ресурс]. – Режим доступа : <https://ru.wikipedia.org/wiki/Симплекс-метод>

ЗАХВАТ ВИДЕО ПОТОКА И ПРЕОБРАЗОВАНИЕ ЦВЕТОВОЙ МОДЕЛИ ЕГО КАДРОВ

Х. А. Полещук

Студент

Н. А. Тюкачев

Доцент

Введение

Обработка видео (в частности, компьютерное зрение) является важным и быстро развивающимся направлением в современном мире. Обработка видео и изображений может выполняться с помощью разных технологий и с разными целями.

Обработка видео в режиме реального времени требует быстрой обработки большого массива данных и, соответственно, оптимизированного приложения.

1. Постановка задачи

Необходимо захватывать видео поток или читать видео файл, преобразовать его кадры в другую цветовую модель (из RGB в YUV) и переслать определенному устройству.

Цветовая модель YUV, в которую надо преобразовать кадры является не стандартной, поэтому не имеется стандартных функций, подходящих для этой такого преобразования. Весь процесс преобразования необходимо написать самостоятельно.

Важно отметить, что при пересылке кадров устройству надо соблюдать частоту, соответствующую формату видео (например, для формата видео высокого качества 720p чаще всего транслируется 60 кадров в секунду, т.к. данная частота обеспечивает высочайшее качество в пределах стандарта ATSC). В данной статье не будет рассматриваться непосредственно работа с устройством, но решение вопроса соблюдения частоты будет представлено.

В данной статье рассмотрен захват видео потока с помощью библиотеки OpenCV, получение кадров из данного потока, а также их обработка.

Для начала необходимо рассмотреть библиотеку OpenCV.

Следующим действием является создание проекта. Пусть это будет C++ приложение. Разработка будет вестись в Microsoft Visual Studio.

2. Библиотека OpenCV

OpenCV – библиотека алгоритмов компьютерного зрения, обработки изображений и численных алгоритмов общего назначения с открытым кодом [1].

Это библиотека, которая до 1-й версии разрабатывалась в Центре разработки программного обеспечения Intel (причём, российской командой в Нижнем Новгороде).

OpenCV написана на языке высокого уровня (C/C++) и содержит алгоритмы для: интерпретации изображений, калибровки камеры по эталону, устранения оптических искажений, определения сходства, анализа перемещения объекта, определения формы объекта и слежения за объектом, 3D-реконструкции, сегментации объекта, распознавания жестов и т. д.

Для более полного понимания поставленной задачи необходимо рассмотреть основные модули библиотеки OpenCV (рисунок).

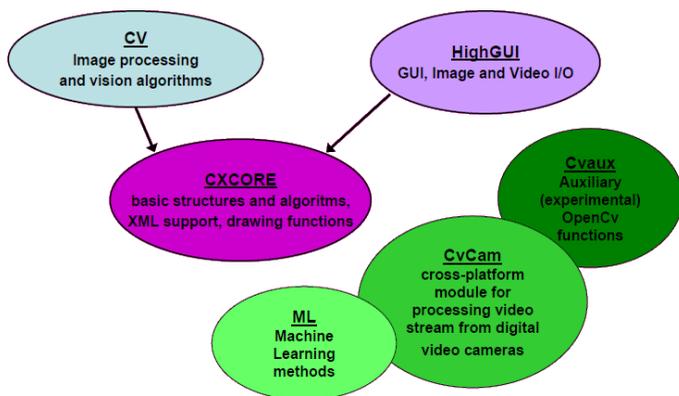


Рисунок. Основные модули библиотеки OpenCV

На рисунке 1 изображены основные модули библиотеки OpenCV:

- Ядро, содержащее базовые структуры данных и алгоритмы (CXCORE);
- Модуль обработки изображения и компьютерного зрения (CV);
- Модуль для ввода/вывода изображений и видео, создания пользовательского интерфейса (HighGUI);
- Экспериментальные и устаревшие функции (CvAux);
- Захват видео (CvCam);
- CXCORE:
- базовые операции над многомерными числовыми массивами;

- матричная алгебра, математические ф-ции, генераторы случайных чисел;

- запись/восстановление структур данных в/из XML;

- базовые функции 2D графики;

CV:

- базовые операции над изображениями (фильтрация, геометрические преобразования, преобразование цветовых пространств и т. д.);

- анализ изображений (выбор отличительных признаков, морфология, поиск контуров, гистограммы);

- анализ движения, слежение за объектами;

- обнаружение объектов, в частности лиц;

- калибровка камер, элементы восстановления пространственной структуры;

HighGUI:

- захват видео с камер и из видео файлов, чтение/запись статических изображений;

- функции для организации простого UI (все демо приложения используют HighGUI);

CvAux:

- пространственное зрение: стереокалибрация, самокалибрация;

- поиск стерео-соответствия, клики в графах;

- нахождение и описание черт лица;

Модуль CvCam позволял захватывать видео с цифровых видео камер (поддержка прекращена и в последних версиях этот модуль отсутствует) [2].

Теперь, когда подготовительные разъяснения по поводу библиотеки OpenCV завершены можно приступить к созданию проекта.

3. Приложение по захвату видео потока и преобразованию цветовой модели его кадров

Сначала необходимо захватить видео (листинг 1).

Здесь video является названием видео файла или индекс камеры, видео поток которой нужно захватить.

Далее задается интервал для регулирования частоты пересылки кадров на устройство (листинг 2). (RATE – необходимая нам частота).

Создаем переменные, необходимы нам в данном приложении (листинг 3).

Забегаю немного вперед, следует отметить, что для оптимизации приложения без подключения оптимизации на уровне компилятора было выбрано использование стандарта OpenMP.

OpenMP – открытый стандарт для распараллеливания программ на языках C, C++ и Фортран. Дает описание совокупности директив компилятора, библиотечных процедур и переменных окружения, которые предназначены для программирования многопоточных приложений на многопроцессорных системах с общей памятью [3].

Для распараллеливания процесса преобразования цветовой модели кадров в данном проекте будет использоваться 12 потоков. Дано количество потоков является наиболее оптимальным по результатам тестов.

Настраиваем все необходимое для использования OpenMP (в том числе в настройках проекта) (листинг 4).

Далее создается бесконечный цикл по получению кадров и преобразованию их цветовой модели.

В нем присутствует цикл, необходимый для регуляции частоты обработки и отправки кадров (листинг 5).

Получаем кадр, проверяем, что он не пустой (например, видео закончилось) и выделяем память под преобразованные данные (листинг 6).

Распараллеливаем блок преобразования с помощью OpenMP (листинг 7).

Далее производится работа с преобразованными кадрами. Например, их пересылка устройству.

Также надо не забывать очищать память и считать время (листинг 8).

Листинг 1

Захват видео

```
VideoCapture capture(video);  
if (!capture.isOpened())  
    return -1;
```

Листинг 2

Задание интервала

```
auto starttime = std::chrono::system_clock::now();  
int interval = 1e9 / RATE;  
auto nexttime = starttime +  
std::chrono::nanoseconds(interval / 2);
```

Листинг 3

Создание переменных

```
Mat frame;  
unsigned char *data;  
unsigned char *yuvData;  
int y, u, y2, v;  
int u1, u2, v1, v2;  
int i, j;  
int row, pixel;
```

Настройка OpenMP

```
omp_set_dynamic(0);
omp_set_num_threads(12);
```

Цикл регуляции частоты обработки

```
while (std::chrono::system_clock::now() < nexttime) {
    std::this_thread::yield();
}
```

Получение кадра

```
capture >> frame;
if (frame.empty())
    return;
data = frame.data;
yuvData = (unsigned char*)malloc(FRAME_SIZE);
```

Цикл преобразования

```
#pragma omp parallel for shared(data, yuvData) private(row,
pixel, y, u1, v1, y2, u2, v2, u, v, i, j)
for (row = 0; row < height; row++) {
    for (pixel = 0; pixel < width; pixel += 2) {
        i = (row * width + pixel) * 3;
        j = i * 5 / 6;
        y = 0.257 * data[i] + 0.504 * data[i + 1] + 0.098 *
data[i + 2] + 16;
        u1 = 0.439 * data[i] - 0.368 * data[i + 1] -
0.071 * data[i + 2] + 128;
        v1 = -0.148 * data[i] - 0.291 * data[i + 1] +
0.439 * data[i + 2] + 128;
        y2 = 0.257 * data[i + 3] + 0.504 * data[i + 4] +
0.098 * data[i + 5] + 16;
        u2 = 0.439 * data[i + 3] - 0.368 * data[i + 4] -
0.071 * data[i + 5] + 128;
        v2 = -0.148 * data[i + 3] - 0.291 * data[i + 4] +
0.439 * data[i + 5] + 128;
        u = 0.5 * (u1 + u2);
        v = 0.5 * (v1 + v2);
        y <<= 2;
        y2 <<= 2;
        u <<= 2;
        v <<= 2;
        yuvData[j] = (u & 0x3fc) >> 2;
        yuvData[j + 1] = (u & 0x3) << 6 | (y & 0x3f0) >> 4;
        yuvData[j] = (u & 0x3fc) >> 2;
```

Окончание листинга 7

```
yuvData[j + 1] = (u & 0x3) << 6 | (y & 0x3f0) >> 4;  
yuvData[j + 2] = (y & 0xf) << 4 | (v & 0x3c0) >> 6;  
yuvData[j + 3] = (v & 0x3f) << 2 | (y2 & 0x300) >> 8;  
yuvData[j + 4] = y2 & 0xff;  
    }  
}
```

Листинг 8

Освобождение памяти и изменение времени следующего преобразования

```
free(yuvData);  
nexttime += std::chrono::nanoseconds(interval);
```

Заключение

В статье было рассмотрено построение приложения, захватывающего видео поток или читающего видео файл, преобразовывающего его кадры в другую цветовую модель (из RGB в YUV).

Преобразование цветовой модели кадров из RGB в YUV обеспечивает меньшую загрузку сети, так как YUV кадры имеют меньший размер.

Рассмотренное в данной статье приложение из-за оптимизации отличается высокой скоростью обработки видео, что позволяет использовать его для преобразования видео в режиме реального времени.

Список литературы

1. OpenCV [Электронный ресурс]. – Режим доступа : <https://ru.wikipedia.org/wiki/OpenCV>
2. OpenCV шаг за шагом. Введение. [Электронный ресурс]. – Электронный документ. – Режим доступа : <http://robocraft.ru/blog/computervision/264.html>
3. OpenMP [Электронный ресурс]. – Режим доступа : <https://ru.wikipedia.org/wiki/OpenMP>

РАЗРАБОТКА ПРОГРАММНОГО МОДУЛЯ ДЛЯ АУГМЕНТАЦИИ ДАННЫХ НА ОСНОВЕ ВНЕСЕНИЯ ИСКУССТВЕННЫХ ИСКАЖЕНИЙ НА РАСТРОВЫЕ ИЗОБРАЖЕНИЯ

М. С. Пузатых

Студент

М. А. Дрюченко

Доцент

Введение

В настоящее время цифровая обработка, анализ и распознавание изображений являются одними из наиболее интенсивно развивающихся направлений научных исследований, что обуславливается широким распространением и практическим использованием систем машинного зрения, видеонаблюдения, моделирования графических объектов в различных отраслях науки и техники. Развитие методов обработки цифровых изображений обуславливается повышением производительности и возрастающими возможностями систем обработки, хранения и передачи информации, а также появлением новых технологий, методов обработки и анализа информации.

Одной из основных проблем при обработке и анализе изображений является наличие разного рода графических искажений, таких как цифровые шумы, аппликативные помехи, локальные изменения уровней яркости и т. д. Наличие искажений затрудняет получение точной измерительной информации и зачастую приводит к ошибкам в алгоритмах обработки. Разработка алгоритмов обнаружения и компенсации искажений должна сопровождаться качественным тестированием с привлечением максимально широкого класса исходных данных, в том числе содержащих различные искажения. Данная работа посвящена разработке и программной реализации комплекса алгоритмов для создания «искусственных искажений» на изображениях микроэлектроники.

1. Моделирование гауссовского шума на изображениях

Аддитивный Гауссов шум характеризуется добавлением к каждому пикселю изображения значений из соответствующего нормального распределения с нулевым средним значением. Такой шум обычно вводится на этапе формирования цифровых изображений. Модель аддитивного шума хорошо описывает действие зернистости фотопленки, флуктуационный шум в радиотехнических системах [1]. Зашумленное

изображение $\tilde{I}(x, y)$ получается путем складывания исходного изображения $I(x, y)$ и шумового поля $J(x, y)$.

$$\tilde{I}(x, y) = I(x, y) + J(x, y). \quad (1)$$

Пример гауссовского шума на растровом изображении, полученного с помощью разработанного программного модуля представлен на рис. 1.

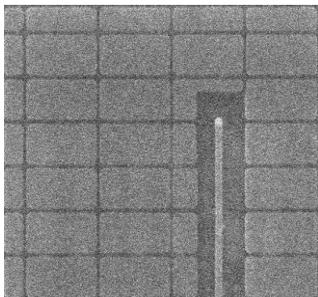


Рис. 1. Пример гауссовского шума

2. Моделирование импульсного шума на изображениях

Импульсный шум характеризуется полным замещением истинной яркости части пикселей значениями фиксированной или случайной величины. На изображении такие искажения выглядят, как изолированные контрастные точки. Импульсный шум характерен для цифровых систем передачи и хранения изображений, для систем передачи изображений по радиоканалам. Основной причиной возникновения импульсного шума являются переходные процессы в аппаратуре, например, неправильная коммутация [2]. Зашумленное изображение $\tilde{I}(x, y)$, получается путем воздействия оператора искажения D на исходное изображение $I(x, y)$.

$$\tilde{I}(x, y) = D(I(x, y)). \quad (2)$$

Пример импульсного шума на растровом изображении, полученного с помощью разработанного программного модуля представлен на рис. 2.

3. Моделирование аппликативных помех на изображениях

Одним из распространенных классов искажений на изображениях являются аппликативные помехи (АП). Появление аппликативных помех связано с появлением различного рода неоднородности. На изображении характеризуются замещением локальных участков полезного изображения фрагментами со случайной амплитудой, формой и текстурой.

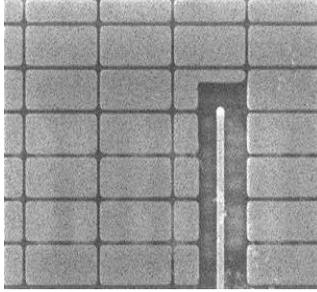


Рис. 2. Пример импульсного шума

Исходное изображение будет рассматриваться, как случайное поле, которое задано на квадратной дискретной сетке. Искаженное изображение будем представлять в виде

$$\tilde{I}(x, y) = (1 - \eta(x, y))I(x, y) + \eta(x, y)w(x, y), \quad (3)$$

где $I(x, y)$ – исходное случайное поле (исходное изображение), $\eta(x, y)$ – случайное бинарное поле, определяющее форму, площадь, расположение и количество локальных областей закрытия (ЛОЗ), $w(x, y)$ – случайное поле, определяющее яркостные локальных областей закрытия. Поле $w(x, y)$ выбрано в виде гауссовской случайного поля. Основной задачей в данном алгоритме является генерация случайного бинарного поля $\eta(x, y)$.

Далее под аппликативной помехой будет иметься ввиду вся совокупности ЛОЗ на изображении. Весь этап внесения аппликативных помех условно можно разбить на две части: этап генерации условных центров ЛОЗ и этап формирования текстуры и формы, каждой отдельно взятой ЛОЗ. Генерация условных центров ЛОЗ описывается пуассоновским распределением с заданной интенсивностью λ . По этой причине количество таких центров на изображении будет подчиняться распределению

$$P(S) = \frac{(\lambda V)^S}{S!} e^{(-\lambda V)}, \quad (4)$$

где V – площадь изображения, $V = (N\Delta x)(N\Delta y)$, где $\Delta x, \Delta y$ – дискреты задания координатной сетки. Параметр λ позволяет управлять количеством условных центров ЛОЗ, следовательно, и количеством самих ЛОЗ. Также все эти центры будут распределены равномерно на изображении [3].

После этапа генерации условных центров ЛОЗ, начинается этап формирования площади каждой отдельной ЛОЗ. Для каждого условного

центра вводится параметр n . Значение этого параметра определяет площадь каждой ЛОЗ. В данном алгоритме для задания этого параметра выбирается пуассоновское распределение с интенсивностью α . С учетом уже выбранного условного центра ЛОЗ

$$P(n) = \frac{\alpha^{n-1}}{(n-1)!} e^{-\alpha}, n \geq 1. \quad (5)$$

$$E[n] = \sum_{n=1}^{\infty} n \frac{\alpha^{n-1}}{(n-1)!} e^{-\alpha} = \frac{S}{\Delta x \Delta y}. \quad (6)$$

где S – средняя площадь ЛОЗ.

После формирования площадей ЛОЗ, алгоритм переходит к формированию формы каждой ЛОЗ.

Генерация формы искажения происходит на бесконечной квадратной решетке. Каждому узлу этой решетки соответствует значение прозрачности $T(x, y) \in [0, 1]$. Далее вводится дополнительный параметр $\tilde{T}(x, y) \in R$, который определяет концентрацию вещества физической среды, которая закрывает изображение в данном узле решетки. Предполагается, что если в конкретной точке сконцентрировано большое количество вещества, то и в окружающих точках повышается ожидаемая концентрация вещества.

$$\tilde{T}(x, y) \sim p_T(x | T, x, y). \quad (7)$$

Данная величина отражает зависимость значений прозрачности ЛОЗ в данной точке от окружающих.

Во время итеративного процесса увеличения ЛОЗ на каждом шаге происходит выбор следующего узла для искажения. В алгоритме используется маска расширения. Своим центром маска помещается на уже искаженный пиксель, все пиксели, которые попадают в зону расширения маски являются кандидатами на искажение. Применяя данную маску ко всем искаженным пикселям, получаем координаты всех пикселей, которые являются потенциальными кандидатами на искажение на данной итерации. При выборе следующего пикселя, который будет искажен используется следующий подход. Каждому кандидату на искажение ставится в соответствие величина, которая обновляется на каждом шагу $p_{dist}(i, j)$. Она определяет вероятность выбора данного пикселя для искажения. Также в алгоритме дополнительно вводится поле D , которое определено в узлах решетки и содержит всю информацию для обновления величины $p_{dist}(i, j | D)$. Для каждого узла $D(i, j)$, включает одно значение. Значение $D(i, j)$ определяется на вероятностной основе

$$D(i, j) \sim p_D(x | D, \tilde{T}(i, j), i, j). \quad (8)$$

Сила искажающего воздействия узла зависит только от значения концентрации вещества в данной точке. Также сила и направление искажения, в данном узле может зависеть от таких же величин в соседних узлах.

Значение случайного бинарного поля $\eta(x, y)$ можно определить как:

$$\eta(x, y) = \begin{cases} 1, & p(\eta(x, y) = 1) = T(x, y) \\ 0, & p(\eta(x, y) = 0) = 1 - T(x, y) \end{cases} \quad (9)$$

Пример аппликативной помехи на растровом изображении, полученной с помощью разработанного программного модуля представлен на рис. 3.

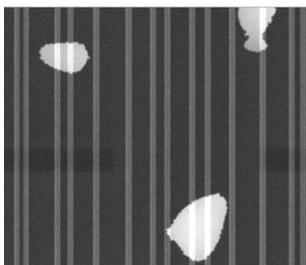


Рис. 3. Пример аппликативной помехи

4. Моделирование локального изменения яркости на изображениях

Часто встречающееся искажение на растровых изображениях - это локальное изменение уровней яркости на случайных областях изображения. Появления таких локальных изменений яркости связано обычно с неправильной настройкой аппаратуры, с помощью которой получали изображение.

Для внесения такого типа искажения в растровые изображения алгоритм использует двумерный градиентный шум (в данном случае шум Перлина G). Выбор именно шума Перлина обусловлен тем, что это один из самых распространенных и простых в плане реализации градиентного шума математических алгоритмов.

На первом этапе работы алгоритма происходит генерация шума Перлина. На следующем этапе, после того, как изображение шума Перлина было сгенерировано происходит его сглаживание.

После получения сглаженного шума Перлина высчитывается средняя яркость шумовой маски, для регулирования яркостью засвета или затемнения:

$$m_j = \frac{1}{wh} \sum_{x=1, y=1}^{w, h} J(x, y), \quad (10)$$

где m_j – средняя яркость шумовой маски; w, h – размерность маски; $J(x, y)$ – сглаженный шум Перлина.

В итоге работы алгоритма мы получаем искаженное изображение с локальными повышением уровней яркости в случайных областях изображения.

$$\tilde{I}(x, y) = I(x, y) + J(x, y) - \alpha m_j. \quad (11)$$

Пример локального изменения яркости на растровом изображении, полученной с помощью разработанного программного модуля представлен на рис. 4.

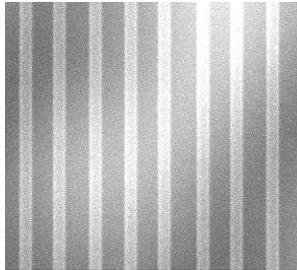


Рис. 4. Пример локального изменения яркости

5. Тестирование разработанного программного модуля

Предварительное тестирование разработанного программного модуля производилось с использованием реальных алгоритмов распознавания изображений и нейронной сети сверточного типа. Была сформирована методика тестирования, по которой: нейронная сеть обучается на изображениях с «искусственными» искажениями с минимальным добавлением изображений с реальными дефектами. На вход нейронной сети подаются прямоугольные фрагменты изображения. На выходе нейронной сети выдается вектор значений, характеризующий принадлежность фрагмента к одному из классов. Далее проводится тестирование алгоритма распознавания на изображениях с реальными дефектами. Пример распознавания реальной аппликативной помехи после обучения нейронной сети на искусственных искажениях, полученных с помощью разработанного программного модуля, представлен на рис. 5.

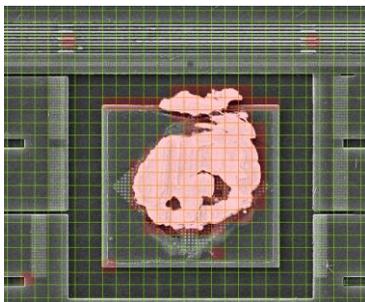


Рис. 5. Пример распознавания аппликативной помехи

Заключение

Данная статья посвящена разработке и реализации программного модуля для аугментации данных на основе внесения искусственных искажений на растровые изображения. Описаны алгоритмы внесения искусственных искажений различного рода и создание самого разрабатываемого программного модуля. В итоге данной разработки получился полностью рабочий программный модуль, реализованный в виде библиотеки и тестового приложения. Использование данного решения позволяет вносить искажения различного рода на растровые изображения, а также использовать искажённые изображения для нагрузочного тестирования и в качестве обучающих выборок, используемых для настройки нейросетевых классификаторов. В дальнейшем предполагается использование данного модуля в системе обработки тематических изображений микроэлектроники

Список литературы

1. Гонсалес Р. Цифровая обработка изображений / Р. Гонсалес, Р. Вудс. – М. : Техносфера, 2005. – 1072 с.
2. Хуанг Т.С. Быстрые алгоритмы в цифровой обработке изображений: преобразования и медианные фильтры / Т.С. Хуанг. – М. : Радио и Связь, 1984. – 224 с.
3. Калинин П.В. Моделирование аппликативных искажений с различной степенью прозрачности и случайной формой / П.В. Калинин, А.А. Сирота // Цифровая обработка сигналов. – 2013. – №1. – С. 28-33.

РЕШЕНИЕ УРАВНЕНИЯ БЛЭКА-ШОУЛЗА С ИСПОЛЬЗОВАНИЕМ РАДИАЛЬНЫХ БАЗИСНЫХ ФУНКЦИЙ

М. В. Пывина

Студент

А. В. Атанов

Доцент

Введение

В настоящее время биржевая торговля приобрела огромные масштабы. При этом если раньше покупались и продавались чаще всего конкретные товары или ценные бумаги, то в последние десятилетия в основном торгуются так называемые производные финансовые инструменты (деривативы). По своей сути, дериватив представляет собой соглашение между двумя сторонами, по которому они принимают на себя обязательство или приобретают право передать определённый актив в установленный срок по согласованной цене. Целью покупки является чаще всего не физическое получение актива, а возможность снизить риск при торговле.

Одним из важнейших деривативов является опцион (см. [1]).

Опцион — производный финансовый инструмент, представляющий собой право купить или продать базовый актив (товары, ценные бумаги, фондовые индексы и т.д.) в будущем по определенной цене.

Типы опционов: опцион колл (опцион на покупку), опцион пут (опцион на продажу).

Виды опционов: европейский, американский и другие.

Одной из важнейших проблем в биржевой торговле является определение цены опциона. Существует несколько моделей, позволяющих вычислять цену опциона; важнейшей из них является модель Блэка-Шоулза (см. [2]), авторы которой были отмечены Нобелевской премией по экономике в 1997 году.

1. Модель Блэка-Шоулза

Модель Блэка-Шоулза определяет теоретическую цену на опционы и основывается на предположении, что если базовый актив торгуется на рынке, то цена опциона на него неявным образом уже устанавливается самим рынком.

Цена опциона $V(S, t)$ в модели Блэка-Шоулза получается из решения уравнения

$$\frac{\partial V(S,t)}{\partial t} + (r-q) \cdot S \cdot \frac{\partial V(S,t)}{\partial S} + \frac{1}{2} S^2 \sigma^2 \cdot \frac{\partial^2 V(S,t)}{\partial S^2} - rV(S,t) = 0. \quad (1)$$

В данной работе рассматриваются только европейские опционы. Для них используются следующие условия:

$$V(S,t)|_{t=T} = \begin{cases} \max(K - S, 0) - \text{"put"} \\ \max(S - K, 0) - \text{"call"} \end{cases} \quad (2)$$

Здесь S – текущая цена базового актива; r – безрисковая процентная ставка; q – дивиденды, ожидаемые в течение срока опциона; σ – волатильность доходности базисного актива; t – время; T – срок действия опциона; K – страйковая цена опциона (цена покупки/продажи по существующему опциону).

Выполняя замену переменной $S = e^y$, уравнение (1) можно переписать в упрощённом виде:

$$\frac{\partial u}{\partial t} + \left(r - q - \frac{\sigma^2}{2} \right) \frac{\partial u}{\partial y} + \frac{\sigma^2}{2} \cdot \frac{\partial^2 u}{\partial y^2} - ru = 0, \quad (3)$$

$u = u(y, t) = V(e^y, t)$. Условия (2) примут вид

$$u(y, t)|_{t=T} = \begin{cases} \max(k - e^y, 0) - \text{"put"} \\ \max(e^y - k, 0) - \text{"call"} \end{cases} \quad (4)$$

2. Схема численного решения уравнения Блэка-Шоулза

Для численного решения задачи (3)-(4) будем использовать метод коллокации Кансы (см. [3]). Решение ищется в виде линейной комбинации

$$u(y, t) = \sum_{j=1}^N \lambda_j(t) \varphi_j(y), \quad (5)$$

где $\varphi_j(y) = \varphi(\|y - y_j\|)$ – радиальная базисная функция (РБФ). Например, в качестве РБФ можно использовать мультиквадратичную функцию Харди

$$\varphi_j(y) = \sqrt{(y - y_j + c)^2}, \quad (6)$$

где c – некоторое выбранное вещественное число.

Подставляя (5) и (6) в (3) и выбирая узлы коллокации: y_1, \dots, y_N , получим следующую схему:

$$\frac{\partial u(y_i, t)}{\partial t} + \frac{\sigma^2}{2} \cdot \frac{\partial^2 u(y_i, t)}{\partial y^2} + \left(r - q - \frac{\sigma^2}{2} \right) \cdot \frac{\partial u(y_i, t)}{\partial y} - ru(y_i, t) = 0. \quad (7)$$

При этом все производные легко считаются аналитически:

$$\begin{aligned}\frac{\partial u(y_i, t)}{\partial t} &= \sum_{j=1}^N \frac{d\lambda_j(t)}{dt} \cdot \varphi(y_i, y_j), \\ \frac{\partial u(y_i, t)}{\partial y} &= \sum_{j=1}^N \lambda_j(t) \cdot \frac{d\varphi(y_i, y_j)}{dy}, \\ \frac{\partial^2 u(y_i, t)}{\partial y^2} &= \sum_{j=1}^N \lambda_j(t) \cdot \frac{d^2\varphi(y_i, y_j)}{dy^2}.\end{aligned}\quad (8)$$

Уравнение (7) можно записать в матричной форме:

$$L\lambda' + \frac{1}{2}\sigma^2 L_{yy} \cdot \lambda + \left(r - q - \frac{\sigma^2}{2}\right) \cdot L_y \cdot \lambda - rL \cdot \lambda = 0. \quad (9)$$

Здесь λ – вектор-столбец неизвестных коэффициентов λ_j ; L, L_y, L_{yy} – матрицы размерностей $N \times N$ с элементами $\varphi(y_i, y_j), \frac{d\varphi(y_i, y_j)}{dy}, \frac{d^2\varphi(y_i, y_j)}{dy^2}$, задаваемыми формулами (6) и (8) соответственно; λ' означает производную λ по t .

Известно, что матрица L из уравнения (9) обратима при несовпадающих узлах коллокации. Поэтому, умножая обе части уравнения на L^{-1} , получим:

$$\lambda' + \frac{1}{2}\sigma^2 \cdot L^{-1}L_{yy} \cdot \lambda + \left(r - q - \frac{\sigma^2}{2}\right) \cdot L^{-1}L_y \cdot \lambda - rI \cdot \lambda = 0. \quad (10)$$

Группируя слагаемые в записи (10), можно прийти к системе дифференциальных уравнений

$$\lambda' = P\lambda,$$

где

$$P = rI - \frac{1}{2}\sigma^2 \cdot L^{-1}L_{yy} - \left(r - q - \frac{\sigma^2}{2}\right) \cdot L^{-1}L_y.$$

Эта система решалась методами первого, второго, четвёртого порядков, а также с помощью неявной схемы.

3. Реализация программы

Рассмотренный алгоритм был программно реализован на объектно-ориентированном языке программирования C#. Использовалась специальная математическая библиотека Math.NET Numerics. Это библиотека с открытым исходным кодом, которая реализует методы и алгоритмы для численных вычислений. Интерфейс программы разделён на две части. Левая часть представляет собой две таблицы (рисунок).

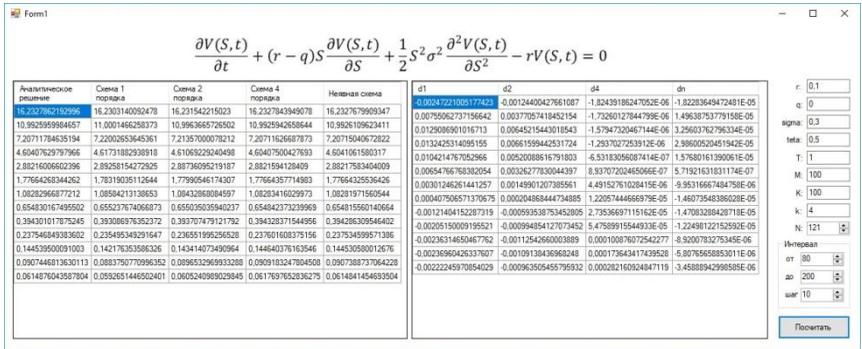


Рисунок. Окно реализованной программы

Первая таблица – таблица значений. Первый столбец – столбец с аналитическим решением, которое было найдено с помощью программы Wolfram Mathematica. Второй столбец показывает стоимость, найденную с помощью схемы первого порядка, третий – с помощью схемы второго порядка, четвёртый – с помощью схемы четвёртого порядка, пятый – с помощью неявной схемы. Вторая таблица показывает разницу между вычислениями, которые получились в ходе реализации программы, и аналитическим решением. В правой части находятся поля с настраиваемыми параметрами, такие как r – безрисковая процентная ставка; q – дивиденды; σ – волатильность доходности базисного актива; T – срок действия опциона; K – страйковская цена опциона и так далее.

Заключение

В тексте подробно описана схема решения уравнения Блэка-Шоулза для оценки стоимости опциона с применением радиальных базисных функций. Реализована программа, способная рассчитывать цену опциона в зависимости от множества параметров. В ходе тестирования программы было установлено, что результаты вычислений даже на схемах первого и второго порядка дают очень хорошие приближения к точному решению.

Отметим, что при незначительной модификации разработанная программа может быть адаптирована и для решения задач оценки

опционов других видов. Кроме того, использование РБФ позволяет перейти от решения рассмотренной модели к её многомерному аналогу, когда переменных S не одна, а множество.

Список литературы

1. Халл, Д. К. Опционы, фьючерсы и другие производные финансовые инструменты / Д.К. Халл. – Вильямс, 2008. – 1051 с.
2. Black, F. The Pricing of Options and Corporate Liabilities / F. Black, M. Scholes // Journal of Political Economy. – 1973. – № 81(3). – P. 637-654.
3. Kansa, E. J. Multiquadric radial basis function approximation methods for the numerical solution of partial differential equations / E. J. Kansa, S. A. Sarra // Advances in Computational Mechanics. – 2009. – № 2. – P. 1-206.

ИССЛЕДОВАНИЕ САМОПОДОБИЯ ТРАФИКА БЕСПРОВОДНЫХ СЕТЕЙ

Е. В. Решетняк

Магистрант

А. С. Коваль

Старший преподаватель

Введение

Одним из важнейших показателей мониторинга сети является ее трафик. Почти до конца XX века считалось, что природа сетевого трафика соответствует Пуассоновскому процессу и описывается классическим Пуассоновским распределением. Однако, со временем количество исследований и измерений характеристик сетевого потока возрастало и стали появляться публикации, например [1], авторы которых исследовали сетевой трафик и обнаруживали, что методы расчета компьютерной сети, основанные на Марковских моделях и формулах Эрланга, с успехом используемые при проектировании телефонных сетей, дают неоправданно оптимистические решения и приводят к недооценке нагрузки. Связано это с тем, что телефонные сети изначально строились по принципу коммутации каналов, а в основе компьютерных сетей, как правило, лежит принцип коммутации пакетов, но методики расчетов остались практически теми же. Трафик в таких сетях имеет явно выраженный всплесковый характер, что повышает вероятность перегрузок в узлах сети, которые ведут к переполнению буферов и вызывают потери и/или задержки. При этом, как оказалось, в компьютерных сетях число

событий на заданном временном интервале зависит от прежних, весьма отдаленных событий. Это означает, что при больших масштабах сети трафик обладает свойством самоподобия, т.е. выглядит качественно одинаково при любых достаточно больших масштабах временной оси.

Однако, несмотря на множество исследований в области самоподобия трафика проводных сетей и его моделирования при помощи так называемого самоподобного процесса [2, 3], вопрос о степени самоподобия трафика беспроводных сетей, где имеют место гораздо большие задержки и потери данных, а также используется другой метод доступа на канальном уровне, остается открытым, а количество исследований в этой области крайне невелико.

В данной статье приводятся вычисления показателя Херста и параметров самоподобия проводного и беспроводного трафика сети факультета компьютерных наук и их сравнительный анализ.

1. Методика эксперимента

Сетевой трафик захватывался на базе факультета компьютерных наук с помощью программы `tcpdump` – это утилита командной строки UNIX, которая дает возможность перехватывать и анализировать трафик, проходящий через компьютер. Для захвата использовалась технология `Monitoring Port`, позволяющая копировать трафик с заданных портов, к которым в данном случае были подключены: файл-серверы, точки доступа беспроводной сети ФКН и порт подключения сети ФКН к внешним сетям (Интернет, корпоративная сеть ВГУ).

Таким образом, были сделаны полные дневные дампы проводного трафика захваченного с файл-серверов (сетевые диски/папки), беспроводного трафика проходящего через точки доступа беспроводной сети и всего внешнего трафика сети.

Прежде всего, может возникнуть необходимость отделить путем фильтрации проводной трафик от беспроводного (или наоборот). Для этого использовалось приложение `Wireshark` — программа-анализатор трафика для компьютерных сетей, которая распространяется под свободной лицензией GNU GPL. Она имеет графический пользовательский интерфейс и большой набор возможностей по сортировке и фильтрации информации. Программа позволяет пользователю разобрать сетевой пакет, отображая значение каждого поля протокола любого уровня. В данном случае фильтрация трафика не потребовалась, однако, проделанная работа была полезна чтобы узнать долю каждого вида трафика во всем трафике сети.

Естественно, считать трафик файлового сервера полностью проводным нельзя, но доля беспроводного трафика оказалась невелика (от 1% и менее), так как пропускные способности клиентов в проводной сети выше, поэтому лишь немногие пользователи подключают сетевые диски и

сетевые папки через Wi-Fi на своих устройствах. Также, по сети может проходить видеотрафик (цифровое ТВ, камеры наблюдения и т.п.) или аудиотрафик (передача голоса, аудио), в зависимости от поставленной задачи может быть необходимо удалить эти составляющие, так как с точки зрения вычисления характеристик самоподобного процесса это белый шум, который искажает результат вычислений.

Для построения временного ряда, с целью дальнейшего анализа на степень самоподобия, нам понадобится такая характеристика, как Delta time. Это время в миллисекундах прошедшее с момента прибытия последнего пакета. По умолчанию данная колонка отключена, добавить ее можно в настройках программы. Средствами Wireshark данные экспортируются в файл с расширением .txt для последующего построения временного ряда и его анализа.

Основным параметром для оценки степени самоподобия является показатель Херста H , который является мерой устойчивости статистического явления или мерой длительности долгосрочной зависимости процесса [4]. Чем ближе значение H к 1, тем выше степень устойчивости долгосрочной зависимости, значение $H = 0,5$, напротив, указывает на отсутствие долгосрочной зависимости. Корреляция между событиями отсутствует. Ряд является случайным, а не фрактальным. При $0,5 < H \leq 1$ временной ряд является трендонеустойчивым (антиперсистентным). Он более изменчив, чем случайный ряд, поскольку состоит из частых реверсов спад-подъем. Тенденция его изменения может быть спрогнозирована. Показатель Херста связан с другими фрактальными параметрами формулами, показанными в таблице [5]. Более подробно про показатель Херста и другие фрактальные параметры, а также характеристики и свойства самоподобных процессов вы можете почитать в работе [6].

Таблица

Фрактальные параметры

Параметр	Расчетная формула
Фрактальная размерность D	$D = 2 - H$
Корреляционный параметр β	$\beta = 2(1 - H)$
Спектральный показатель b	$b = 2H + 1$
Фрактальный показатель α	$\alpha = 3 - 2H$

Для определения значения H в работе используется метод R/S-анализа временных рядов, реализованный в среде Fractan. В статье приводится лишь краткий алгоритм R/S-анализа, более подробно про этот и другие методы анализа временных рядов вы можете узнать в работах [5] и [7].

Fractan – программа для фрактального анализа временных рядов, которая позволяет выполнять вычисление корреляционной размерности, корреляционной энтропии и показателя Херста (рисунок). Фрактальный анализатор Fractan написана Вячеславом Суховым (Институт Математических проблем биологии Российской Академии Наук) и распространяется бесплатно с согласия автора [8].

Вычисление показателя Херста производится по следующему алгоритму [9]:

1. Сначала вычисляются отклонения от среднего значения:

$$X_{t,N} = \sum_{u=1}^t (e_u - M_N), \quad (1)$$

где N – длина периода, меняющаяся от 2 до <длины временного ряда>; t – переменная, меняющая своё значение от 1 до $N-1$; M_N – среднее N элементов; e – конкретный элемент временного ряда.

2. На каждой итерации получаем $N-1$ значений $X_{t,n}$, которые используем в следующей формуле:

$$R = \text{Max}(X_{t,N}) - \text{Min}(X_{t,N}), \quad (2)$$

где R – размах отклонения X .

3. Далее нормируем размах делением на стандартное отклонение S , которое вычисляется по N значениям.

4. Логарифмируем R/S и N и строим на основании полученных данных график.

5. По графику функции $\log(R/S)$ от $\log(N)$ находим наклон путём линейной аппроксимации. Тангенс угла этого наклона и является показателем Херста.

Остальные фрактальные параметры связаны с показателем Херста и могут быть вычислены в Excel (или аналогичной программе) по формулам, приведенным в таблице.

2. Обсуждение результатов

Цель эксперимента в данной статье – вычисление показателя Херста и параметров самоподобия проводного и беспроводного трафика сети и их сравнительный анализ.

Вычислить показатель Херста для всего трафика за один расчёт затруднительно, поскольку сложность расчёта растёт как квадрат количества отсчётов, к тому же, время вычислений затянется, а точность может быть низкой. По этой причине полученные временные ряды были разделены на отрезки по сто тысяч отсчетов в каждом, чтобы облегчить вычисления. Всего было проведено 150 вычислений, по 50 для каждого

вида трафика, результаты были сведены в три таблицы и проанализированы.

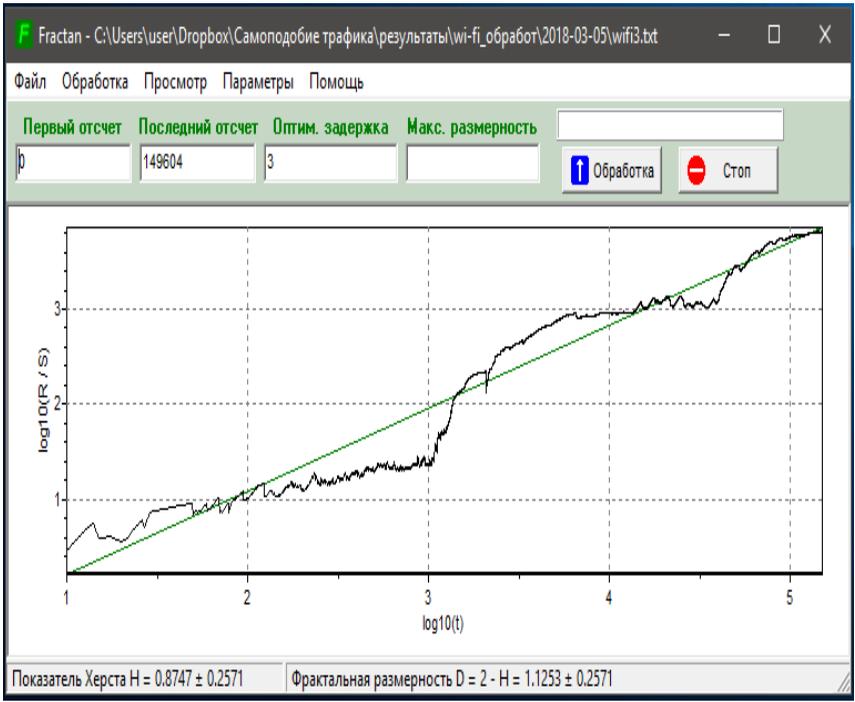


график временной зависимости нормированного размаха в двойном логарифмическом масштабе (кривая) и его линейная аппроксимация (прямая)

Рисунок. Результат вычисления в программе Fractan

В случаях, когда показатель Херста равен 0,5, ряд близок к случайному. Для такого ряда фрактальные параметры процесса: $D = 1,5$, $\alpha = 2$, $\beta = 1$, $b = 2$. Временной ряд стохастичен, а спектральный показатель $b = 2$, такой процесс называют белым шумом. Для проанализированного беспроводного трафика доля таких рядов составила всего 6%, а для проводного – 56%, что значительно больше. Во всем трафике сети самоподобие отсутствует в 48% рядов, что объясняется тем, что он состоит из проводного и беспроводного трафика, доля последнего составляла до 30%, что дало свой вклад в итоговый результат.

Если показатель Херста равен 1, то самоподобие ярко выражено. Для такого ряда фрактальные параметры процесса: $D = 1$, $\alpha = 1$, $\beta = 0$, $b = 3$. Ряд является персистентным (трендоустойчивым) и обладает долговременной памятью. Спектральный показатель $b = 3$, такой процесс называют черным шумом. Для проанализированного беспроводного трафика доля таких рядов составила 78%, для проводного – 16%, а для всего трафика сети – 40%.

Во всех остальных случаях показатель Херста лежит в интервале $0,5 < H < 1$, то есть самоподобие в таких рядах выражено, но в разной степени. Самоподобие тем больше, чем ближе H к 1, потому что чем больше корреляция между процессами, тем более одинаково они себя ведут. Доля таких рядов для беспроводного трафика составила – 16%, для проводного – 28%, а для всего трафика сети – 12%.

Стоит также отметить, что среднее значение показателя Херста для беспроводного трафика гораздо ближе к единице, $H_{\text{сред.}} = 0,932$, в то время как для проводного трафика $H_{\text{сред.}} = 0,661$, а для всего трафика сети $H_{\text{сред.}} = 0,730$.

Все выводы делались с учетом погрешности вычислений.

Заключение

В данной работе было проведено исследование сетевого трафика сервера факультета компьютерных наук ВГУ. Главным направлением исследования являлось выявление свойства самоподобия трафика беспроводной сети. С этой целью была проведена обработка данных проводного, беспроводного и всего трафика сети методом R/S анализа.

Предположение о самоподобной структуре трафика основывается на рассмотрении имеющихся работ других авторов [1-3].

Таким образом, получено подтверждение предположения о самоподобию сетевого трафика в сети ФКН, причем для беспроводного трафика самоподобие свойственно в большей степени. Данный вывод был сделан на основе вычислений показателя Херста и других фрактальных параметров, которые показывают, что в рядах имеются автокорреляции, накопление изменений приводит к появлению долговременной тенденции к возрастанию или убыванию, что говорит о том, что данный трафик по своей структуре является персистентным.

Подобные исследования необходимы для понимания поведения сетевого трафика и разработки новых моделей, учитывающих самоподобие трафика с целью прогнозирования нагрузки и поведения сервера в будущем, оптимального управления ресурсами сервера и многого другого.

В дальнейших работах в этом направлении считаю необходимым проведение подробных исследований беспроводного трафика с применением дополнительных методов для определения параметров

самоподобия. Также, интересным вопросом является изучение причин, по которым самоподобие в беспроводных сетях проявляется в большей степени.

Список литературы

1. Leland W. E. On the self-similar nature of Ethernet traffic (extended version) / M. S. Taqqu, W. Willinger, D. V. Wilson // *IEEE/ACM Transactions of Networking*. – vol. 2. – N 1. – 1994. – P. 1-15.
2. Paxson V. Wide-Area Traffic: The Failure of Poisson Modeling / V. Paxson, S. Floyd // *IEEE/ACM Transactions on Networking*. –1995. – V. 3. – N. 3. – P. 226-244.
3. Willinger W. Self Similarity Through High-Variability: Statistical Analysis of Ethernet LAN Traffic at the Source Level / W. Willinger, M. Taqqu, R. Sherman, D. Wilson // *IEEE/ACM Transactions on Networking*. – February 1997. – Vol. 5. – No. 1. – P. 71-86.
4. Петров В. В. Структура телетрафика и алгоритм обеспечения качества обслуживания при влиянии эффекта самоподобия: автореф. дисс. канд. техн. наук: 05.12.13. – М : 2004. – 20 с.
5. Бельков Д. В. Статистический анализ сетевого трафика / Д. В. Бельков, Е. Н. Едемская, Л. В. Незамова // *Информатика, кибернетика и вычислительная техника*. – 2011. – № 13 (185). – С. 66-75.
6. Федорова М. Л. Об исследовании свойства самоподобия трафика мультисервисной сети / М. Л. Федорова, Т. М. Леденева // *Вестник ВГУ. Серия: системный анализ и информационные технологии*. – 2010. – № 1. – С. 46-54.
7. Зиненко А. В. R/S анализ на фондовом рынке / А.В. Зиненко // *Бизнес-информатика*. – 2012. – № 3 (21). – С. 25-30.
8. Сычев В. В. FRACTAN. Программа для вычисления корреляционной размерности и корреляционной энтропии по временному ряду данных [Электронный ресурс] / В.В Сычев. – Режим доступа : <http://www.iki.rssi.ru/magbase/RESULT/APPENDIX/fractan.boom.ru/soft.htm>
9. Бутаков В. Оценка уровня стохастичности временных рядов произвольного происхождения при помощи показателя Херста / В. Бутаков, А. Граковский // *Computer Modelling and New Technologies*. – 2005. – № 2 (9). – С. 27-32.

ПОВЫШЕНИЕ ЭФФЕКТИВНОСТИ СОТОВЫХ СЕТЕЙ ПОДВИЖНОЙ РАДИОСВЯЗИ НА ПРИМЕРЕ СТАНДАРТА GSM

А. М. Рзуки
Магистрант
Ю. Б. Нечаев
Профессор

Введение

Сотовые сети подвижной радиосвязи (ССПРС) являются наиболее динамичной частью подвижной радиосвязи.

Первоначально сеть подвижной радиосвязи была ориентирована на прием-передачу речевых сообщений и вполне удовлетворяла интересам пользователей и операторов.

Необходимость внедрения новых видов услуг (передача данных, мультимедиа и т. д.) потребовали разработки сетей связи с большей эффективностью.

Главное препятствие – ограниченность частотного ресурса.

Количество фиксированных частот в заданном частотном диапазоне не может увеличиваться бесконечно. Причина – взаимные помехи при использовании рабочих каналов с близкими частотами.

Цель работы – повышение эффективности действующих сотовых сетей связи за счет поиска и разработки новых схемотехнических решений при построении топологических структур и оптимального использования заданного частотного спектра.

Для достижения поставленной цели необходимо решить несколько задач, в частности:

- определить основные показатели эффективности SSPRC;
- исследовать новые топологии сетей связи;
- провести сравнительную оценку показателей эффективности действующих сетей связи, использующих типовую шестиугольную структуру соты, и предлагаемой в качестве альтернативы пятиугольной на примере стандарта GSM.

Основными техническими параметрами SSPRC как единого целого являются емкость сети и зона покрытия. Чем больше емкость и зона покрытия сети, тем выше возможности сети как системы массового обслуживания. Но при оценке эффективности надо учитывать также,

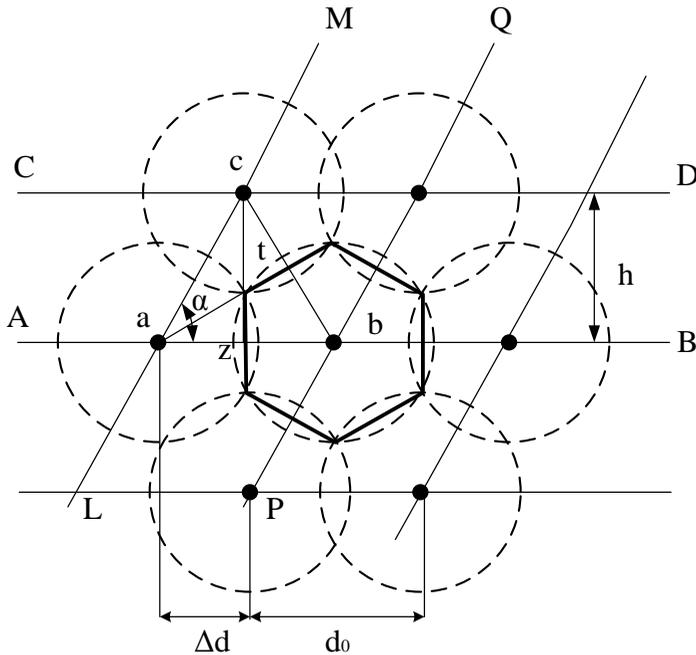
сколько базовых станций используется для получения такой емкости и зоны.

1. Топология и принцип действия сотовых сетей радиосвязи

На рис. 1 показана шестиугольная структура соты, используемая в действующих сетях [1].

Основными параметрами модели сети являются: расстояние между узлами (соседними базовыми станциями (БС)) $d_0 = r\sqrt{3}$; расстояние между параллельными прямыми $h = 1,5r$; смещение узлов, расположенных на соседних параллельных линиях, относительно друг друга $\Delta d = d_0/2 = r\sqrt{3}/2$; расстояние от БС до границы соты $r = \Delta d$.

В однородной сети все БС имеют: одинаковые значения эквивалентной изотропной излучаемой мощности; антенны с круговой диаграммой направленности и одинаковыми высотами; энергетические параметры всех абонентских станций также одинаковы.



$$at = r; ac = cb = ab = r \sin \alpha = r\sqrt{3}; cz = ct + tz = r + r \sin \alpha = 1,5r.$$

Рис. 1. Сеть с регулярной сотовой структурой

Основная идея сотовой связи – принцип повторного использования частот в несмежных сотах. Группа сот с различными наборами частот представляет собой кластер. Такое построение определяет высокую эффективность использования частотного спектра в сотовых сетях связи. Принцип повторного использования частот снижает дефицит радиоканалов, но приводит к относительному избытку базовых станций.

Управление излучением частотных каналов между зонами осуществляется в районе обслуживания и обеспечивает переход подвижного объекта из одной зоны в другую без прерывания радиообмена. Однако это приводит к перекрытию зон и вызывает нежелательную межканальную интерференцию, увеличивает требуемую мощность передатчика [2, 3].

В работе рассмотрено АС РФ №16264312 А1 «Способ радиосвязи с подвижными объектами в системах связи сотовой структуры», принятое за прототип [4].

На рис. 2 приведена структура расположения БС и зона связи. БС расположены в вершинах треугольников.

Они управляют мощностью передатчиков при условии, что радиообмен не прерывается при переходе подвижного объекта из одной зона в другую.

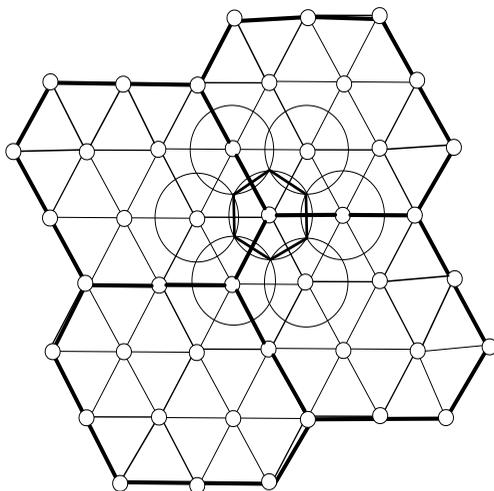


Рис. 2. Структура расположения базовых станций

Для обеспечения эффективности радиосвязи за счет расширения зоны обслуживания при сохранении числа БС может быть использовано круговое сканирование с помощью ФАР на БС.

Недостатки:

- использование ФАР требует запоминание азимутального угла;
- необходимо равномерное распределение подвижных объектов по азимутальному углу диаграммы направленности.

В итоге:

- возрастают межканальные помехи;
- расходуется дополнительная мощность передатчика.

Из рис. 2 видно, что в сотовой системе-прототипе зона обслуживания полностью покрыта сотами, образуя правильные треугольники. Толстыми линиями показаны равные шестиугольники, полностью заполняющие плоскость. Заполнение плоскости равными фигурами возможно, если эти фигуры параллельны друг другу, и представляют собой многоугольники [5]. На рис. 3 толстыми линиями показана совокупность равных шестиугольников, образованных поворотом на 90° в горизонтальной плоскости.

При пересечении шестиугольников получается структура из пятиугольников. Она и взята в статье за основу построения пятиугольной топологии ССПРС [4].

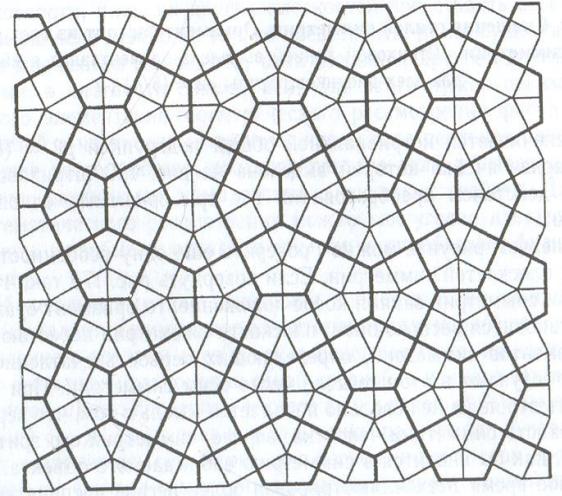


Рис. 3. Система равных шестиугольников

Предлагается располагать базовые станции в вершинах равносторонних пятиугольников с двумя несмежными прямыми углами. Пятиугольники покрывают всю обслуживаемую территорию без разрывов (рис. 4).

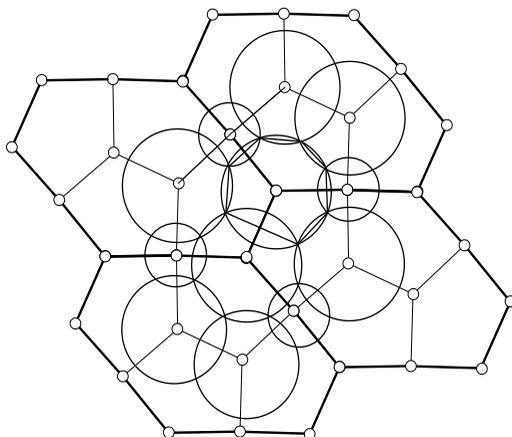


Рис. 4. Предлагаемая структура расположения базовых станций

Станции с меньшим радиусом располагаются в вершинах квадратов, образуемых точками пересечений взаимно перпендикулярных прямых, проходящих через три последовательно расположенных базовых станции. При этом обеспечивается минимальное взаимное перекрытие. Данный способ построения сети основывается на решении задачи оптимального размещения БС с равномерной (круговой) зоной обслуживания [6]. На рис. 5 показаны зоны обслуживания базовых станций, на рис. 6 – зоны перекрытия территории обслуживания базовых станций.

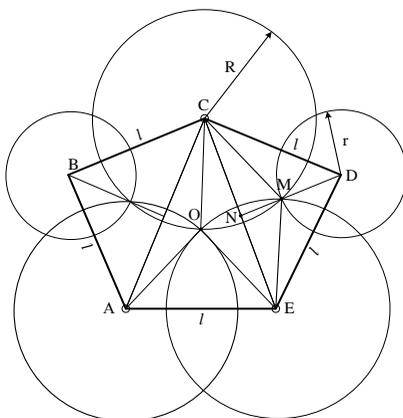


Рис. 5. Зоны обслуживания базовых станций

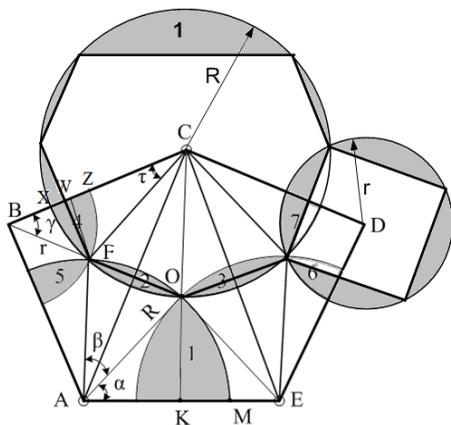


Рис. 6. Зоны перекрытия территорий обслуживания БС

Проведена оценка зон радиопокрытия базовых станций пятиугольной топологии сети, на основе которой рассчитаны геометрические показатели топологии сети и приведены в табл. 1.

Таблица 1

Геометрические показатели ТС подвижной радиосвязи

Наименование показателя	Ед. Изм.	Структура топологии сети		Отношение показателей
		Пятиугольная	Шестиугольная	
Радиус зоны обслуживания БС	м	0,756 l 0,44 l	0,577 l	1,3 0,76
Площадь зоны обслуживания БС	м ²	1,8 l^2 0,61 l^2	1,05 l^2	1,7 0,58
Площадь кластера	м ²	1,66 l^2	2,6 l^2	0,64
Площадь зоны интерференционных помех	м ²	17,58 l^2	18,2 l^2	0,97
Перекрытие зон обслуживания БС в кластере	%	26	21	1,24
Относительная величина зоны взаимного перекрытия в соте	%	21 36	17	1,24 2,1

Наименование показателя	Ед. Изм.	Структура топологии сети		Отношение показателей
		Пятиугольная	Шестиугольная	
Защитный интервал	м	2,57 <i>l</i>	2,65 <i>l</i>	0,97
Коэффициент интерференционных помех	–	3,4 5,8	4,59	0,74 1,26
Количество БС в кластере	–	5	7	0,71

На основе рассчитанных значений площади покрытия территории шестиугольной типовой топологии и предлагаемой пятиугольной топологии, получены их основные технические характеристики (табл. 2).

2. Частотно-территориальное планирование предлагаемой сотовой сети подвижной радиосвязи кластера 5

Известны алгоритмы распределения частот в сотовых сетях связи, основанные на теории раскраски карт. В статье использован алгоритм ЧТП на основе теории «латинский квадрат».

Для обеспечения ЭМС БС определение частотного ресурса пятиугольной топологии сети проведено по регулярной сетке в соответствии с правилами «латинский квадрат» кластера 5 для частотных групп. При этом обеспечивается разнесение одинаковых частотных групп на расстояние L .

В соответствии с полученными данными производится присвоение частотных групп базовым станциям [7]. На рис. 7 представлен частотно-территориальный план пятиугольной топологии сети.

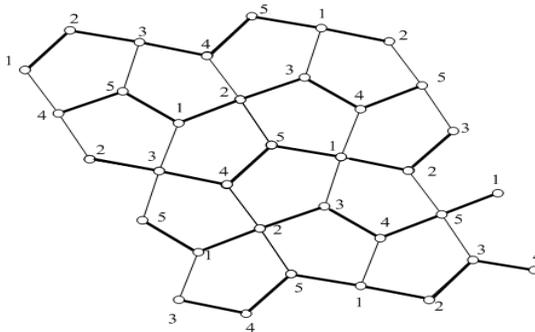


Рис. 7. Частотно-территориальный план пятиугольной ТС

Таблица 2

Основные технические характеристики ТС

Наименование показателя	Ед. величины показателя	Значение показателя ТС		Отношение показателей
		пятиугольная	шестиугольная	
Площадь радиопокрытия фрагментом	км ²	26,6 l ²	27,72 l ²	0,96
Число БС для радиопокрытия фрагмента	–	36	44	1,22
Средняя площадь радиопокрытия одной БС	км ²	0,74 l ²	0,63 l ²	1,18
Число соединительных линий связи между БС фрагмента	–	35	43	1,23

Согласно полученным данным на рис. 8 представлен частотно-территориальный план сотовой сети кластера 5.

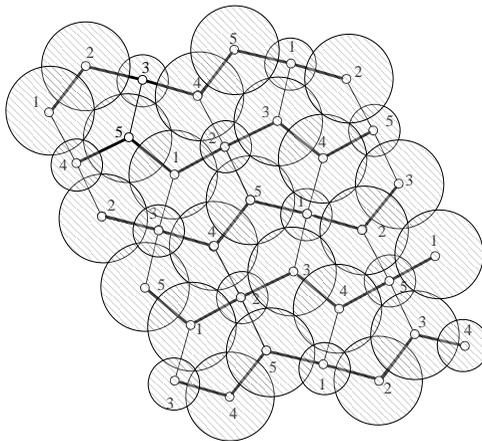


Рис. 8. Частотно-территориальный план предлагаемой топологии сотовой сети кластера 5

Заключение

Топология сети на основе равностороннего пятиугольника имеет улучшенные показатели по сравнению с типовой сотовой структурой (кластер 7). В частности, число базовых станций может быть сокращено на 22% при условии покрытия одинаковой территории; коэффициент повторного использования частотных каналов для территорий равной площади увеличивается на 14%, а кластер сети сокращен по размерам на 40%. При этом незначительно (на 3%) уменьшается коэффициент защитного интервала внутриканальных помех.

Таким образом, пятикластерную топологию сотовой структуры подвижной радиосвязи можно рекомендовать при проектировании перспективных сотовых сетей связи.

Список литературы

1. Солодовник В.Ф. Системы подвижной радиосвязи: учебное пособие / В. Ф. Солодовкин, А. В. Воробьев. – Харьков : ХАИ (Национальный аэрокосмический университет им. Н.Е. Жуковского), 2014. – 142 с.
2. Рекомендации МСЭ-R [Электронный ресурс]. – Режим доступа : <https://www.itu.int/pub/R-REC/ru>
3. Архитектура сети с высокоскоростной коммутацией // Патент США №5418779 по классу H 04 L 12 /48.
4. Способ радиосвязи с подвижными объектами в системе связи сотовой структуры // Авторское свидетельство СССР №1626412 А1 по классу H04В7/27.
5. Шубников А.В. Симметрия в науке и искусстве / А.В. Шубников. – М. : Институт компьютерных исследований, 2004. – 560 с.
6. Беленький В.Г. Расчет зоны покрытия базовых станций в системах связи с подвижными объектами. Методические указания / В.Г. Беленький. – Новосибирск : СибГУТИ, 2003. – 61 с.
7. Бабков В. Ю. Сети мобильной связи. Частотно-территориальное планирование / В.Ю. Бабков, М.А. Вознюк, Г.А. Михайлов. – М. : Горячая линия-Телеком, 2007. – 224 с.

АНАЛИЗ ИТ-РИСКОВ С ИСПОЛЬЗОВАНИЕМ ТЕОРИИ БАЙЕСА

О. А. Рощупкина

Студент

С. В. Власов

Доцент

Введение

Многие риски связаны с разработкой программного обеспечения, и управление рисками стало одним из ключевых направлений в разработке программного обеспечения. Ошибки программного проекта часто являются результатом недостаточного и неэффективного управления рисками [1]. Чтобы получить надлежащий и эффективный контроль рисков, планирование рисков должно выполняться на основе причинности риска, которая может предоставить больше информации о рисках для принятия решений.

Цель настоящей работы является исследование возможностей применения байесовских сетей (БС) для определения риска при разработке ИТ проекта.

1. Неопределенность и вероятность

Несмотря на постоянное совершенствование методик управления проектом неопределенность остаётся неотъемлемой частью характеристик проекта, обуславливающих его успех. Неопределенность в программном обеспечении обусловлена неопределенностью при принятии решений. Любая неопределенность порождает риск, требующий анализа и соответствующей оценки и реакции. Существует несколько определений понятия «риск»:

$$\text{РИСК} = \frac{\text{ОПАСНОСТЬ}}{\text{ЗАЩИТА}},$$

$$\text{РИСК} = \text{ВЕРОЯТНОСТЬ} \cdot \text{ПОСЛЕДСТВИЯ},$$

$$\text{РИСК} = \text{НЕОПРЕДЕЛЕННОСТЬ} \cdot \text{ПОТЕРИ}.$$

2. Качественная оценка риска

Качественная оценка рисков позволяет идентифицировать виды рисков, определить и описать причины и факторы, которые влияют на уровень этого риска. Также описывается оценка возможных последствий реализации этого риска и мероприятия по минимизации их. Одна из самых простых методик оценки заключается в выборе двух параметров для

любого из рисков: вероятность возникновения риска и его степень влияния [2].

Вероятность возникновения риска – это возможность того, что вследствие риска действительно наступит. Степень влияния риска – мера серьезности последствий, уровень потерь, связанных с риском. Она может оцениваться в денежных единицах.

Для каждого из рисков в реестре должен быть определен приоритет, который упорядочивает риски по важности и необходимости детального анализа. PMBOK определяет качественный анализ рисков как процесс приоритизации индивидуальных рисков проекта через оценку вероятности появления риска и его возможного влияния на характеристики проекта.

Для оценки и ранжирования риска используют матрицу 5x5. Необходимо определить RR (risk ranking), формула будет иметь вид:

$$RR = \text{Вероятность} \cdot \text{Последствия} . \quad (1)$$

3. Количественная оценка риска

PMBOK определяет количественный анализ рисков как процесс численного анализа обобщенного влияния идентифицированных рисков проекта и других источников неопределенностей на результаты проекта [3].

Байесовские сети имеют то преимущество, что могут исследовать влияние множества факторов, в то же время признавая связанные с ними неопределенности. Также они обладают необходимой гибкостью [4].

Байесовская сеть обычно состоит из направленного ациклического графа (DAG), который представляет собой структуру сети и связанные с ней набор таблиц условных вероятностей, которые являются сетевыми параметрами.

Этот инструмент основывается на теореме Байеса теории вероятностей для определения апостериорной вероятности события при условии, что имеет место событие по их априорным вероятностям. В общем случае формула Байеса имеет следующий вид для группы несовместных событий A_1, A_2, \dots, A_n при условии, что произошло событие B , где $P(A_i/B)$ – апостериорная вероятность состояния A_i после того, как стал известен набор признаков B , $P(A_i)$ – априорная вероятность состояния объекта A_i , а $P(B/A_i)$ – вероятность появления набора признаков B при состоянии объекта A_i :

$$P(A_i/B) = \frac{P(A_i)P(B/A_i)}{\sum_{i=1}^n P(A_i)P(B/A_i)} \quad (2)$$

Формула Байеса (2) позволяет пересчитывать априорные вероятности гипотез после получения сведений о том, что событие произошло. В

результате чего корректируются вероятности гипотез при получении новой информации.

В [5] определены 27 программных рисков, классифицированных в 6 измерений. Эта структура обобщает предыдущие исследования, подходит для широкого спектра программных проектов и для темы этого исследования.

В результате строим реестр рисков проекта, часть таблицы приведена (Табл. 1).

Таблица 1

Часть реестра рисков

Название риска	Обозначение	Содержание	Последствия
Риск изменения управления	Орг1	Изменение организационного управления во время проекта	Отставание по срокам
Риск консервативных пользователей	Пол1	Пользователи, устойчивые к изменениям	Увеличение сроков, снижение качества
Конфликт между пользователями	Пол2	Пользователи, не могут сформировать общие требования	Увеличение сроков, Закрытие проекта
Риск негативных пользователей.	Пол3	Пользователи с негативным отношением к проекту	Закрытие проекта
Безучастие пользователей	Пол4	Пользователи совсем не участвуют в проекте	Закрытие проекта
Недостаточное участие пользователей	Пол5	Недостаток сотрудничества со стороны пользователей	Увеличение сроков, снижение качества, закрытие проекта
Постоянно изменяющаяся система	Тр1	Пользователи постоянно изменяют требования	Увеличение сроков, снижение качества, увеличение стоимости

Чтобы описать Байесовскую сеть количественно, необходимо составить таблицы условных вероятностей (ТУВ). В качестве априорных и условных вероятностей наступления рисков используем экспертные данные. Рассмотрим три взаимосвязанных риска: «Риск изменения управления» (Орг1), «Недостаточное участие пользователей» (Пол5) и «Плохое планирование» (Пл4). Априорные вероятности представлены в таблице (Табл. 2).

Таблица 2

Априорные вероятности Орг1, Пол5 и Пл4

Риск	Априорная вероятность наступления риска	Априорная вероятность не реализации риска
Риск изменения управления (Орг1)	0,1	0,9
Недостаточное участие пользователей (Пол5)	0,2	0,8
Плохое планирование (Пл4)	0,2	0,8

Условные вероятности (Табл. 3):

Таблица 3

Условные вероятности для Орг1, Пол5 и Пл4

	Орг1 = да		Орг1 = нет	
	Пол5 = да	Пол5 = нет	Пол5 = да	Пол5 = нет
Пл4 = да	0,7	0,5	0,6	0,1
Пл4 = нет	0,3	0,5	0,4	0,9

Слово «да» означает наступление риска, а слово «нет» – его нереализацию.

В таблице представлены следующие вероятности:

$$P(\text{Пл4}/\text{Орг1}\overline{\text{Пол5}}), P(\text{Пл4}/\text{Орг1}\text{Пол5}), P(\text{Пл4}/\overline{\text{Орг1}}\overline{\text{Пол5}}), \\ P(\text{Пл4}/\overline{\text{Орг1}}\text{Пол5}), P(\overline{\text{Пл4}}/\text{Орг1}\overline{\text{Пол5}}), P(\overline{\text{Пл4}}/\text{Орг1}\text{Пол5}), \\ P(\overline{\text{Пл4}}/\overline{\text{Орг1}}\overline{\text{Пол5}}), P(\overline{\text{Пл4}}/\overline{\text{Орг1}}\text{Пол5})$$

Рассчитываем полную вероятность риска Пл4 по формуле (3):

$$P(A) = \sum_{i=1} P(H_i) \cdot (A/H_i) \quad (3)$$

$$\begin{aligned}
P(\text{Пл4}) &= P(\text{Пл4}/\text{Opz1Пол5}) \cdot P(\text{Opz1}) \cdot P(\text{Пол5}) + \\
&P(\text{Пл4}/\overline{\text{Opz1Пол5}}) \cdot P(\overline{\text{Opz1}}) \cdot P(\text{Пол5}) + \\
&P(\text{Пл4}/\text{Opz1}\overline{\text{Пол5}}) \cdot P(\text{Opz1}) \cdot P(\overline{\text{Пол5}}) + \\
P(\text{Пл4}/\overline{\text{Opz1}}\overline{\text{Пол5}}) \cdot P(\overline{\text{Opz1}}) \cdot P(\overline{\text{Пол5}}) &= 0,7 \cdot 0,1 \cdot 0,2 + \\
&0,1 \cdot 0,9 \cdot 0,8 + 0,5 \cdot 0,1 \cdot 0,8 + 0,6 \cdot 0,9 \cdot 0,2 = 0,234
\end{aligned}$$

Нами был разобран достаточно простой пример для влияния на риск двух факторов. Однако не учитывается влияние еще одного фактора: «Неэффективная коммуникация» (Пл7) на «Плохое планирование» Пл4. То есть, на Пл4 оказывает влияние четыре переменные, произвести расчеты в ручную и учесть все существующие взаимосвязи значительно сложнее. Процесс расчета будет более продолжительным и трудозатратным.

Построим байесовскую сеть (рис. 1), где узлами являются определенные нами важные риски. Проектирование сети происходит в средстве AgenaRisk [6].

После того, как все необходимые экспертные данные были внесены, а направления соотнесены, запускаем вычисление и получаем результат (Рис. 2).

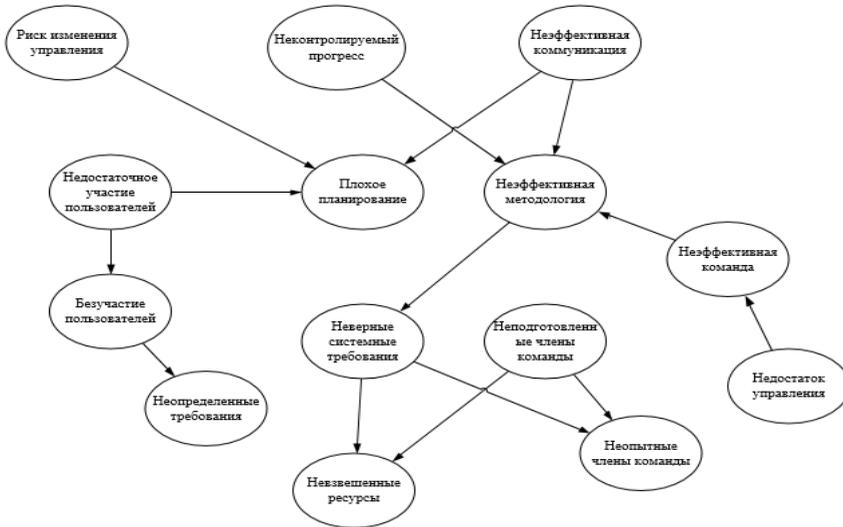


Рис. 1. Байесовская сеть

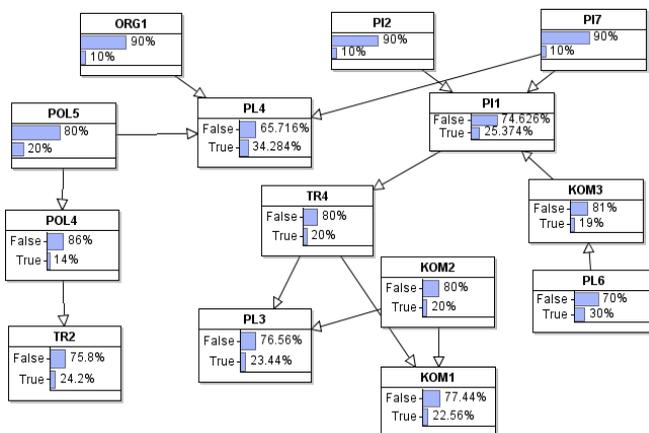


Рис. 2. Результирующая байесовская сеть

Таким образом, можно составить реестр рисков, оцененных количественно (Табл. 4).

Таблица 4

Отсортированная количественная оценка рисков

Риск	Апостериорная вероятность
PL4 Плохое планирование	0,34284
PL6 Недостаток управления	0,3
PL1 Неэффективная методология	0,25374
TR2 Неопределенные требования	0,242
PL3 Невзвешенные ресурсы	0,2344
KOM1 Неопытные члены команда	0,2256
POL5 Недостаточное участи пол-лей	0,2
TR4 Неверные системные требования	0,2
KOM2 Неподготовленные члены команды	0,2
KOM3 Неэффективная команда	0,19
POL4 Безучастие пользователей	0,14
ORG1 Риск изменения управления	0,1
PL2 Неконтролируемый прогресс	0,1
PL7 Неэффективная коммуникация	0,1

Таким образом, можно сформулировать основной вопрос: какова вероятность того, что эффективность проекта (сроки, качество, стоимость) снизится при возникновении одного из факторов риска? Глядя на Таблицу 5 можно получить ответ на этот вопрос.

Заключение

Для более эффективного анализа рисков и их планирования важно выявить причинность между факторами риска и результатами проекта в управлении рисками, то есть построить байесовскую сеть. Основным результатом нашего исследования является проектирование байесовской сети причинности для анализа рисков проекта, использование которой поможет повысить эффективность проекта.

Подводя итог исследованию, касающемуся теории Байеса в анализе ИТ-рисков, нами сделан вывод о том, что предлагаемая модель байесовской сети причинности позволяет повысить эффективность контроля факторов риска. Таким образом, метод Байеса в анализе ИТ-рисков дает надежный результат.

Список литературы

1. Авдошин, С.М. Информатизация бизнеса. Управление рисками / С.М. Авдошин, Е.Ю. Песоцкая. – М. : ДМК Пресс, 2011. – 176 с.
2. Cai R. BASSUM: a Bayesian semi-supervised method for classification feature selection [Электронный ресурс] / R. Cai, Z. Zhang, Z. Hao. – Режим доступа : <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.187.3147&rep=rep1&type=pdf>
3. Массель Л.В. Применение байесовских сетей доверия для интеллектуальной поддержки исследований проблем энергетической безопасности / Л.В. Массель, Е.В. Пяткова // Вестник ИрГТУ. – 2012. – № 2. – С.8-13
4. Зинкевич В. Информационные риски: анализ и количественная оценка / В. Зинкевич, Д. Штатов // Бухгалтерия и банки. – 2007. – № 1. – С. 50-55
5. Cheng J. Learning Bayesian networks from data: an information-theory based approach / J. Cheng, R. Greiner, J. Kelly, D. Bell, W. Liu // Artificial Intelligence. – 2002. – 137 (1-2). – P. 43-90.
6. Bayesian Network Software for Risk Analysis and Decision Making [Электронный ресурс]. – Режим доступа : <http://www.agenarisk.com/>

АНАЛИЗ И ВИЗУАЛИЗАЦИЯ АЛГОРИТМА КВАНТОВОГО РАСПРЕДЕЛЕНИЯ КЛЮЧЕЙ BB84

Е. А. Рыбалкин

Студент

А. Ф. Клиньских

Профессор

Введение

Процедура, известная как квантовая криптография, или квантовое распределение ключей, использует принципы квантовой механики, чтобы сделать безопасной передачу секретной информации. Квантовая криптография является одним из замечательных применений квантовой теории информации.

1. Квантовое распределение ключей

Квантовое распределение ключей (КРК) является протоколом, который гарантированно надежен, и посредством которого биты закрытого ключа могут быть созданы в процессе коммуникации двух сторон по открытому каналу. Эти биты ключа могут быть использованы для реализации классической криптосистемы с закрытым ключом, что обеспечивает безопасную связь.

Единственное требование для протокола КРК состоит в том, что при передаче кубитов по открытому каналу частота появления ошибок должна быть меньше определенного порога. Безопасность получающегося в результате ключа гарантируется свойствами квантовой информации и, следовательно, обусловлена только фундаментальными законами физики.

Квантовое распределение ключей основано на том, что Ева («шпион») не может извлечь никакой информации из кубитов, передаваемых от Алисы к Бобу, не нарушив их состояние. Во-первых, согласно теореме о невозможности копирования [1] Ева не может копировать кубиты Алисы. Во-вторых, при любой попытке различить два неортогональных квантовых состояния извлечение информации в полном объеме невозможно [1-3].

2. Алгоритм квантового распределения ключей BB84

Рассмотрим алгоритм квантового распределения ключей BB84, он состоит из следующих этапов:

1. Алиса выбирает $4n$ случайных битов данных; $n > 1$.
2. Алиса выбирает случайную $4n$ – битовую строку a и кодирует каждый бит данных как $\{|0\rangle, |1\rangle\}$, если соответствующий бит a равен 0, или $\{|+\rangle, |-\rangle\}$, если бит равен 1.
3. Алиса посылает Бобу полученное состояние.
4. Боб получает $4n$ кубит, объявляет об этом событии и измеряет каждый кубит в базисе X или Z случайным образом.
5. Алиса объявляет базисы.
6. Алиса и Боб отбрасывают все биты, которые Боб измерил в базисе, отличном от того, в котором их приготовила Алиса. С большой вероятностью остается, по меньшей мере, $2n$ бит.
7. Алиса выбирает n бит, чтобы использовать их для проверки вмешательства Евы, и сообщает Бобу, какие биты выбраны.
8. Алиса и Боб объявляют и сравнивают значения n контрольных бит.
9. Алиса и Боб выполняют согласование информации и усиление конфиденциальности по оставшимся n битам для получения p совместных битов ключа.

3. Анализ и визуализация алгоритма

В соответствии с представленным выше алгоритмом была разработана блок-схема. Можно выделить три основных этапа (рис. 1).

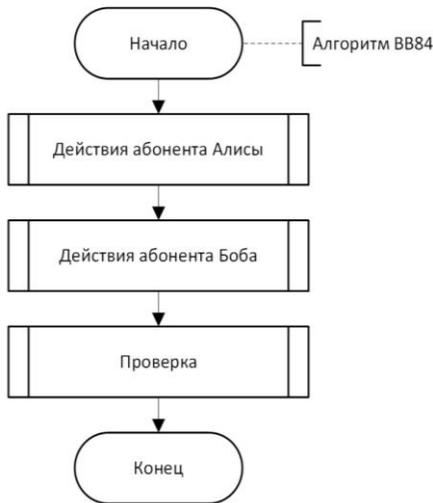


Рис. 1. Блок-схема алгоритма BB84

На первом этапе приведены действия абонента Алисы (рис. 2).

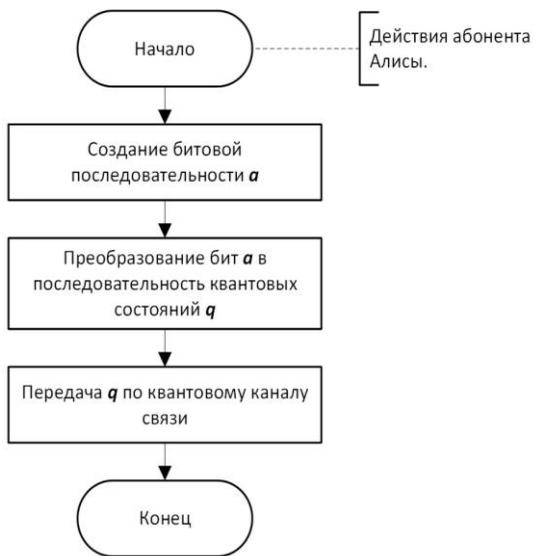


Рис. 2. Действия абонента Алисы

На втором этапе приведены действия абонента Боба (рис. 3).

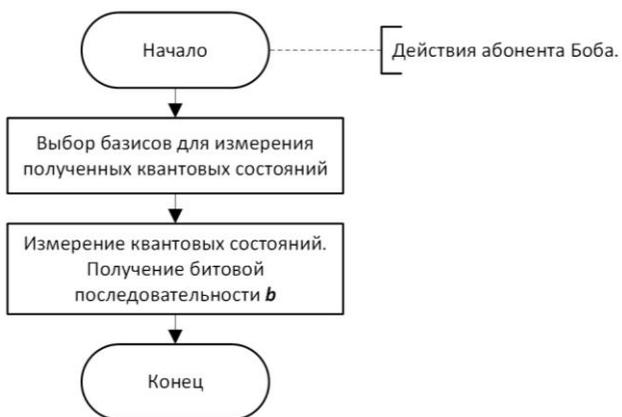


Рис. 3. Действия абонента Боба

На завершающем этапе рассмотрен процесс проверки (рис. 4).

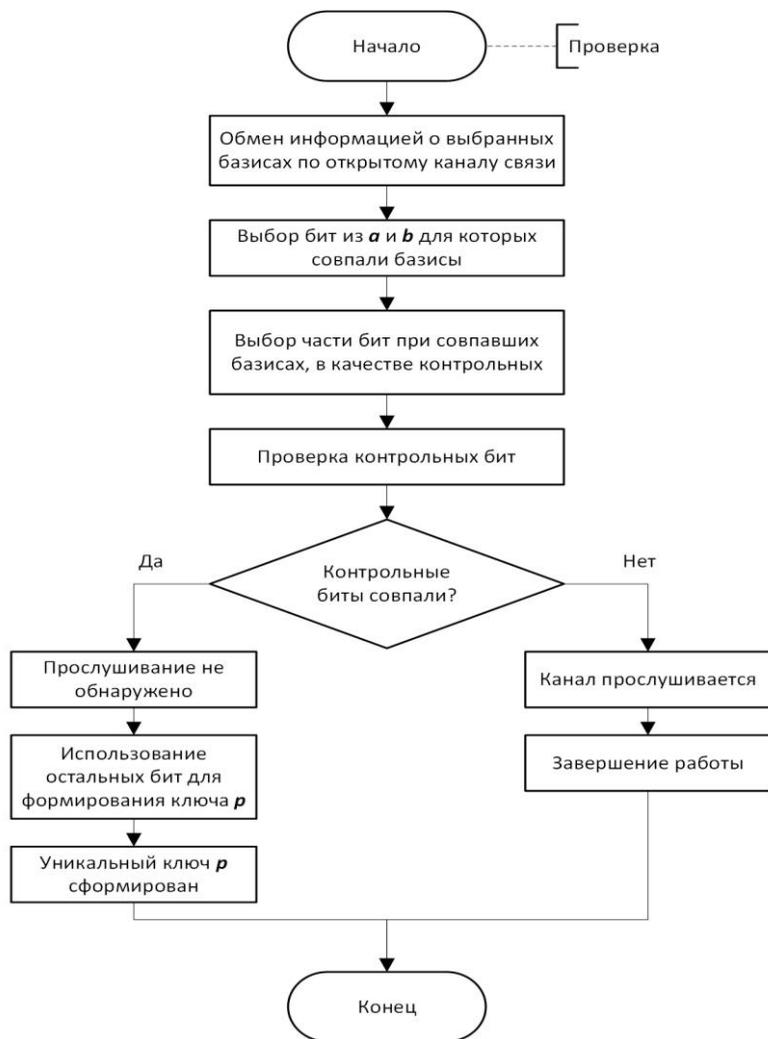


Рис. 4. Проверка

После анализа алгоритма была реализована программа по блок-схеме для визуализации алгоритма квантового распределения ключей. Для реализации программы в качестве языка программирования был выбран C#, а средой разработки Microsoft Visual Studio 2015.

Результат выполнения программы показан на рис. 5.

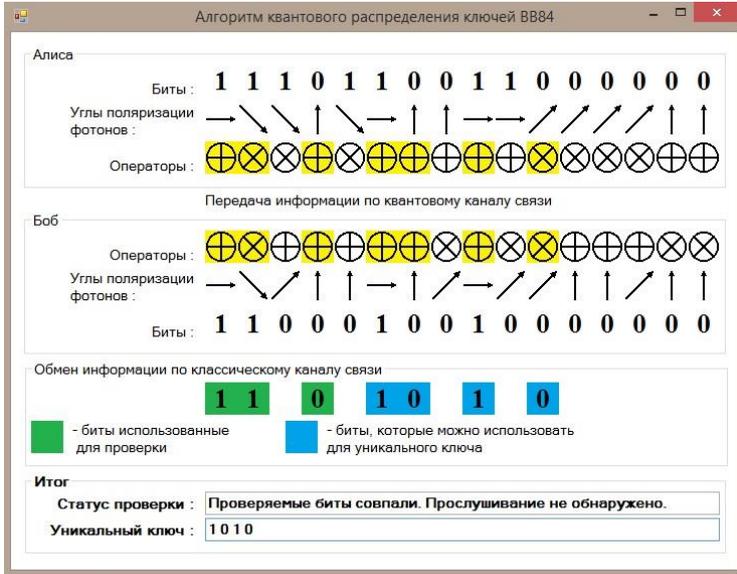


Рис. 5. Результат работы программы

Видно, что программа фиксирует факт прослушивания и секретный ключ. Ниже на рис. 6 представлено процентное содержание ключей с заданной длиной по формируемой выборке.

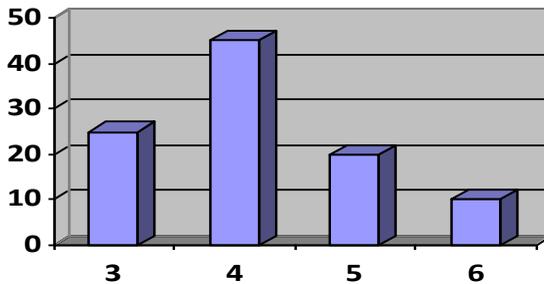


Рис. 6. Гистограмма процентного содержания ключей

Заключение

Данная статья посвящена алгоритму квантового распределения ключей. Описан алгоритм квантового распределения ключей, построена блок-схема алгоритма и реализована программа для визуализации алгоритма.

Список литературы

1. Нильсен М. Квантовые вычисления и квантовая информация / М. Нильсен, И.Чанг; пер. с англ. – М. : Мир, 2006. –824 с.
2. Запрягаев С.А. Введение в квантовые информационные системы: учебное пособие / С. А. Запрягаев. – Воронеж : Издательский дом ВГУ, 2015. – 219 с.
3. Скалли М.О. Квантовая оптика / М.О. Скалли, М.С. Зубайри; пер с англ.; под ред. В.В. Самарцева. – М. : Физматлит, 2003. – 512 с.

ОБ ОДНОМ ПРИЛОЖЕНИИ ФУНКЦИИ МЕБИУСА

Т. М. Сивоплясов

Студент

Е. В. Вахитова

Профессор

Введение

В работе рассмотрено приложение функции Мебиуса к доказательству частного случая теоремы Дирихле об арифметической прогрессии.

Теорема 1 (Дирихле). *Арифметическая прогрессия вида $a + dk$:*

$$a, a + d, a + 2d, \dots,$$

при $a, d, k \in \mathbf{N}$ и $(a, d) = 1$ содержит бесконечное множество простых чисел.

Отметим, что требование $(a, d) = 1$ важно, так как если $(a, d) > 1$, то арифметическая прогрессия будет содержать самое большее одно простое число, то есть или одно, или ни одного простого числа.

Вспомогательные сведения и сведения о Л. Дирихле можно найти в книгах [1-3].

Доказательство теоремы 1 Л. Дирихле получил в 1837 г. Оно основано на применении характеров и L-рядов Дирихле с использованием теории функций комплексной переменной. Доказательство теоремы 1 или более сильного результата можно найти в книгах [4-6]. О характерах и их свойствах можно узнать из книги [7].

Некоторые частные случаи теоремы 1 могут быть элементарно доказаны, например, для арифметических прогрессий видов $4k + 1$, $4k + 3$, $6k + 1$, $6k + 5$ ($k \in \mathbf{N}$). Эти доказательства приведены в книге [1].

Элементарное доказательство в общем случае требует сравнительно больших усилий. В 1949 г. А. Сельберг получил элементарное доказательство теоремы 1, то есть без использования теории функций комплексной переменной. Это доказательство является технически сложным, использует аппроксимацию числовой функции. Элементарное доказательство теоремы 1 приведено в книге [8].

В 1961 г. А. Роткевич получил элементарное доказательство для частного случая теоремы Дирихле об арифметической прогрессии, а именно, для арифметической прогрессии вида $nk+1$, где $n, k \in \mathbf{N}$. Доказательство, полученное А. Роткевичем, приведено в статье [9], в книгах его нет. Идея доказательства состоит в том, что достаточно было доказать существование простого числа вида $nk+1$ для любого натурального числа n , то есть простого числа вида $nmt+1$, которое больше m и имеет вид $nk+1$ при $n=mt$, где $n, m, k, t \in \mathbf{N}$. При этом использованы понятия: сравнение по натуральному модулю, показатель числа по простому модулю и его свойства, функция Мебиуса и ее свойства и теорема Ферма.

В настоящей работе приведено приложение функции Мебиуса к доказательству частного случая теоремы Дирихле об арифметической прогрессии, изучена функция Мебиуса и построен график функции Мебиуса. Выполнен перевод статьи А. Роткевича с французского языка. Идея доказательства частного случая, полученного А. Роткевичем, приведена в работе [10].

1. Определение функции Мебиуса

Определение. *Функцией Мебиуса $\mu(n)$ ($n \in \mathbf{N}$) называется функция, значение которой равно: 1 при $n=1$; $(-1)^s$ при $n=\rho_1\rho_2\dots\rho_s$; 0 при $n:\rho^2$, где $\rho_1, \rho_2, \dots, \rho_s$ – попарно различные положительные простые числа, $s \in \mathbf{N}$, ρ – простой делитель числа n .*

Это определение можно записать в виде формулы так:

$$\mu(n) = \begin{cases} 1, & \text{если } n=1, \\ (-1)^s, & \text{если } n=\rho_1\rho_2\dots\rho_s, \\ 0, & \text{если } \rho^2 | n, \end{cases}$$

где $\rho_1, \rho_2, \dots, \rho_s$ – попарно различные положительные простые числа, $s \in \mathbf{N}$, ρ – простой делитель числа n .

Отметим, что Август Фердинанд Мебиус (1790 – 1868) – немецкий математик и астроном. Он известен как геометр. Мебиус рассмотрел

свойства функции $\mu(n)$, где $n \in \mathbf{N}$. Эта функция получила его имя. Функция $\mu(n)$ встречается впервые в работе Мебиуса в 1832 г.

Приведем свойства функции Мебиуса.

1. Если $d, n \in \mathbf{N}$, то

$$\sum_{d|n} \mu(n) = \begin{cases} 1, & \text{если } n = 1, \\ 0, & \text{если } n > 1. \end{cases}$$

2. (Формула обращения Мебиуса). Если $d, n \in \mathbf{N}$, $f(n), F(n)$ – числовые функции, такие, что для любого натурального n

$$F(n) = \sum_{d|n} f(d),$$

то выполнено равенство:

$$f(n) = \sum_{d|n} \mu(d) F\left(\frac{n}{d}\right).$$

Вторая формула называется обращением первой формулы суммирования по делителям.

Отметим, что числовой функцией называется функция, определенная для любого натурального числа $n \in \mathbf{N}$.

Значением числовой функции может быть действительное число или комплексное число.

2. График функции Мебиуса

Для вычисления значения функции Мебиуса $\mu(n)$, где $n \in \mathbf{N}$, при больших n необходима факторизация числа n . Факторизацией называется процесс представления числа n в каноническом виде:

$$n = \rho_1^{\alpha_1} \rho_2^{\alpha_2} \dots \rho_s^{\alpha_s},$$

где $\rho_1, \rho_2, \dots, \rho_s$ – попарно различные положительные простые числа, $\alpha_1, \alpha_2, \dots, \alpha_s \in \mathbf{N}, s \in \mathbf{N}, n \in \mathbf{N}, n > 1$.

Общий метод факторизации заданного числа n заключается в том, что пробуют делить последовательно на простые числа $2, 3, 5, \dots, \rho_k, \rho_k \leq \sqrt{n}$, до тех пор, пока не найдется простое число ρ такое, что $n:\rho$.

Если такое ρ находится, то факторизация числа n сводится к факторизации меньшего числа n_1 , где $n = \rho \cdot n_1$, $n_1 \in \mathbf{N}, n_1 < n$. Если среди всех этих простых чисел

$$2, 3, 5, \dots, \rho_k, \rho_k \leq \sqrt{n}$$

нет ни одного простого делителя числа n , то число n является простым числом.

Отметим, что для больших чисел n этот алгоритм требует долгих вычислений. О разложении на множители натурального числа и распознавании простоты числа можно сказать, что распознавание простоты числа, содержащего 125 и более цифр в десятичной записи, проводится за несколько минут. Но разложение на множители такого числа требует при использовании самых современных алгоритмов очень много времени.

Задача разложения на множители для числа, о котором известно, что оно есть произведение двух простых чисел, по-прежнему остается более трудной, чем определение простоты числа с таким же числом десятичных знаков.

При написании алгоритма вычисления функции Мебиуса необходимо учитывать следующее.

1. Для всех натуральных чисел n от 1 до заданного m выполнить: если $n=1$, то присвоить $\mu(n)=1$ и перейти к следующему n ; иначе начать деление n на простое число p для всех заданных p .

2. В случае, если n не делится ни на одно простое p , $p \leq \sqrt{n}$, то число n является простым числом, поэтому присвоить $\mu(n)=-1$ и перейти к следующему n .

3. В случае, если n делится только на одно простое число p , частное $q=1$, то присвоить $\mu(n)=-1$.

4. В случае, если n делится на s простых чисел, но не делится на квадрат простого числа, присвоить $\mu(n)=(-1)^s$ и перейти к следующему n .

5. Если n делится на квадрат простого числа, то присвоить $\mu(n)=0$ (таблица).

Таблица

Значения функции Мебиуса

n	1	2	3	4	5	6	7	8	9	0	...
$\mu(n)$	1	-1	-1	0	-1	1	-1	0	0	1	...

Отметим, что графиком функции Мебиуса $\mu(n)$ (рисунок), где $n \in \mathbf{N}$, будет множество точек плоскости, которые не будут соединены между собой, так как $n \in \mathbf{N}$.

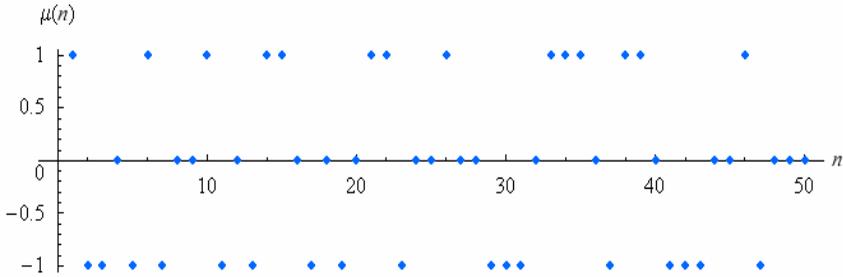


Рисунок. График функции Мебиуса

3. Приложение функции Мебиуса к доказательству частного случая теоремы Дирихле об арифметической прогрессии

Теорема 2. *Каково бы ни было натуральное число n , существует бесконечное множество простых чисел вида $nk+1$, где k – натуральное число.*

Приведем приложение функции Мебиуса к доказательству теоремы 2 о бесконечности множества простых чисел вида $nk+1$ где $n, k \in \mathbf{N}$, которое получил А. Роткевич в 1961 г. Теорема 2 является частным случаем теоремы Дирихле об арифметической прогрессии. Доказательство теоремы 2 является элементарным доказательством, то есть не содержит теории функций комплексной переменной.

Предположим, что натуральное число $n > 2$. Выделим две части:

- доказать, что существует простое число вида $nk+1$ для любого натурального $n > 2$, то есть простое число вида $nmt+1$, которое $> t$ и имеет вид $nk+1$ при $k = mt$ ($n, k, m, t \in \mathbf{N}$);
- доказать, что существование простого числа вида $nk+1$ для любого натурального числа n является достаточным условием для существования бесконечного множества простых чисел $nk+1$, где $k \in \mathbf{N}$.

В первой части определяется некоторая функция равенством:

$$f_n(a) \stackrel{\text{def}}{=} \prod_{d|n} (a^d - 1)^{\mu\left(\frac{n}{d}\right)},$$

где $\mu(k)$ – функция Мебиуса, $d, k \in \mathbf{N}$, $\prod_{d|n}$ – произведение по натуральным делителям d числа n ; $a, n \in \mathbf{N}$, $n > 2$, $n|a$.

Так как функция Мебиуса принимает значения 1 или (-1), или 0, то при разложении числа $a^d - 1$ на простые множители (для всех $d|n$) получим, что число в правой части является произведением степеней простых чисел с целым показателем: положительным или отрицательным, или нулевым.

Заключение

В настоящей работе приведено приложение функции Мебиуса к доказательству частного случая теоремы Дирихле об арифметической прогрессии, полученного А. Роткевичем. Изучена функция Мебиуса и построен график функции Мебиуса. Выполнен перевод статьи А. Роткевича с французского языка.

Функция Мебиуса применяется при доказательстве частного случая теоремы Дирихле об арифметической прогрессии.

Список литературы

1. Бухштаб А. А. Теория чисел / А. А. Бухштаб. – 2-е изд., исправл. – М. : Просвещение, 1966. – 384 с.
2. Рыбников К. А. История математики / К. А. Рыбников. – М. : Изд-во Московского университета, 1994. – 496 с.
3. Стройк Д. Я. Краткий очерк истории математики / Д. Я. Стройк. – М. : Наука, 1984. – 285 с.
4. Гельфонд А. О. Элементарные методы в аналитической теории чисел / А. О. Гельфонд, Ю. В. Линник. – М. : ГИФМЛ, 1962. – 272 с.
5. Карацуба А. А. Основы аналитической теории чисел / А. А. Карацуба. – 2-е изд., перераб. и доп. – М. : Наука, 1983. – 240 с.
6. Чандрасекхаран К. Введение в аналитическую теорию чисел / К. Чандрасекхаран; перевод с англ. С. А. Степанова. – М. : Мир, 1974. – 187 с.
7. Виноградов И. М. Основы теории чисел / И. М. Виноградов. – 9-е изд., перераб. – М. : Наука, 1981. – 176 с.
8. Трост Э. Простые числа / Э. Трост. – перевод с нем. Н. И. Фельдмана. – М. : ГИФМЛ, 1959. – 135 с.
9. Rotkiewicz A. Demonstration arithmetique de l'existence d'une infinite de nombres premiers de la forme $nk+1$ / A. Rotkiewicz // L'enseignement Mathematique. – 1961. – v. VII. – P. 277-280.
10. Сивоплясов Т. М. О доказательстве частного случая теоремы Дирихле об арифметической прогрессии / Е. В. Вахитова, С. Р. Вахитова, Т. М. Сивоплясов. – Материалы международной конференции «Воронежская зимняя математическая школа» (ВГУ, 27 января – 2 февраля 2018 г.). – Воронеж : Издательско-полиграфический центр «Научная книга», 2018. – С. 166-168.

ОЦЕНКА ПОПУЛЯРНОСТИ ФАЙЛОВЫХ РЕСУРСОВ КОРПОРАТИВНОЙ ИНФОРМАЦИОННОЙ СЕТИ

С. В. Смотров
Студент

В. В. Гаршина
Доцент

Е. А. Копытина
Ассистент

Введение

В настоящее время человека окружает огромное количество информации. Человек стремится обработать весь этот поток информации, сохраняя необходимую её часть на дисковом пространстве в виде файла.

На персональном компьютере человек может хранить абсолютно любые файлы сколь угодно долго. Но у корпоративной информационной сети большое число пользователей и количество хранимой информации оказывает влияние, в частности, на быстродействие. В некоторых ситуациях для предприятия быстродействие играет решающую роль и может влечь за собой значительную потерю денежных средств.

В распоряжении системных администраторов необходим инструмент, который позволяет производить оценивание важности хранимых в корпоративной сети файловых ресурсов предприятия. По степени важности, хранимые в корпоративной сети файлы можно разделить на:

- Статически важные – файл должен храниться, хотя и редко используется;
- Динамически важные – файл часто и регулярно или нерегулярно, но часто используется пользователями (например, один период в году).

На основе данных оценки важности хранимых в корпоративной сети файловых ресурсов администраторы предприятия должны управлять файлами пользователей, например, удалять файлы неиспользуемые длительное время, таким образом, экономя информационные ресурсы, и обеспечивая быстродействие, непосредственно связанное с денежными ресурсами предприятия.

1. Цель работы

В настоящее время в администрировании сетей отсутствует инструмент оценки популярности (или актуальности) файловых ресурсов, на основе которого и должно происходить управление файловыми

ресурсами пользователя. По этой причине необходима разработка программного обеспечения, позволяющего проводить аналитику файлового сервера корпоративной ИС, с целью выявления скрытых закономерностей в распределении и использовании данных с учетом временных, количественных, текстовых данных, информации о создателях и пользователях ресурсов для решения задач оптимизации содержания файлового сервера.

2. Подходы к анализу популярности и актуальности файловых ресурсов

Имеется корпоративная информационная компьютерная сеть факультета компьютерных наук (рис. 1). Необходимо проанализировать популярность и актуальность файловых ресурсов системы.



Рис. 1. Структура сети ФКН

Предполагается 2 варианта анализа: базовый и расширенный.

Базовый анализ проводится без предварительной настройки аудита файловой системы и направлен в основном на анализ актуальности файловых ресурсов корпоративной сети.

Расширенный анализ предполагает дополнительные шаги, в том числе работу с аудитом файловой системы, и этот анализ позволит проверить уже популярность файлов корпоративной информационной сети.

3. Базовый анализ

Базовый анализ основан на обработке результатов с помощью стандартных средств Qt, а именно использование таких классов, как QFileInfo и QDir.

У всех операционных сетей есть свои особенности расположения файловых ресурсов в системе. В ОС Windows реализовано размещение по буквам дисков, в ОС UNIX используется root.

Класс `QDir` не предоставляет методов для определения текущего каталога приложения. Для определения, из какого каталога было запущено приложение, необходимо воспользоваться методом `QApplication::applicationDirPath()`, либо `QApplication::applicationFilePath()`, который вернёт, в добавок ко всему, и имя приложения [1].

Существование директории можно проверить с помощью метода `exists()`. Для перемещаться по директориям, применяется метод `cd()`, берущий, в качестве параметра, абсолютный путь директории, и `cdUp()`. Вызов `cd("../")` эквивалентен вызову метода `cdUp()`.

По завершении выбора директории программа выберет все файлы в выбранном ресурсе (папки, подпапки, подпапки папок, файлы) и получит необходимые для анализа актуальности файловой системы свойства всех файлов выбранного ресурса с помощью базового класса `Qt – QFileInfo`.

Работа данного класса заключается в предоставлении информации о свойствах файла, включая его имя, размер, время последнего изменения, время создания, время последнего открытия, права доступа и т. д. Объект класса `QFileInfo` создается передачей в его конструктор пути к файлу, но так же можно передавать и объекты класса `QFile`.

4. Расширенный анализ

В расширенном анализе будут получены данные по количеству открытий каждого файла, пользователю, который открывал этот файл.

Для этого будет использоваться «Аудит файловой системы».

Для использования режима отслеживания действий пользователя требуется предварительная настройка аудита в системе:

- В групповых политиках безопасности включается возможность аудита доступа к файлам и папкам;
- В локальных политиках безопасности выбираются «Конфигурация расширенной политики», а затем «Доступ к объектам»;
- Выбираются необходимые для проведения аудита папки;
- В свойствах папки общего доступа на файловом сервере переходим в закладку «Безопасность» – «Дополнительно» и переходим в закладку «Аудит» – «Изменить» – «Добавить»;
- Выбираются пользователи, для которых проводится аудит;
- Определяются действия, над которыми будет проводиться аудит. Например, «Создание файлов/дозапись данных» – Успех/отказ, «Создание папок/дозапись данных» – Успех/отказ, «Удаление подпапок/удаление данных» – Успех/отказ;
- Применение политик аудита на выбранные файлы;
- Обзор событий доступа к файловым ресурсам через журнал событий безопасности (лог-файлы системы) [2].

Аудит проводится на основе временного анализа данных, то есть в основе лежат действия, выполняемые пользователями файловой системы. В центре внимания находится именно информация о времени работы с файлами, которая очень полезна при отслеживании пользовательских действий с файлами. Анализу подвергаются два файловых атрибута:

- \$ STANDARD_INFORMATION;
- \$ FILE_NAME.

Интересующие данные находятся в логах системы – необходимо извлечь их с помощью скрипта. Для этого построчно данные записываются в текстовый файл, разделяя данные символами табуляции с целью дальнейшего открытия табличным редактором [3].

Сформированный файл используется для дальнейшего анализа, который проводится с помощью языка программирования R. Язык R предоставляет огромное количество алгоритмов для анализа данных. В данной работе использован кластерный анализ методом *k*-средних. С помощью кластерного анализа можно исследовать любые данные, так как данный вид анализа не накладывает ограничения других статистических методов, такие как однородность и объем выборки [4].

5. Практическая реализация

Разработана программа «Анализ популярности файловых ресурсов».

В качестве средства реализации использован статистически типизированный язык программирования C++ и R; кроссплатформенная библиотека разработки GUI Qt; встраиваемая база данных SQLite; объектно-реляционная система управления базами данных PostgreSQL.

При запуске программы «Анализ популярности файловых ресурсов» администратор наблюдает главное окно приложения для анализа системных файлов / папок / локальных дисков.

Для проведения базового анализа системы следует нажать кнопку «Выбрать папку» и в появившемся диалоговом окне сделать выбор файлового ресурса, который будет анализироваться. Выделив курсором необходимый для анализа ресурс нужно только лишь нажать на кнопку подтверждения «Выбор папки».

Анализ начался, по его завершению в главном окне программы можно увидеть основную информацию, касающуюся файлов выбранного файлового ресурса (рис. 2).

Для получения статистических данных, необходимых для анализа, необходимо нажать кнопку «Статистика». В появившемся окне у пользователя появляется возможность просмотреть информацию о популярности выбранного файлового ресурса. Информация предоставляется в виде гистограмм.

Файл	Размер	Дата создания	Дата последнего редактирования	Дата последнего чтения
1 D:/Program Files/TortoiseHg/help/extensions.txt	1K	22.01.2017	22.01.2017	05.02.2018
2 D:/Program Files/TortoiseHg/help/dates.txt	1K	22.01.2017	22.01.2017	05.02.2018
3 D:/Program Files/TortoiseHg/help/diffs.txt	1K	22.01.2017	22.01.2017	05.02.2018
4 D:/Program Files/TortoiseHg/help/filesets.txt	1K	22.01.2017	22.01.2017	05.02.2018
5 D:/Program Files/TortoiseHg/help/urls.txt	2K	22.01.2017	22.01.2017	05.02.2018
6 D:/Program Files/TortoiseHg/help/patterns.txt	2K	22.01.2017	22.01.2017	05.02.2018
7 D:/Program Files/TortoiseHg/help/phases.txt	2K	22.01.2017	22.01.2017	05.02.2018
8 D:/Program Files/TortoiseHg/help/hgignore.txt	3K	22.01.2017	22.01.2017	05.02.2018
9 D:/Program Files/TortoiseHg/help/hgweb.txt	3K	22.01.2017	22.01.2017	05.02.2018
10 D:/Program Files/TortoiseHg/help/merge-tools.txt	3K	22.01.2017	22.01.2017	05.02.2018
11 D:/Program Files/TortoiseHg/help/environment.txt	3K	22.01.2017	22.01.2017	05.02.2018
12 D:/Program Files/TortoiseHg/help/templates.txt	5K	22.01.2017	22.01.2017	05.02.2018
13 D:/Program Files/TortoiseHg/help/scripting.txt	6K	22.01.2017	22.01.2017	05.02.2018
14 D:/Program Files/TortoiseHg/help/revisions.txt	6K	22.01.2017	22.01.2017	05.02.2018
15 D:/Program Files/TortoiseHg/help/subversion.txt	6K	22.01.2017	22.01.2017	05.02.2018

Рис. 2. Базовый анализ файловых ресурсов

Реализовано 3 вида гистограмм. Гистограмма, построенная на основе даты создания файла (рис. 3), гистограмма, построенная на основе даты последнего редактирования файла и гистограмма, построенная по дате последнего чтения (открытия) файла.

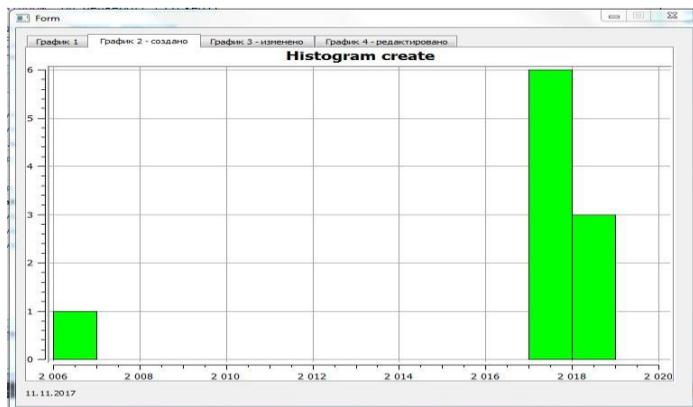


Рис. 3. Гистограмма на основе даты создания файла

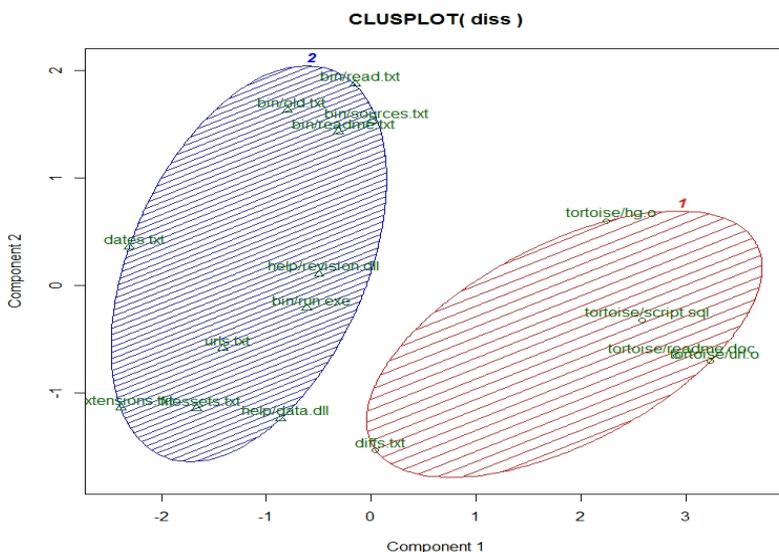
На главном окне имеется кнопка «Сгенерировать отчет», которая позволяет сохранить отчет об анализе какого-либо файлового ресурса в файл формата XML, который можно открыть любым табличным редактором. Отчет содержит полную информацию о свойствах каждого файла в анализируемом ресурсе, включая даты создания, последнего чтения, последнего редактирования файлов, атрибуты файла, размер, расширение, расположение, уникальный идентификатор отчета,

комментарий администратора к сформированному отчету, а также добавляются статистические выкладки о состоянии ресурса с рекомендациями для пользователя (информация о файлах, которые пользователь должен проверить на актуальность). Также имеется возможность добавления в отчет данных о семестре, в котором файл использовался последний раз, или в котором он считается актуальным. Для расширенного анализа предусмотрена кнопка «Загрузить лог-файл» на главном окне программы, которая предназначена для подготовки данных, полученных из журнала безопасности с помощью скрипта Power Shell, к проведению кластерного анализа.

Для того чтобы провести расширенный анализ, требуется провести включение и настройку аудита безопасности, выполнение скрипта в Power Shell, который сгенерирует необходимый лог-файл из журнала безопасности в формате txt.

Полученный текстовый файл будет задействован при проведении кластерного анализа, который был реализован в виде скрипта на языке программирования R.

После выполнения скрипта для проведения кластерного анализа, администратору будет доступна информация о том, в каком из семестров года использовался конкретный файл (рис. 4).



Разбиение файлов по использованию в семестрах

Рис. 4. Кластерный анализ

Заключение

На основе базового и расширенного подхода к оценке популярности файловых ресурсов пользователя корпоративной сети разработано программное обеспечение «Анализ популярности файловых ресурсов», которое позволяет:

- Просматривать параметры файла (дата открытия, дата изменения, дата создания, расширение, размер, расположение);
- Генерировать отчет для базы данных;
- Генерировать отчет в файл с расширением .xml;
- Анализировать данные.

Получив информацию о популярности ресурса и использовании ресурсов по семестрам учебного года, администратор ИС сможет, например, заархивировать файловые ресурсы, которые не используются в текущем семестре, тем самым сэкономив важный ресурс – дисковое пространство.

На основе анализа данных, пользователь – владелец ресурса, получив отчет, получит возможность проверить файлы в своем ресурсе, повысить их актуальность, а неиспользуемые – удалить.

Список литературы

1. Tejpal S. Time Rules for NTFS File System for Digital Investigation / S. Tejpal, K. Manjot // International Journal of Advanced Research in Computer Engineering & Technology. – 2015. – Vol. 4. – No. 4. – P. 1146-1151.
2. Кастер Х. Основы Windows NT и NTFS / Х. Кастер – М. : Русская редакция, 1996. – 440 с.
3. Using the R Software for Log File Analysis [Электронный ресурс]. – Режим доступа : https://www.usenix.org/system/files/1403_11-15_tsoukalos.pdf
4. Кластерный анализ в R [Электронный ресурс]. – Режим доступа : <http://mymagictools.blogspot.ru/2015/07/r.html?m=1>

РАЗРАБОТКА ПЛАГИНА К ANDROIDSTUDIO ДЛЯ ГЕНЕРАЦИИ СЕРВИСНОГО СЛОЯ ANDROID-ПРИЛОЖЕНИЙ ПО СПЕЦИФИКАЦИИ API BLUEPRINT

В. А. Соколова

Студентка

А. Л. Гавшин

Ассистент

Введение

В последние годы рынок мобильных приложений стремительно растет, и всё больше компаний хотят построить свой бизнес или расширить его путём запуска собственных приложений. Они встраиваются в информационную систему компании или пишутся параллельно с ней. Как правило, такие системы построены по принципу клиент-серверного взаимодействия. Мобильное приложение в этой схеме является «тонким клиентом», который получают все необходимые для функционирования данные от сервера. Для построения такого взаимодействия наиболее популярна в данный момент REST архитектура, при которой вызов удалённой процедуры представляет HTTP-запрос, а необходимые данные передаются в качестве параметров этого запроса. Какие данные и в каком виде будут передаваться, описывает спецификация сервера. Набор компонент, отвечающих за отправку и обработку запросов, а также за хранение и преобразование данных в бизнес-модели, называется сервисным слоем приложения.

Настройке взаимодействия клиента и сервера выделяется весомая часть времени при разработке любого подобного приложения, и обычно эта работа сводится к рутинному написанию сервисного слоя по спецификации. Задача автоматизации этого процесса представляется очень актуальной, так как поможет сократить время на разработку и количество опечаток и неточностей. В данной работе описана реализация плагина для среды AndroidStudio, позволяющего автоматически сгенерировать код для сервисного слоя приложения из спецификации в формате API Blueprint.

1. Формат API Blueprint

Весь набор запросов и ответов, составляющих API сервера, описывается с помощью спецификации, которая является своего рода контрактом о том, что и в каком виде клиент рассчитывает получить. Как и любой контракт, этот имеет своё формальное представление в виде

документированного описания. Одним из популярных сервисов для написания спецификаций является Ariagu, он позволяет через графический интерфейс создать свою спецификацию в формате API Blueprint и сделать ее доступной определенному кругу коллег. Внутри спецификация хранится в файле формата .arib, который может быть скачан из Ariagu в любой момент. Язык спецификации напоминает Markdown разметку, но со своими особенностями. Базовые разделы спецификации [1]:

- Название проекта
- Адрес сервера
- Группы
- Запросы
- Ответы
- Используемые data-модели

Пример спецификации представлен на листинге.

Листинг

Пример спецификации в формате API Blueprint

```
FORMAT: 1A
HOST: https://test.com/api/
# Test API
## Текущий пользователь [/user]
### Регистрация пользователя [POST /signup]
+ Attributes (object)
  + phone (optional)
  + clientId (required)
+ Request (application/json)
  + Attributes (UserInfoUpdate)
+ Response 201 (application/json; charset=utf-8)
  + Attributes (object)
    + key: supasecretAuthToken (string, required)
    + expires: `2017-07-18T08:56:35Z` (string, required)

# Data Structures
## UserInfoUpdate
+ name: `Иванов Иван Иванович` (string, optional)
+ email: `email@user.com` (string, optional)
+ birthDate: `2017-07-18T08:56:35Z` (string, optional)
```

2. Структура сервисного слоя android-приложения

Самая популярная связка библиотек для реализации клиент-серверного взаимодействия на Android это Retrofit+RxJava2+GSON. Retrofit – отвечает за поддержку сетевого взаимодействия, помогает формировать запросы и обрабатывать ответы. GSON - библиотека для сериализации и десериализации данных из JSON. RxJava 2 – библиотека для реализации реактивных источников данных.

Структура сервисного слоя будет ориентирована на принципы Clean Architecture [2], схематично эта архитектура представлена на рис. 1.

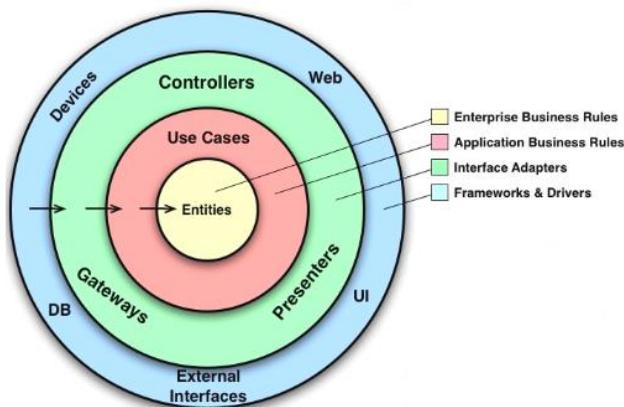


Рис. 1. Структура приложения в соответствии с Clean Architecture

Clean Architecture предполагает деление приложения на следующие слои:

- **Entities** – бизнес-объекты приложения
- **UseCases** – события и действия в приложении
- **InterfaceAdapters**
 - **Controllers** – преобразует действия пользователя в вызовы слоя UseCase
 - **Gateways** – обертки над сервером, базой данных и т.п., которые конвертируют данные из низкоуровневого источника в данные для удобного использования в UseCase (в основной модели Entity)
 - **Presenters** – управляет отображением, конвертирует данные полученные от слоя UseCase в структуру, подходящую для UI и наоборот.
- **Frameworks and Drivers**
 - **UI** – отвечает за взаимодействие с пользователем
 - **DB, Web** – API сервера, базы данных и других источников данных.

На самом деле, количество слоев может быть различным, оно зависит от сложности системы – чем сложнее система, тем на большее количество подсистем ее нужно разделить для уменьшения сложности. Для базовой генерации мы объединим слои UseCases и Gateways в один, так как

разделять их имеет смысл, только если логика данного фрагмента приложения очень сложна. Рассмотрим распределение сущностей по слоям (рис. 2). Именно эти типы классов генерирует плагин.

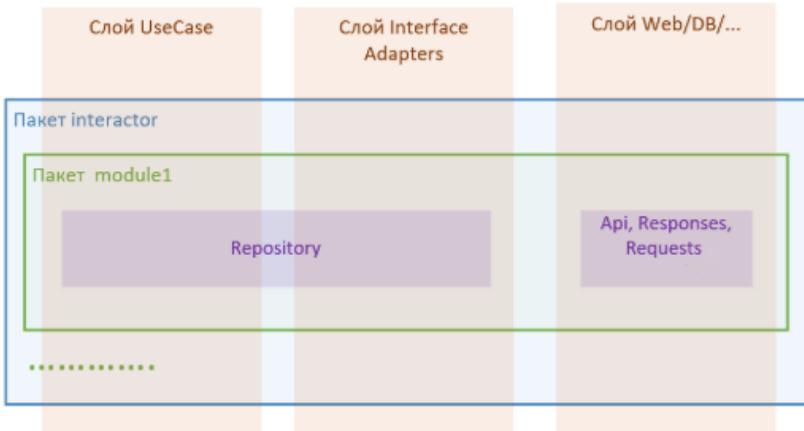


Рис. 2. Структура сервисного слоя приложения на базе Clean Architecture

- **Api** – классы, определяющие интерфейс для Retrofit
- **Responses, Requests** - классы, с моделями для сериализации и десериализации через GSON
- **Repository** – обёртка над Api, Responses, Request. Производит конвертацию входных параметров в Requests и преобразование ответов сервера (Responses) в сущности слоя Entity.

3. Разработка плагина

На данный момент самая популярная IDE среди android-разработчиков – AndroidStudio. Она является продуктом компании JetBrains. JetBrains предоставляют возможность разрабатывать плагины для любых своих продуктов посредством IntelliJ IDEA [3]. Готовый плагин собирается в .jar и легко подключается через настройки IDE.

В процессе работы плагин формирует структуру будущих пакетов и классов и представляет ее пользователю (рис. 3). Пакет с проектом содержит вложенные пакеты групп запросов и моделей. Каждая группа имеет два класса Repository и Api, в названии класса содержится префикс – имя группы. Для облегчения редактирования на превью эти классы представлены пакетами, содержащими отдельные методы. Можно выбрать конкретный класс и сменить его сигнатуру, название или комментарий, или выбрать отдельный метод, для которого доступны те же операции.

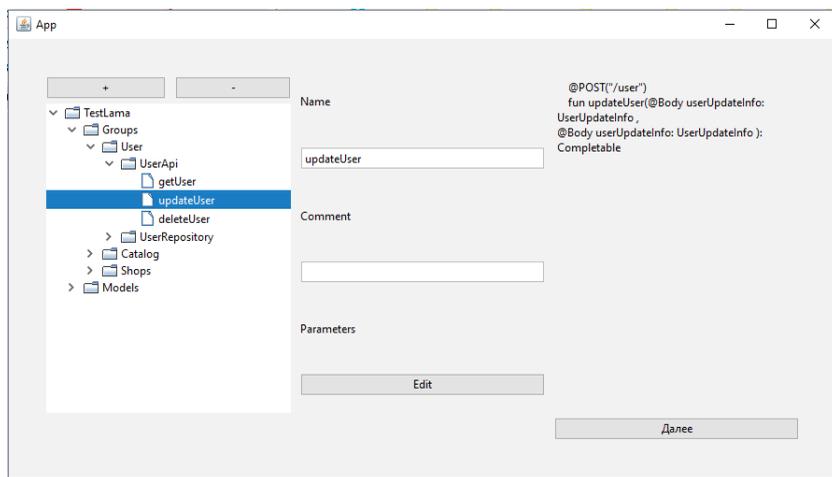


Рис. 3. Внешний вид панели превью и редактирования перед генерацией

После завершения редактирования пользователь указывает директорию, куда положить сгенерированные пакеты и классы. Плагин генерирует код на языке Kotlin, каждый класс и метод снабжается шаблоном для комментария.

Заключение

На практике удалось разработать систему, которая позволяла бы генерировать сервисный слой приложения из спецификации API Blueprint. В частности, классы репозитория, интерфейсы Retrofit и модели для парсинга в JSON. Разработанный плагин решает проблемы: написания скучного однотипного кода, больших временных затрат, недостаточного документирования, а также опечаток и ошибок, неизменно возникающих в аннотациях моделей и параметрах запросов.

Список литературы

1. API Blueprint Documentation [Электронный ресурс]. – Режим доступа : <https://apiblueprint.org/documentation/>
2. Robert Martin The Clean Architecture [Электронный ресурс]. – Режим доступа : <https://www.8thlight.com/blog/uncle-bob/2012/08/13/the-clean-architecture.html>
3. IntelliJ Platform SDK DevGuide : Part I [Электронный ресурс]. – Режим доступа : <https://www.jetbrains.org/intellij/sdk/docs/basics.html>

АРХИТЕКТУРА СИСТЕМЫ ИДЕНТИФИКАЦИИ P2P ТРАФИКА

А. Н. Сотников

Студент

А. С. Коваль

Старший преподаватель

Введение

Трафик peer-to-peer (P2P) представляет собой значительную часть сегодняшнего глобального интернет-трафика [1]. Поэтому для сетевых администраторов важно определять и классифицировать P2P-трафик, пересекающий границы сети, чтобы можно было реализовать соответствующие мелкомасштабные политики сетевого управления и безопасности. Кроме того, возможность категоризации трафика P2P может помочь повысить точность сетевых систем обнаружения вторжений.

Несмотря на то, что существует большое количество работ, посвященных обнаружению трафика P2P [2], большая часть этих работ требует глубокой проверки пакетов (DPI) или идентификации на основе номера порта. Поскольку современные приложения P2P не используют фиксированные номера портов и используют шифрование для предотвращения обнаружения на основе DPI [1], в более поздних работах была рассмотрена проблема идентификации данных P2P на основе статистического анализа трафика. Однако очень немногие из этих исследований выходят за рамки обнаружения трафика P2P в сторону решения проблемы категоризации трафика P2P [3]. Кроме того, в ряде исследований основное внимание было уделено обнаружению бот-сетей P2P, но без какого-либо углубления, возможность точно различать различные типы бот-сетей P2P на основе их шаблонов трафика P2P в них практически не рассматривается. Насколько известно, ни одно из предшествующих исследований не предложило общий подход к категоризации сетевого трафика, связанного с разрешенный и вредоносными (например, связанными с ботнетом) P2P-приложениями.

В данной работе используется архитектура системы классификации трафика P2P представленная в работе [4], проводится ее оценка и анализ возможных алгоритмов машинного обучения, для дальнейшего использования. Система основана на общем подходе к классификации, который использует высокоуровневые статистические характеристики трафика и способен точно определять и классифицировать трафик,

создаваемый различными приложениями P2P, включая обычные приложения для совместного использования файлов.

1. Описание архитектуры

На рис. 1 представлен обзор архитектуры, который будет подробно обсужден далее. Первый шаг включает идентификацию хостов P2P в контролируемой сети. Затем модуль классификации трафика P2P анализирует сетевой трафик, генерируемый этими хостами, и пытается связать его с данным приложением P2P путем сопоставления профиля приложения, ранее полученного из выборок трафика, генерируемого известными приложениями P2P. Если трафик P2P не соответствует ни одному из доступных профилей, трафик классифицируется как принадлежащий к «неизвестному» P2P-приложению (например, это может представлять собой новую версию приложения P2P или ранее неизвестную бот-сеть P2P), и ее следует дополнительно проанализировать сетевым администратором. С другой стороны, если трафик P2P соответствует более чем одному профилю, для того чтобы «разорвать связь» используется вспомогательный модуль устранения неоднозначности, а трафик помечен как принадлежащий к ближайшему профилю приложения P2P.



Рис. 1. Архитектура приложения идентификации P2P приложений

Профили приложений могут моделировать характеристики трафика разрешенных приложений P2P, а также различные P2P-ботнеты. Общеизвестно, что исследователи безопасности запускают образцы ботнета в контролируемой среде для изучения их системных и сетевых действий. Трафик, собранный во время этого процесса, затем может быть использован в качестве образца для обучения конкретному профилю

приложения P2P-botnet, который может быть подключен к нашему модулю категоризации трафика P2P.

Основная цель разрабатываемого приложения - обнаружение нежелательного трафика P2P в контролируемой компьютерной сети. Поскольку точное определение того, что трафик нежелателен, зависит от политик управления и безопасности каждой сети, мы принимаем общий подход к категоризации трафика P2P и оставляем окончательное решение о том, какой трафик нарушает политики для сетевого администратора.

Приложение разделяет поток сетевого трафика в реальном времени на временные окна постоянного размера W (например, $W = 10$ минут). В конце каждого временного окна оно извлекает ряд статистических характеристик из наблюдаемого сетевого трафика и преобразует трафик, генерируемый каждым хостом H в сети, в отдельный вектор F_H . Каждый вектор F_H функции может быть затем подан в ранее подготовленный статистический классификатор, который специализируется на обнаружении того, может ли H быть использован приложением, генерирующим P2P трафик, как указано в его характеристиках в рассматриваемом временном окне. Разделение анализа трафика во временных окнах позволяет генерировать периодические отчеты и приводит к более точным результатам путем объединения результатов, полученных в последовательных временных окнах.

Классификатор, используемый в обнаружении хоста P2P, обучается с использованием образцов сетевого трафика, создаваемого хостами, которые, как известно, используют множество приложений P2P, а также образцы трафика с хостов, которые, как полагают, не используют какое-либо известное приложение P2P. Когда хост H классифицируется как хост P2P в течение заданного временного окна W первым модулем, его текущий сетевой трафик (т.е. Трафик, собранный во время текущего временного окна W) отправляется в модуль классификации трафика P2P. Этот модуль состоит из нескольких классификаторов одного класса, именуемых «профили приложений» как видно из изображения на рис 1, причем каждый классификатор специализируется на обнаружении того, используется ли H конкретным приложением P2P или нет. Каждый классификатор одного класса обучается с использованием только ранее собранных проб трафика, относящихся к известному P2P-приложению. Например, мы обучаем классификатор одного класса для обнаружения трафика Skype, один для eMule, один для Storm P2P-botnet и т. Д. Это позволяет нам создавать новый профиль приложения независимо от ранее изученных моделей трафика. Таким образом, мы можем обучать и развертывать другую конфигурацию оптимального классификатора для каждого целевого приложения P2P и временного окна анализа W .

Трафик H , переводится в вектор функций категоризации или трафик профиля, P_H (обратите внимание, что эти функции отличаются от функций обнаружения F_H , используемых в предыдущем модуле). Затем трафик H передается каждому из доступных классификаторов одного класса, и каждый классификатор выводит оценку, которая показывает, насколько близко профиль P_H относится к профилю приложения, который был обучен классификатору для распознавания. Например, если классификатор Skype выводит высокий балл, это означает, что P_H очень напоминает трафик P2P, созданный Skype. Если ни один из классификаторов одного класса не выдает достаточно высокий балл для P_H , Приложение не может приписать P2P-трафик H известному P2P-приложению, а профиль трафика P2P P_H обозначается как «неизвестный». Такое решение может быть вызвано разными причинами. Например, на обнаруженном хосте P2P может быть запущено новое приложение P2P, для которого образец трафика не был доступен во время обучения профилей приложения или может быть заражен ранее неизвестным P2P-ботнетом.

Из-за характера статистических классификаторов, более чем один классификатор может заявить, что P_H близок к их профилю приложения. Другими словами, возможно, что модуль классификации трафика P2P может вынести решение, что на H работает как Skype, так и eMule, например. В этих случаях, чтобы попытаться разорвать несоответствие, приложению необходимо отправить профиль P_H в модуль устранения неоднозначности, который состоит из специального классификатора, который специализируется на определении того, какой профиль приложения фактически является самым близким к входному профилю P_H . По сути, вывод модуля устранения неоднозначности может быть использован сетевым администратором в сочетании с выходом отдельных профилей приложений, которые «сопоставляют» трафик, чтобы помочь в дальнейшем выяснении и определении того, является ли узел нарушением политик.

2. Определение узла P2P

Первым этапом, в соответствии с архитектурой, приведенной на рис. 1 является определение будет ли узел узлом P2P. Данный модуль представляет собой реализацию архитектуры, приведенной на рис. 2.

На вход данному алгоритму подается набор характеристик пакета: srcIP, srcPort, destIP, destPort, tcp. Результатом классификации будет являться ответ, является ли трафик хоста H P2P трафиком, или не-P2P. Данный алгоритмы был выбран так как он просты в реализации, а также имплементированы в программной платформе анализа данных WEKA на языке Java.

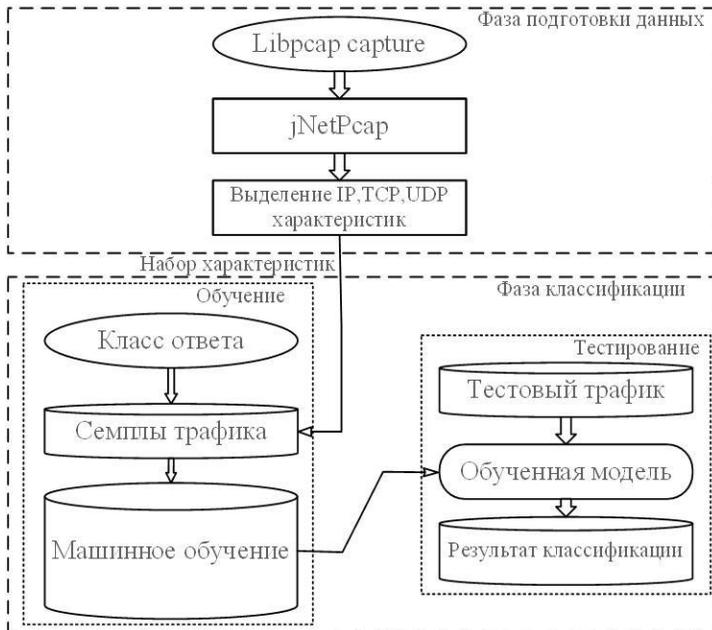


Рис. 2. Архитектура модуля определения узла P2P

3. Классификация трафика P2P

После того, как мы идентифицировали hosts P2P в контролируемой сети, модуль классификации трафика P2P определяет, какой тип приложения P2P использует эти hosts. Поскольку различные приложения P2P (в том числе P2P-ботнеты) используют разные протоколы и сети, они показывают различное поведение с точки зрения их моделей сетевой связи. Поэтому мы строим систему классификации, которая может изучать различные профили приложений P2P из прошлых пробных проб трафика и может точно классифицировать новые экземпляры трафика P2P.

Как показано на рис 1, модуль классификаций состоит из ряда классификаторов, специализирующихся на распознавании конкретного профиля приложения. Например, мы обучаем классификатор для распознавания трафика P2P, генерируемого Skype, другой для распознавания трафика eMule и т.д. Систему можно легко расширить до новых приложений P2P, обучив специализированный классификатор одного класса на новом P2P-трафике и подключив полученный профиль приложения к модулю категоризации.

Учитывая P2P трафик, генерируемый ранее обнаруженным хостом H, мы сначала извлекаем ряд статистических признаков (описанных ниже),

которые составляют профиль P_H , H трафика в течение заданного временного окна. Затем мы передаем P_H каждому из ранее подготовленных классификаторов, и для каждого из них мы получаем оценку обнаружения. Например, пусть s_k - результат оценки классификатором, предназначенным для распознавания Skype. Если s_k превышает predetermined порог θ_k , который автоматически изучается во время обучения, существует высокая вероятность того, что H используется Skype. Если ни один классификатор не выводит оценку s_i (где индекс i указывает i -й классификатор), превышающий порог обнаружения приложения θ_i , мы обозначаем трафик P2P из H как «неизвестный». То есть приложение обнаружило тот факт, что H используется приложением P2P, но профиль трафика не подходит ни к одной из ранее подготовленных моделей.

Стоит обратить внимание, что порог θ_i задается во время фазы обучения для ограничения ложноположительной вероятности срабатывания до $\leq 1\%$. В частности, ложные срабатывания, создаваемые i -м классификатором по трафику из приложений P2P, отличных от такового, на которые нацелен классификатор, составляют $\leq 1\%$.

Из-за природы статистических классификаторов возможно, что более чем один классификатор может получить оценку s_i больше, чем соответствующий порог обнаружения θ_i , тем самым заявляя, что P_H соответствует их профилю приложения. В этом случае, чтобы разрешить коллизию, мы используем модуль устранения неоднозначности трафика P2P, который состоит из классификатора с несколькими классами, обученного различать различные приложения P2P. В этом случае классификатор с множеством классов определенно назначит одно приложение среди доступных. Эта информация, наряду с выводами каждого классификатора одного класса, затем может быть использована администратором сети, чтобы помочь решить, является ли трафик из H вредоносным или нет.

4. Оценка модуля определение узла P2P

В этом разделе проводится оценка модуля определения узла трафика P2P. Так как объем статьи ограничен в данной работе будет проведена оценка только первого модуля – модуля определения узла P2P.

Для обучения модуля классификации основанном на дереве принятия решений используется программная платформа WEKA и образцы трафика P2P приложений из свободного доступа, а именно трафик, принадлежащий приложению BitTorrent, SoulSeek и SMPP, Skype и др все экземпляры указанного трафика находятся в свободном доступе, часть из них собрана при помощи программного средства WireShark.

Для того чтобы приступить к обучению классификатора, необходимо представить данные полученные при помощи платформы JNetPcap в виде

формата arff (Attribute Relation File Format), данный формат используется программной платформой WEKA. Это выполняется при помощи набора инструментов arff tool.

В результате обучения были получены 2 вида классификаторов как был указано выше:

- Дерево принятия решений C4.5(J48)
- Наивный байесовский классификатор

Для тестирования полученных классификаторов использовали экземпляры трафика, содержащие как P2P, так и не-P2P потоки (таблица). В результате тестирования при помощи встроенных программных средств платформы WEKA были получены результаты, отображены на рис 3.

Таблица

Выборка данных для тестирования

Категория приложения	Приложения/протоколы	Количество потоков
Не-P2P	Mail, http, https, ftp	25 542
P2P	BitTorrent, SoulSeek, Skype, eMule, SMPP	50 896

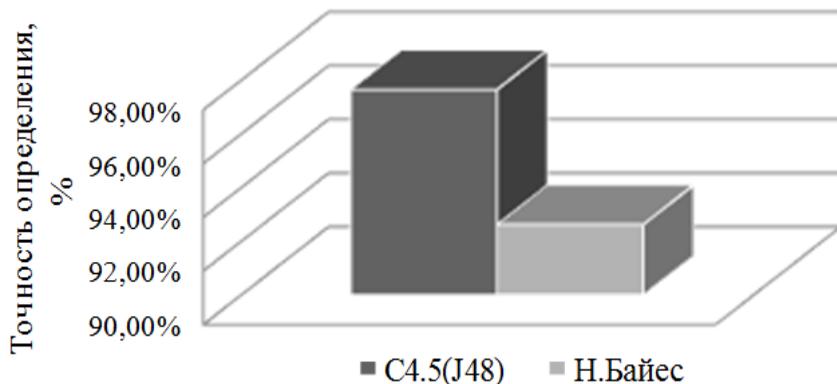


Рис. 3. Результаты эксперимента

В ходе эксперимента было выяснено что решение, основанное на алгоритме C4.5(J48) является более точным и быстрым. Что подтверждает положение, предложенное в работе [5], в которой изучались алгоритмы построения деревьев решений, и был сделан вывод, что и в скорости обучения алгоритм C4.5 имеет преимущества по сравнению с другими алгоритмами. Недостатком этого алгоритма является построение деревьев больших размеров.

Заключение

Хотя обнаружение трафика P2P было предметом большого количества исследований, идентификация трафика P2P получила очень мало внимания. В данной работе была представлена лишь малая часть исследования – проведена оценка предложенной архитектуры системы идентификации P2P трафика, и выбран алгоритм классификации для одного из модулей.

Дальнейшие работы будут связаны с извлечением характеристик трафика для более точной классификации P2P приложений.

Список литературы

1. Madhukar A. Longitudinal study of p2p traffic classification / A. Madhukar, C. Williamson // A In Proceedings of the 14th IEEE International Symposium on Modeling, Analysis, and Simulation. – 2006. – № 6. – P. 277-298.
2. Gomes J. Detection and classification of peer-to-peer traffic / J. Gomes, P. Inacio // A survey ACM Computing Surveys. – 2012. – P. 4-8.
3. Chiu D. Profiling and identification of P2P traffic / D. Chiu // Comput. Netw. – 2009. – №4. – P. 849-863.
4. Perdisci R. Mining for Unwanted P2P Traffic / R. Perdisci, B. Rahbarinia // Detection of Intrusions and Malware, and Vulnerability Assessment: 10th International Conference, DIMVA 2013, Berlin, Germany, July 18-19. – 2013. – P. 2-4.
5. Lim T. A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms / T. Lim, W. Loh, Y. Shih // Machine Learn. – 2000. – №8. – P. 203-229.

ПРИМЕНЕНИЕ BDD-ПОДХОДА В ЗАДАЧЕ КРОСС-ПЛАТФОРМЕННОГО АВТОМАТИЗИРОВАННОГО ТЕСТИРОВАНИЯ МОБИЛЬНЫХ ПРИЛОЖЕНИЙ

М. В. Старкина

Студент

В. В. Гаршина

Доцент

Введение

За последние годы мобильные устройства, такие как телефоны или планшеты, только набирают свою популярность. С этим связано увеличение на рынке и количества приложений для этих устройств. Приложения используются для удовлетворения совершенно разных

потребностей и для выполнения различных задач. Однако в связи с большим количеством мобильных платформ становится не так просто иметь правильно работающие приложения.

Полноценное тестирование [1] мобильных приложений требует тестирования приложения на разных устройствах, которые имеют различные размеры экранов, версии операционные системы и другие отличающиеся характеристики.

Из-за абсолютного понимания важности тестирования приложений появились особые инструменты, созданные специально для тестирования кросс-платформенных мобильных приложений. Однако, даже эти инструменты являются новыми и постоянно развивающимися. Мобильные операционные системы имеют явную тенденцию к постоянному обновлению, а, в следствие, необходимо и обновлять эти инструменты, поскольку к существующим операционным системам добавляются новые функции или изменяются старые.

1. Концепция BDD (Behavior Driven Development)

BDD (Behavior Driven Development) – это процесс разработки программного обеспечения, направленный на тестирование [2]. Этот подход вырос из Test Driven Development (TDD) примерно с середины 2000-х годов. BDD фокусируется на четкой идентификации желаемого поведения функции с самого начала разработки. Поведение идентифицируется с использованием спецификации, например: поведенческие характеристики будут написаны для иллюстрации желаемого поведения с реалистичными примерами, а при помощи абстрактного, общего жаргона. Они служат как требования продукта / критерии приемки (до разработки), так и его тестовыми примерами (после разработки).

Gherkin – один из самых популярных языков для написания формальных характеристик поведения – он фиксирует поведение как сценарии «Given-When-Then». С помощью средств автоматизации сценарии можно легко превратить в автоматизированные тест-кейсы. Любой из инженеров для владельцев продуктов может писать сценарии BDD, так как они просто английские фразы (также поддерживается русский язык). BDD помогает разработчикам сосредоточиться на представлении продукта именно таким, каким его хочет видеть владелец продукта. Он также ускоряет тестирование. Таким образом, BDD прекрасно сочетается с Agile Software Development.

Gherkin – это предметно-ориентированный язык для написания поведенческих сценариев. Это простой язык программирования, а его «код» записывается в файлы функций (текстовые файлы с расширением «.feature»). Официальным языковым стандартом Gherkin является

Cucumber, одна из наиболее распространенных структур автоматизации BDD. Сценарии Gherkin должны быть короткими и звучать как простые предложения. Каждый сценарий имеет следующую структуру:

- Когда есть некоторое состояние системы;
- Если совершается действие над системой;
- Тогда система переходит в новое состояние.

Последующая автоматизация – одно из преимуществ BDD. Фактически, основную цель BDD можно резюмировать как быстрое превращение концептуального поведения в автоматически тестируемое поведение. В то время как процесс и Gherkin универсальны, базовая автоматизация может быть построена с использованием одной из многих инфраструктур.

2. Структура системы автоматизации BDD

Каждая система автоматизации BDD имеет пять основных частей [3]:

1. Feature file (Файлы функций).

Файлы функций Gherkin очень важны для автоматизации. Они действуют как тестовые сценарии - каждый сценарий по существу является тест-кейсом.

Вот пример файла с именем google_search.feature:

Feature: Google Searching

As a web surfer, I want to search Google, so that I can learn new things.

The same scenario

@automated @google-search

Scenario: Simple Google search

Given a web browser is on the Google page

When the search phrase "ВГУ" is entered

Then results for "ВГУ" are shown

2. Step definition (Определения шага).

Определение шага – это блок кода, который реализует логику для выполнения шага. Обычно это метод или функция с англоязычной фразой в качестве аннотации. Определения шагов могут принимать аргументы, строки doc и таблицы шагов. Когда сценарий выполняется, драйвер соответствует каждой фразе шаг шага к его определению шага. (Большинство фреймворков используют регулярные выражения для сопоставления фраз). Таким образом, каждый шаг в файле функций требует определения шага.

3. Hooks (Крючки).

Определенная логика автоматизации не всегда может обрабатываться с помощью определений шагов. Например, для сценариев могут потребоваться специальные операции настройки и очистки. Большинство

BDD-фреймворков предоставляют перехваты, которые могут вставлять вызовы до или после разделов Gherkin, которые обычно фильтруются с использованием тегов. Крюки схожи по концепции с аспектно-ориентированным программированием.

4. Support Code (Код поддержки).

Код поддержки (библиотеки или пакеты) относится к любому коду, называемому определениями шагов и перехватами. Код поддержки может быть пакетом зависимостей, загружаемым с использованием таких менеджеров, как Maven (Java), NuGet (.NET) или PyPI (Python). Например, Selenium WebDriver – это хорошо известный пакет для автоматизации веб-браузера. Код поддержки может также быть компонентами для автоматизации, такими как объекты страниц или другие шаблоны проектирования. Определения шагов и крючки не должны содержать всю логику для запуска действий - они должны как можно больше использовать общий код.

5. Driver (Драйвер).

Каждая инфраструктура автоматизации имеет драйвер, который запускает тесты, а рамки BDD ничем не отличаются. Драйвер выполняет каждый сценарий в файле свойств независимо. Всякий раз, когда происходит сбой, драйвер сообщает об ошибке и прерывает сценарий.

Драйвер behave можно запустить из командной строки следующим образом:

```
> behave google_search.py --tags @automated
```

3. **Достижение кросс-платформенности.**

Стоит отметить такой важный момент, что создание тест-кейсов может привести к сложно обслуживаемому проекту. Одна из причин заключается в том, что используется слишком много дублированного кода. В ситуации, когда нам нужно работать с двумя платформами, эта проблема как никогда актуальна, так как различие платформ никак не означает различие ожидаемого поведения. Получится дублированный, вызванный дублированными функциями, и это приведет к дублированию использования локаторов. Если какой-либо локатор изменится, придется пройти весь тестовый код, чтобы при необходимости настроить локаторы. Для решения такой проблемы используют паттерн page object [4], при помощи которого уменьшается или даже исключается дублирующий тестовый код. Кроме того, он улучшает читаемость и позволяет создавать интерактивную документацию. И последнее, но не менее важное: мы можем создавать тесты с меньшим нажатием клавиши. Реализация объектной модели страницы может быть достигнута путем разделения абстракции тестового объекта и тестовых скриптов.

Page Object – это шаблон проектирования, который широко используется в автоматизированном тестировании и позволяет разделять логику выполнения тестов от их реализации. Page Object как бы моделирует страницы тестируемого приложения в качестве объектов в коде. В результате его использования у вас получатся отдельные классы, отвечающие за работу каждой конкретной функции. Такой подход значительно уменьшает объем повторяющегося кода, потому что одни и те же объекты страниц можно использовать в различных тестах. Основное преимущество Page Object заключается в том, что в случае изменения пользовательского интерфейса, можно выполнить исправление только в одном месте, а не исправлять каждый тест, в котором этот интерфейс используется.

Класс PageObject не обязательно должен представлять собой всю страницу. Он может быть частью страницы, которая часто используется (или даже на одной странице). Это может быть, например, пагинатор или меню. Основной принцип состоит в том, что есть только один класс в проекте, который знает о структуре конкретной страницы или ее части.

Заключение

В статье представлен метод тестирования мобильных приложений с использованием BDD-подхода к созданию требований, основанный на поведении системы, а также способ последующей автоматизации тестировании, основанной на этих требованиях.

Тестирование при помощи BDD имеет некоторые существенные преимущества по сравнению с платформами тестирования без BDD. Прежде всего, этапы делают автоматизацию BDD очень модульной и, следовательно, многогранной. Каждый шаг – это самостоятельное действие, похожее на то, как каждый сценарий является самостоятельным поведением. Как только будет определено определение шага, оно может быть повторно использовано любым количеством сценариев. Это имеет решающее значение, поскольку большинство действий для функции имеют общие действия. И все шаги по своей сути самодокументированы, поскольку они написаны простым языком.

Теги позволяют легко выбирать тесты для запуска, особенно из командной строки. Неудачные тесты также очень информативны. Драйвер точно определяет, какой шаг не удался и для какого сценария. И так как поведение изолировано, неудача для одного сценария менее подвержена влиянию на другие тестовые сценарии, чем это было бы при проверке процедур.

Проверками в стиле BDD можно покрыть все возможные сценарии, но полностью автоматизировать эти проверки нецелесообразно из-за сложности реализации, например переходов в сторонние приложения

(камера, вебью), проверки без интернета. После покрытия всех основных функций тестами можно проводить регрессионное тестирование приложения, что вручную делать очень затратно. Стоит отметить, что немало времени на старте приложения уходит на подготовку автоматизации, и если бы проект длился всего несколько месяцев, то затраты были бы бессмысленны, поэтому логичнее применять такие методы для достаточно крупных проектов.

Методология BDD является эффективным подходом благодаря двойному применению написанных тестов – такие тесты можно использовать как для прямых целей автоматизации, так и для целей обычного ручного тестирования.

Список литературы

1. A Short History of Software Quality [Электронный ресурс]. – Режим доступа : <http://qualityintelligence.net/articles/a-short-history-of-software-quality>
2. Introducing BDD [Электронный ресурс]. – Режим доступа : <https://dannorth.net/introducing-bdd/>
3. The 4 Basics of Calabash Steps and Step Definitions – How to Implement Those [Электронный ресурс]. – Режим доступа : <https://bitbar.com/the-basics-of-calabash-steps-and-step-definitions/>
4. Reusing Test Code [Электронный ресурс]. – Режим доступа : <https://codeception.com/docs/06-ReusingTestCode>

УСТОЙЧИВОСТЬ СИСТЕМ РАВНОМЕРНЫХ СДВИГОВ, ПОРОЖДЕННЫХ КОМПЛЕКСАМИ ИЗ ФУНКЦИЙ ГАУССА

Е. Г. Супонев
Студент
Е. А. Киселев
Ассистент

Введение

В настоящее время цифровые технологии широко внедряются в область биомедицинских исследований. Важным направлением является поиск эффективных моделей для биомедицинских сигналов. Достаточно хороших результатов позволяет добиться использование с этой целью функций Гаусса. Однако, как нами было установлено в предыдущих исследованиях [1], при проведении расчетов с реальными оцифрованными

сигналами электрокардиограмм (ЭКГ) это сопровождается рядом вычислительных проблем. Одной из них является то, что для надежной аппроксимации пиков ЭКГ, требуется большое количество функций Гаусса. В то же время, системы уравнений для нахождения оптимальных параметров этих функций, заметно усложняются с увеличением их количества. Также при этом теряется вычислительная устойчивость. Теоретическое объяснение этого эффекта дано в статье [2] с помощью понятия констант Рисса.

В данной работе, опираясь на подход, аналогичный статье [2], нами предложен способ повышения вычислительной устойчивости процедуры разложения по функциям Гаусса. Основной идеей является разложение сигнала не по отдельным функциям Гаусса, а сразу по некоторым их комбинациям.

1. Система Рисса

Важным критерием устойчивости неортогональных систем функций служат константы Рисса [3].

Определение. Функции $\varphi_k(x) \in L_2(R), k \in Z$ образуют систему Рисса с положительными константами A и B , если для любой последовательности коэффициентов $\{c_k\} \in l_2$ выполнена следующая двусторонняя оценка

$$A\|c\|_2^2 \leq \left\| \sum_{k=-\infty}^{\infty} c_k \varphi_k(x) \right\|_{L_2}^2 \leq B\|c\|_2^2, \quad (1)$$

где нормы задаются обычным образом:

$$\|c\|_2^2 = \sum_{k=-\infty}^{\infty} |c_k|^2, \|f\|_{L_2}^2 = \int_{-\infty}^{\infty} |f(x)|^2 dx. \quad (2)$$

Под A и B обычно понимаются их наилучшие оценки. Тогда наибольшее значение A называют нижней константой Рисса, а наименьшую величину B верхней константой Рисса. Для ортонормированных систем A и B равны 1.

Для конечномерных систем функций константы Рисса A и B представляют собой минимальное и максимальное собственные значения матрицы Грама соответственно [4]. Если их отношение B/A слишком велико, то матрица называется плохо обусловленной. Если входные данные имеют некоторую погрешность, то погрешность в коэффициентах разложения по данной системе функций будет в среднем в B/A раз больше. В этом случае требуются специальные приемы для обеспечения устойчивости вычислений [5].

Отметим также, что для систем Рисса возможна устойчивая ортогонализация и построение биортогональной системы, что важно для создания алгоритмов разложения по данной системе функций [3].

Дискретный экспериментальный сигнал регистрируется, как правило, с некоторым постоянным шагом. Приняв его в качестве единицы измерения, мы получим целочисленную сетку отсчетов. Поэтому достаточно популярными являются системы целочисленных сдвигов одной функции, в частности функции Гаусса. Для таких систем справедливо следующее утверждение [3].

Теорема. Для того, чтобы семейство функций $\varphi(x-k), k \in Z$ являлось системой Рисса с константами A и B , необходимо и достаточно, чтобы для почти всех $\xi \in R$ выполнялось соотношение

$$A \leq 2\pi \sum_{k=-\infty}^{\infty} \left| \hat{\varphi}(\xi + 2\pi k) \right|^2 \leq B, \quad (3)$$

где $\hat{\varphi}(\xi)$ означает преобразование Фурье

$$\hat{\varphi}(\xi) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \varphi(x) e^{-i\xi x} dx. \quad (4)$$

Обозначим

$$P(\xi) = 2\pi \sum_{k=-\infty}^{\infty} \left| \hat{\varphi}(\xi + 2\pi k) \right|^2. \quad (5)$$

Функция $P(\xi)$ является 2π -периодической. Если она окажется непрерывной, то константы Рисса могут быть найдены как максимум и минимум данной функции на отрезке $[0, 2\pi]$.

2. Системы сдвигов одной и нескольких функций Гаусса

В случае целочисленных сдвигов одной функции Гаусса

$$\varphi(x) = \exp\left(-\frac{x^2}{2\sigma^2}\right) \quad (6)$$

формулы для констант Рисса были получены в статье М. В. Журавлева [2]. Они зависят от параметра σ , причем нижняя константа очень быстро убывает с ростом σ , что ведет к росту отношения B/A .

В данной работе мы рассмотрим в качестве функции $\varphi(x)$, порождающей систему сдвигов, комбинацию двух смещенных друг относительно друга функций Гаусса

$$\varphi(x) = C_1 e^{-\frac{(x+k_1)^2}{2\sigma^2}} + C_2 e^{-\frac{(x+k_2)^2}{2\sigma^2}}, \quad (7)$$

где k_1, k_2 – некоторые целые числа, C_1, C_2 – вещественные параметры.

Для нахождения констант Рисса воспользуемся формулой (3). Фурье образ функции (7) имеет вид

$$\hat{\varphi}(\xi) = \sigma \left(C_1 e^{ik_1 \xi} + C_2 e^{ik_2 \xi} \right) \cdot e^{-\frac{\sigma^2 \xi^2}{2}}, \quad (8)$$

Подставив (8) в (5), получим следующее выражение

$$P(\xi) = 2\pi \sum_{k=-\infty}^{\infty} \sigma^2 \left| C_1 e^{ik_1(\xi+2\pi k)} + C_2 e^{ik_2(\xi+2\pi k)} \right|^2 e^{-\sigma^2(\xi+2\pi k)^2}. \quad (9)$$

Воспользуемся тем, что $e^{i2\pi k k_1} = e^{i2\pi k k_2} = 1$. Тогда, после вынесения общего множителя в (9), приходим к соотношению

$$P(\xi) = 2\pi \sigma^2 \left| C_1 e^{ik_1 \xi} + C_2 e^{ik_2 \xi} \right|^2 \sum_{k=-\infty}^{\infty} e^{-\sigma^2(\xi+2\pi k)^2}. \quad (10)$$

Если в формуле (10) положить $C_1 = 0, C_2 = 1, k_2 = 0$, то получим выражение для случая сдвигов одной функции Гаусса, которое, согласно статье [2], имеет вид

$$P(\xi) = \sigma \sqrt{\pi} \theta_3 \left(\frac{\xi}{2}, q \right), q = \exp \left(-\frac{1}{4\sigma^2} \right), \quad (11)$$

где $\theta_3(x, q)$ – третья тета-функция Якоби

$$\theta_3(x, q) = \sum_{k=-\infty}^{\infty} q^{k^2} e^{2ikx}. \quad (12)$$

Пользуясь этим, формулу (10) можем переписать в виде

$$P(\xi) = \sigma \sqrt{\pi} \left| C_1 e^{ik_1 \xi} + C_2 e^{ik_2 \xi} \right|^2 \theta_3 \left(\frac{\xi}{2}, q \right), \quad q = \exp \left(-\frac{1}{4\sigma^2} \right). \quad (13)$$

Поскольку $P(\xi)$ получилась непрерывной, в силу непрерывности тета-функции Якоби, для нахождения констант Рисса, теперь необходимо найти максимальное и минимальное значение функции (13) на отрезке $[0, 2\pi]$.

3. Особенности вычисления констант Рисса

Для семейства сдвигов одной функции Гаусса, значения констант Рисса при разных значениях параметра σ были рассчитаны в статье М. В. Журавлева [2]. Теперь рассмотрим случай двух функций Гаусса. Как уже говорилось, константы могут быть найдены как максимум и минимум функции (13) на отрезке $[0, 2\pi]$. Введем следующие обозначения

$$M(\xi) = \left| C_1 e^{ik_1 \xi} + C_2 e^{ik_2 \xi} \right|^2, N(\xi) = \sigma \sqrt{\pi} \theta_3 \left(\frac{\xi}{2}, q \right). \quad (14)$$

Экспериментальным путем численно было выяснено, что наименьшее отношение B/A при $\sigma \geq 1$ достигается, если взять два гауссиана $k_1=0, k_2=1$ с коэффициентами $C_1=1, C_2=-0,9$. В этом случае максимум $M(\xi)$ функции приходится на минимум $N(\xi)$ в точке $\xi = \pi$. Нижняя константа Рисса, равная минимальному значению $P(\xi)$, становится за счет этого эффекта несколько больше. В то же время, минимум $M(\xi)$ в точке $\xi = 0$ совпадает с максимумом $N(\xi)$. Верхняя константа становится от этого несколько меньше. В итоге отношение констант Рисса становится меньше, чем в случае, когда систему сдвигов порождает одна функция Гаусса.

В таблице ниже приведены результаты расчетов констант Рисса и их отношения для системы целочисленных сдвигов комбинации двух гауссианов и сдвигов одной функции Гаусса при разных параметрах σ ($k_1=0, k_2=1, C_1=1, C_2=-0,9$).

Таблица

Значения констант Рисса и их отношения

σ	Две функции Гаусса			Одна функция Гаусса		
	A	B	B/A	A	B	B/A
1	2.3×10^{-3}	1.9	8.3×10^{-2}	6.4×10^{-4}	6.2	9.7×10^{-3}
2	1.2×10^{-15}	2.1	1.6×10^{15}	3.5×10^{-16}	25	7.0×10^{16}
3	1.0×10^{-36}	2.2	2.1×10^{36}	2.9×10^{-37}	57	1.9×10^{38}

Как видим из таблицы, за счет использования комплексов из двух функций Гаусса, удается понизить отношение констант Рисса в 10 и более раз. Таким образом, выгоднее проводить разложение исследуемого сигнала не по отдельным функциям Гаусса, а сразу по их комплексам.

Заключение

В данной работе с помощью констант Рисса мы провели исследование устойчивости семейств равномерных сдвигов, порожденных комплексами из нескольких функций Гаусса. Было показано что, использование комплексов из двух гауссианов позволяет в среднем снизить отношение констант Рисса в 10 раз по сравнению со случаем одной функции Гаусса, следовательно, повысить вычислительную устойчивость исследуемой системы функций.

Отметим, что нами проводились аналогичные расчеты и с комплексами, состоящими из трех функций Гаусса. Заметного снижения отношения констант Рисса в этом случае, однако, добиться не удалось. Использование большего количества функций Гаусса приводит к весьма сложным вычислениям и будет являться темой будущих исследований. Также перспективным представляется изучение комбинаций гауссианов, имеющих разные значения параметра σ .

Список литературы

1. Киселев Е. А. Комбинированный алгоритм сжатия сигнала электрокардиограммы с помощью всплесков Добеши и функции Гаусса / Е. А. Киселев, Н. М. Насер, Е. Г. Супонев // Системы управления и информационные технологии. – 2017. – Т. 69. – № 3. – С. 53-56.
2. Журавлев М. В. О константах Рисса для систем целочисленных сдвигов функции Гаусса / М. В. Журавлев // Научные ведомости Белгородского государственного университета. – 2011. – № 5(100). – Вып. 22. – С. 39-46.
3. Чуи Ч. Введение в вэйвлеты / Ч. Чуи. – М. : Мир, 2001. – 412 с.
4. Кострикин А. И. Линейная алгебра и геометрия / А. И. Кострикин, Ю. И. Манин. – М. : Наука, 1986. – 304 с.
5. Бахвалов Н. С. Численные методы / Н. С. Бахвалов, Н. П. Жидков, Г. М. Кобельков. – М. : Наука, 1987. – 598 с.

КЛАССИФИКАЦИЯ ЭЛЕМЕНТОВ ЗЕРНОВЫХ СМЕСЕЙ МЕТОДАМИ ГЛУБОКОГО ОБУЧЕНИЯ

Д. М. Суходолов

Студент

А. А. Крыловецкий

Доцент

Введение

Численность населения Земли в настоящее время значительно увеличивается год от года [1]. Таким образом, одна из наиболее актуальных проблем современного общества – обеспечение количества продовольственных товаров, необходимого для комфортного существования обитателей планеты. Для успешного выполнения данной задачи необходимо использовать не только экстенсивный, но и интенсивный подход в развитии аграрного хозяйства. Иными словами, нужно не только увеличивать объёмы посевов, так как количество плодородных земель не безгранично, но и улучшать качество техники, обслуживающей сопутствующие процессы посевов, сбора и хранения.

В виду того, что на том или ином этапе контроль чистоты зерновой смеси традиционными способами из-за объёма исследуемых семян является трудоёмкой и затратной задачей, в настоящее время часто используются подходы, использующие результаты применения методов

компьютерного зрения для процессов классификации и распознавания. Тем не менее, не смотря на то, что данные методы порою являются довольно эффективными, данная сфера до сих пор является не до конца изученной и в этой сфере производится множество исследований [2].

1. Методика эксперимента

Одной из лучших библиотек, использующихся для глубокого обучения, является TensorFlow. Это библиотека с открытым исходным кодом для численного расчёта с использованием графа потока данных, разработанная Google. Узлы на графе представляют собой математические операции, в то время как ребра графа представляют собой многомерные массивы данных (тензоры), передаваемые между ними. По своему функционалу и возможностям TensorFlow заслуженно считается очень мощным инструментом для разработки, однако является довольно сложной для использования библиотекой и требует длительного предварительного изучения своих особенностей и архитектуры. Чтобы облегчить её использование для глубокого обучения была разработана библиотека Keras.

Keras не занимается низкоуровневыми операциями, такими как работа с тензорами, свёртка и т.д. Вместо этого используется библиотека, которая выступает в качестве бекэнда: TensorFlow, Theano или CNTK. Тип модели в Keras называется последовательностью, которая представляет собой стек слоёв. После обучения можно использовать полученную модель для прогнозирования результатов на новых данных.

VGG16 – это сеть, разработанная Visual Geometry Group для распознавания объектов на изображениях, состоящая из 16 слоёв [3]. На вход данная сеть получает RGB изображение и выдаёт на выходе в формате one hot encoding данные о том, к какому классу относится объект на изображении. Она содержит два основных блока. Первый выделяет характерные признаки в изображении и состоит из чередующихся каскадов свёртки и подвыборки. Сначала идут два каскада свёртка-свёртка-подвыборка, а затем три каскада свёртка-свёртка-свёртка-подвыборка. При подвыборке производится выбор Max Pooling (максимального значения). Другой блок выполняет задачу классифицирования объекта на изображении в соответствии с данными, полученными на предыдущем шаге. В первых двух полносвязных слоях находится 4096 нейронов, а в третьем – 1000 нейронов, что соответствует количеству классов объектов.

VGG19 – другая сеть от Visual Geometry Group для распознавания объектов, содержащая уже 19 слоёв, но не сильно отличающаяся от своей предшественницы в плане архитектуры, что можно видеть на рис. 1. Сеть MobileNet [4] имеет более отличающуюся структуру, что позволяет получать с помощью неё выходные файлы обученных сетей меньшего

размера (приблизительно 50 МБ против 117 МБ и 174 МБ у VGG16 и VGG19 соответственно), которые проще использовать во встраиваемых системах и ПО для смартфонов.

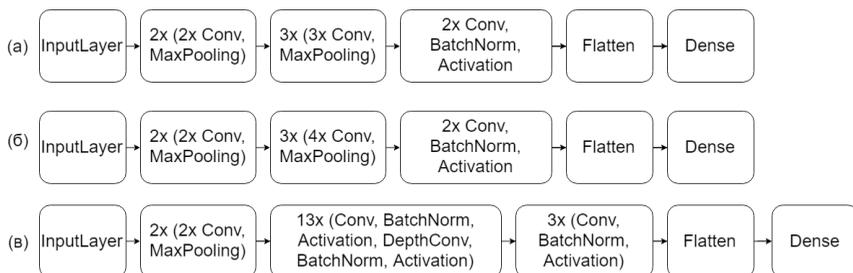


Рис. 1. Архитектура сетей: *a* – VGG16, *б* – VGG19, *в* – MobileNet

В ходе исследования был разработан классификатор семян красного льна, зелёного льна, маша, нуга, подсолнечника, пшеницы обычной, пшеницы красной, пшеницы стекловидной, свёклы, чечевицы, ячменя и кедра. Часть изображений, рассмотренных в исследовании, можно наблюдать на рис. 2.

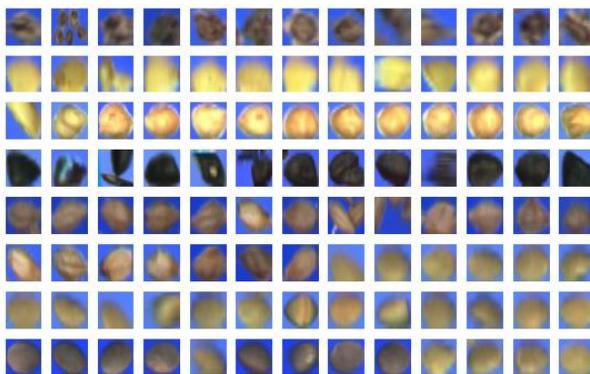


Рис. 2. Визуализация части рассмотренных в ходе эксперимента данных

Представим данные о выборке в виде таблицы (табл. 1). Ключевая сложность задачи классификации в данном случае заключается в том, что изображения имеют малый размер, который равняется в среднем 20×10 , а в наихудшем в выборке случае 6×6 пикселей. Таким образом, исходные изображения несут в себе ограниченный объём информации, что усложняет задачу классификации.

Описание исследованной выборки

Название растения	В обучающей выборке, шт	В тестовой выборке, шт
Красный лён	480	119
Зелёный лён	270	52
Маш	248	45
Нут	112	28
Подсолнечник	95	23
Пшеница обычная	488	122
Пшеница красная	362	72
Пшеница стекловидная	284	58
Свёкла	398	56
Чечевица	415	73
Ячмень	213	42
Кедр	131	32

Программный модуль был написан на языке программирования Python в среде разработки Jupyter Notebook с применением таких библиотек, как Tensorflow, Keras, CUDA, Pandas и ряда других. В качестве моделей при обучении использовались VGG16, VGG19 и MobileNet.

В ходе исследований использовался ПК с 12 ГБ оперативной памяти, процессором Intel Core i3-4030U и видеокартой NVIDIA GeForce 840M. Были установлены библиотеки Python 3.6.3, Anaconda 5.0.1, CUDA Toolkit 8.0, cuDNN v6.0, TensorFlow 1.4.0 и Keras версии 2.1.2. Обучение производилось в течение 20 эпох (epochs).

2. Полученные результаты

При использовании TensorFlow с GPU, базирующимся на технологии CUDA, на обучение в среднем для разных моделей было затрачено 41,9833 минут. В случае выполнения обучения без использования CUDA, в режиме CPU, данный процесс обучения занимает около 13 часов. Поэтому необходимо учитывать, что дальнейшие вычисления производились в режиме GPU и для режима CPU не рассматривались.

При учёте того, что 3146 экземпляров объектов использовалось для обучения и 350 для валидации, то на обучение для каждого отдельного изображения в среднем для разных моделей было затрачено 0,8009 секунд. На выполнение классификации в среднем было затрачено 25,6669 секунд для 722 изображений, в том числе на каждое отдельное изображение 0,0355 секунды машинного времени. В ходе исследований получилось в среднем классифицировать тестовую выборку с помощью моделей, обученных на обучающей выборке, с точностью 94,6352%.

Среди отдельных моделей были выявлены следующие закономерности. Самую высокую точность классификации удалось получить VGG19 (95,0692%) против VGG16 (94,1274%) и MobileNet (94,7091%).

При этом MobileNet имеет наилучшие показатели по общему времени обучения и времени классификации (1808,2264 секунд и 13,0165 секунд соответственно), против VGG16 (2509,5991 и 34,9570 секунд) и VGG19 (3241,6330 и 29,0271 секунд соответственно). Количество верно распознанных в ходе исследования изображений для различных моделей в процентном соотношении представлено в табл. 2 и на рис. 3.

Таблица 2

*Процент верно классифицированных семян
для рассматриваемых нейросетей*

Название растения	Верно распознанных VGG16, %	Верно распознанных MobileNet, %	Верно распознанных VGG19, %
Красный лён	93.2773	97.6470	93.6134
Зелёный лён	80.3846	78.8461	88.4615
Маш	100.0000	97.7777	100.0000
Нут	100.0000	100.0000	99.2857
Подсолнечник	93.0434	93.9130	97.3913
Пшеница обычная	98.5245	97.3770	98.0327
Пшеница красная	83.8888	88.8888	88.3333
Пшеница стекловидная	95.5172	90.6896	96.8965
Свёкла	93.2142	97.5000	95.3571
Чечевица	99.1780	100.0000	99.4520
Ячмень	94.7619	90.4761	88.5714
Кедр	100.0000	100.0000	97.5000

Заключение

Опыты, произведённые с изображениями семян и нейросетями, показывают, что оптимальным решением является использование MobileNet. Она требует меньше времени на обучение, меньше времени на классификацию (что более важно) и даже сам обученный файл требует в несколько раз меньшее количество места в файловой системе, чем у конкурентов. Таким образом, MobileNet удобнее для использования во встраиваемых системах. Конечно, VGG19 показал несколько большую точность классификации, но это различие не столь велико (95.0692% против 94.7091%) и при учёте других преимуществ MobileNet представляется более перспективной для использования в целях классификации семян растений.

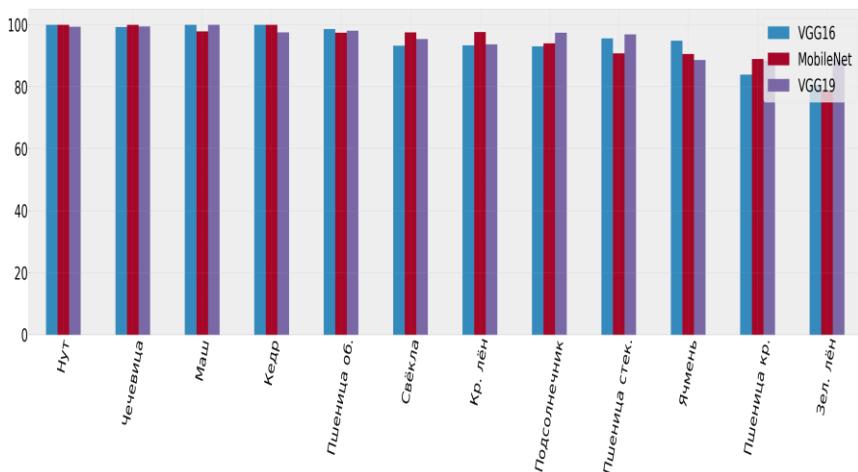


Рис. 3. Визуализация верно классифицированных зёрен в процентном соотношении для различных моделей

Список литературы

1. World Population Prospects. The 2017 Revision [Electronic resource] : United Nations. – Access mode : https://esa.un.org/unpd/wpp/Publications/Files/WPP2017_KeyFindings.pdf
2. Методы классификации элементов зерновых смесей на основе анализа спектральных характеристик в видимом и инфракрасном диапазоне длин волн / А. А. Сирота [и др.] // Вестник Воронежского Государственного Университета. Серия : Системный анализ и информационные технологии. – 2016. – № 1. – С. 150-160.
3. Simonyan K. Very deep convolutional networks for large-scale image recognition [Electronic resource] / K. Simonyan, A. Zisserman. – 2014. – Access mode : <https://arxiv.org/pdf/1409.1556/>
4. Howard A. G. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications [Electronic resource] / A. G. Howard, M. Zhu, B. Chen and others. – 2017. – Access mode : <https://arxiv.org/pdf/1704.04861.pdf>

СОЗДАНИЕ АЛЬТИМЕТРА

В. А. Ушаков

Студент

Д. П. Донских

Студент

М. А. Теплякова

Студент

С. А. Зуев

Доцент

Введение

В повседневной практике нам бывает довольно часто нужна информация о высоте. Для ее измерения существуют различные приборы с различным устройством. Общее их название – альтиметр. Альтиметр – прибор, предназначенный для измерения высоты. В случае пилотируемого летательного аппарата, высотомер является пилотажно-навигационным прибором указывающим высоту полёта. По принципу устройства высотомеры делятся на барометрические, радиотехнические, инерциальные, ионизационные и прочие. Интерес для нашего проекта представляют барометрические альтиметры [1-7].

1. Постановка задачи

Разработать барометрический альтиметр, обладающий низкой ценой, работа которого основана на современных электронных датчиках и информацию с которого легко можно передавать на различные электронно-вычислительные устройства.

Принцип действия барометрического высотомера основан на измерении давления атмосферы.

С увеличением высоты уменьшается и действующее атмосферное давление. Данный принцип положен в основу прибора, который на самом деле измеряет не высоту, а давление воздуха [1].

Известно, что с увеличением высоты H уменьшается абсолютное атмосферное давление P . Так как это давление для одного и того же момента времени связано с высотой однозначной зависимостью, то, измерив на некоторой высоте абсолютное давление, можно определить высоту точки измерения [2].

Барометрический метод основан на закономерном изменении атмосферного давления с высотой. Для нахождения этой закономерности в

атмосфере выделяется вертикальный столб воздуха постоянного сечения F (рис. 1). Давление воздуха у Земли обозначается через P_0 , а на высоте H через P_H . При изменении высоты на dH атмосферное давление уменьшится на величину dP , равную весу dQ элементарного объема воздуха dV , поделенному на площадь его основания F [4].

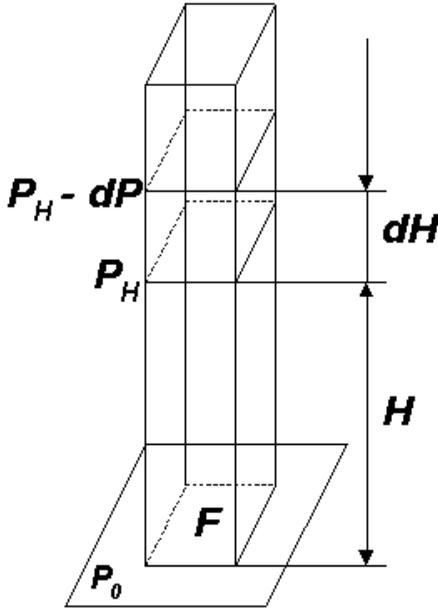


Рис. 1. Изменение давления с высотой

$$\frac{dP}{P} = -\frac{dH}{RT} \quad (1)$$

В это уравнение входит абсолютная температура воздуха, изменяющаяся с высотой. Закон ее изменения неодинаков для тропосферы и стратосферы (рис. 2). Это уравнение решается для каждого слоя отдельно [5].

Известно, что температура воздуха в тропосфере, то есть до высоты 11000 м, изменяется примерно по линейному закону, а в стратосфере до высоты 33000 м остается постоянной [6].

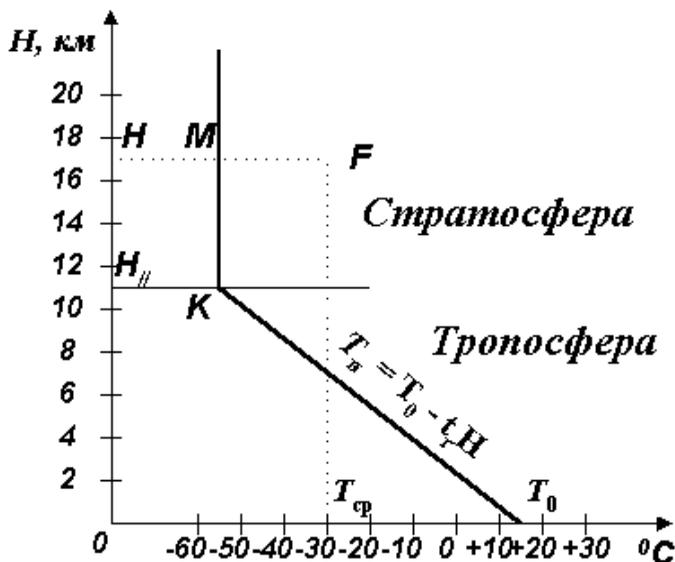


Рис. 2. Изменение температуры воздуха в атмосфере

Формула для высот от 0 до 11000 м [7]:

$$H = \left[1 - \left(\frac{P_H}{P_0} \right)^{t_r R} \right] \cdot \frac{T_0}{t_r} \quad (2)$$

где T_0 — абсолютная температура воздуха у Земли; t_r — вертикальный температурный градиент, град/м; H — высота м., P_0 — Давление воздуха у Земли, $R = 29,27$ м/град — газовая постоянная сухого воздуха.

2. Контроллер

Основной альтиметра стал контроллер Arduino Mini (рис. 3) [8]. Он обладает достаточно малыми габаритами и широким спектром возможностей. Платформа содержит 14 цифровых входов и выходов, 6 аналоговых входов, резонатор, кнопку перезагрузки и отверстия для монтажа выводов. Кроме того, контроллер Arduino содержит богатую общедоступную бесплатную библиотеку на языке C, а низкая стоимость китайских клонов позволяет приобрести данные контроллеры даже тем студентам, которые не получают стипендию. Блок из шести выводов может подключаться к кабелю FTDI или плате-конвертеру Sparkfun для обеспечения питания и связи через USB.

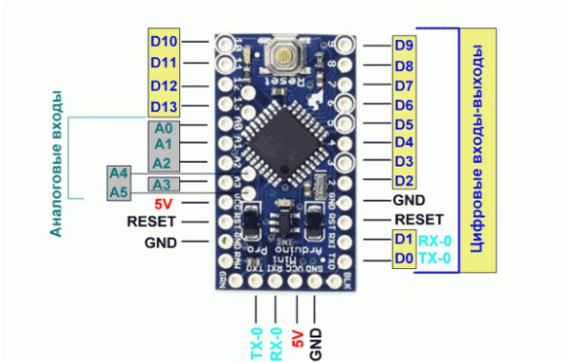


Рис. 3. Arduino Mini

К контроллеру подключён датчик BMP 280 (рис. 4), который представляет собой цифровой датчик абсолютного барометрического давления от Bosch Sensortec [9], [10]. Этот датчик разработан специально для мобильных приложений, где малые габариты и низкое энергопотребление очень важно. На основе этого данного датчика мы разработаем свою модель альтиметра. Именно датчик давления BMP 280 может быть использован его в наших различных приложениях, включая приложения с Arduino [11].

BMP 280 ultrasonic sensor module

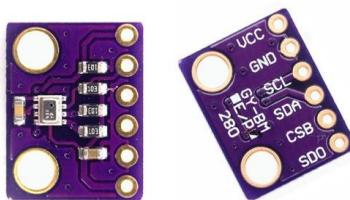


Рис. 4. BMP 280

3. Вывод информации

Для вывода информации решено было использовать lcd 1602 с платой расширения Keypad Shield для возможности взаимодействия пользователя с метеостанцией. LCD дисплеи размерности 1602, на базе контроллера HD44780, являются одними из самых простых, доступных и востребованных дисплеев для разработки различных электронных устройств. На дисплее имеется 16pin разъем для подключения. Схема подключения датчика, экрана и контроллера представлена ниже (рис. 5).

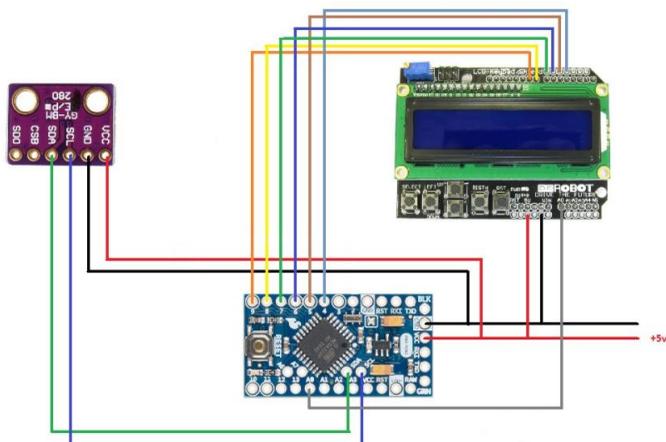


Рис. 5. Схема подключения

4. Программирование

Интерфейс включает в себя вывод двух значений: верхнее значение показывает результат без поправки, нижнее значение дает информацию об измеренной высоте с учетом поправки (рис. 6). Программное обеспечение включает в себя опрос датчиков, вычисление относительных высот, вывод информации на дисплей.



Рис. 6. Фотография интерфейса

Все используемые подпрограммы (скетчи) расположены по адресу: <https://github.com/SteamDiver/ArduinoMeteo>

5. Калибровка

Баровысотомерам присущи инструментальные, аэродинамические и методические ошибки.

В первую очередь нас интересуют температурные ошибки. Причиной ошибки является несоответствие фактического распределения температуры воздуха с высотой стандартным значениям, принятым в расчете механизма работы высотомера.

Формула для определения температурной поправки, которая дает достаточную точность вычислений для малых высот полета и широко используется на практике:

$$\Delta H_1 = H_{np} \cdot \left(\frac{t_0 - 15}{300} \right) \quad (3)$$

где t_0 - фактическая температура у земли в градусах Цельсия.

В данной формуле за T_{cp} принято 288 К (+15°C), в знаменателе это значение округлено до 300.

Заключение

Нам удалось собрать барометрический альтиметр, обладающий низкой ценой, работа которого основана на современных электрических датчиках и информацию с которого легко можно передавать на различные электронно-вычислительные устройства

Список литературы

1. Оборудование самолетов : учебник / А. П. Волкостов, Э. Г. Паленый. – 2-е изд., перераб. и доп. – М. : Машиностроение, 1980. – 229 с.
2. Кучумова, И.П. Радиооборудование самолетов Ту-134 и Ту-134А и его летная эксплуатация : учеб. пособие для школ высш. лет. подгот. и учеб.-тренировоч. отрядов гражд. авиации. – Москва : Машиностроение, 1978. – 121 с.
3. Высотомер [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/Высотомер>
4. Барометрический высотомер [Электронный ресурс]. – Режим доступа : <http://poznayka.org/s85598t1.html>
5. Стратостат «СССР-1» [Электронный ресурс]. – Режим доступа : <http://xn--80aa2bkafhg.xn--p1ai/article.php?nid=33863>
6. Барометрические высотомеры [Электронный ресурс]. – Режим доступа : <https://studfiles.net/preview/6810199/page:4/>
7. Электронное учебное пособие «Воздушная навигация и элементы самолетовождения» [Электронный ресурс]. – Режим доступа : <http://liric.narod.ru/>

8. Arduino Pro Mini [Электронный ресурс. – Режим доступа: <http://arduino.ru/Hardware/ArduinoBoardProMini>

9. BME280 – датчик давления, температуры и влажности [Электронный ресурс]. – Режим доступа: <https://micro-pi.ru/bme280-давления-температуры-влажности/>

10. Метеодатчик BME280 – подключение к Arduino [Электронный ресурс]. – Режим доступа : <https://volti.ru/bme280-and-arduino/>

11. Датчик давления BMP-280 с Arduino учебник [Электронный ресурс. – Режим доступа : <https://iknowvations.in/ru/Arduino/pressure-sensor-bmp-280-arduino-tutorial/>

СОЗДАНИЕ ДОМАШНЕЙ МЕТЕОСТАНЦИИ С НИЗКОЙ СЕБЕСТОИМОСТЬЮ

В. А. Ушаков

Студент

Д. П. Донских

Студент

М. А. Теплякова

Студент

С. А. Зуев

Доцент

Введение

В обыденной жизни мы часто сталкиваемся с необходимостью получать сведения об атмосферном давлении или о температуре воздуха. Для сбора и анализа информации о различных параметрах, отражающих состояние окружающей среды, необходимо множество датчиков. Данные датчики должны иметь низкую себестоимость, ввиду использования их в большом количестве. Также они должны предоставлять всю необходимую информацию об окружающей среде и иметь задел для последующей модернизации. Необходимо рассмотреть несколько вариантов датчиков для измерения температуры окружающей среды, давления и влажности воздуха. Необходимо получить информацию о возможных вариантах подключения, программной составляющей и особенностях работы. Следующим шагом будет рассмотрение различных вариантов отображения информации, включая варианты с возможностью взаимодействия

пользователя с интерфейсом. Также необходимо рассмотреть различные варианты питания устройства.

1. Постановка задачи

Необходимо создать компактную метеостанцию, способную получать достаточное количество информации об окружающей среде. Данная метеостанция должна обладать низкой себестоимостью, малой потребляемой мощностью и иметь возможность расширения своего функционала для последующей модернизации.

2. Анализ данных, которые должен обрабатывать модуль

Проектируемая метеостанция должна будет работать со следующими параметрами:

1. Температура воздуха
2. Давление
3. Влажность воздуха

Рассмотрим подробнее каждый из параметров.

2.1 Температура воздуха

В большинстве стран, в том числе и в России, при измерении температуры результат полученных измерений выражают в градусах по шкале Цельсия ($^{\circ}\text{C}$). Ноль этой шкалы приходится на температуру, при которой тает лёд, а $+100^{\circ}\text{C}$ – на температуру кипения воды [1].

Кроме того, в теоретической метеорологии применяется абсолютная шкала температур (шкала Кельвина), К. Ноль этой шкалы соответствует полному прекращению теплового движения молекул, то есть самой низкой возможной температуре. По шкале Цельсия это $-273,15^{\circ}\text{C}$, но на практике это значение округляют до -273°C .

В своей работе мы будем пользоваться исключительно шкалой Кельвина, так как она даёт непосредственное представление о том, в каком агрегатном состоянии находится в данный момент вода, что, несомненно, важно для пользователя метеостанции.

2.2 Давление

Давление – физическая величина, которая определяется как отношение нормальной составляющей силы, действующей на малый элемент поверхности, к его площади. Для обозначения давления обычно используется символ P – от лат. *pressūra* (давление). В Международной системе единиц (СИ) измеряется в паскалях (русское обозначение: Па; международное: Pa) [2]. Паскаль равен давлению, вызываемому силой, равной одному ньютону, равномерно распределённой по нормальной к ней поверхности площадью один квадратный метр.

Наряду с паскалем в Российской Федерации допущены к использованию в качестве внесистемных единиц измерения давления следующие единицы:

- бар;
- килограмм-сила на квадратный сантиметр;
- атмосфера техническая;
- миллиметр ртутного столба

Данная метеостанция выводит измеренное атмосферное давление в миллиметрах ртутного столба, так как именно эта единица измерения наиболее распространена в метеорологии.

2.3 Влажность воздуха

Рассмотрим понятие абсолютной влажности воздуха. Абсолютная влажность воздуха – количество влаги, содержащейся в одном кубическом метре воздуха. Абсолютная влажность используется тогда, когда необходимо сравнить количество воды в воздухе при разных температурах или в большом диапазоне температур. Обычно измеряют в г/м³. Но в связи с тем, что при определённой температуре воздуха в нём максимально может содержаться только определённое количество влаги (с увеличением температуры это максимально возможное количество влаги увеличивается, с уменьшением температуры воздуха максимальное возможное количество влаги уменьшается), ввели понятие относительной влажности.

Относительная влажность воздуха — отношение массовой доли водяного пара в воздухе к максимально возможной при данной температуре [3]. Измеряется в процентах и определяется по формуле:

$$RH = \frac{P_{(H_2O)}^*}{P_{(H_2O)}|_{\max}} \cdot 100, \quad (1)$$

где: RH — относительная влажность рассматриваемой смеси (воздуха); $P_{(H_2O)}$ — плотность водяного пара (абсолютная влажность); $p_{(H_2O)}^*$ — плотность насыщенного водяного пара при данной температуре.

3. Контроллеры и датчики

Для работы с датчиками нам потребуется контроллер. Изначально тесты проводились на контроллере Arduino Uno, однако затем было принято решение перейти на контроллер Arduino Mini (рис. 1), т.к. он имеет меньшие габариты и для создания данного проекта контроллер Arduino Uno обладал излишним количеством входов [4]. Платформа содержит 14 цифровых входов и выходов, 6 аналоговых входов, резонатор, кнопку перезагрузки и отверстия для монтажа выводов. Блок из шести выводов может подключаться к кабелю FTDI или плате-конвертеру Sparkfun для обеспечения питания и связи через USB.

Компактный корпус устройства имеет высокую ударопрочность. К сожалению, датчик не способен измерять давление.

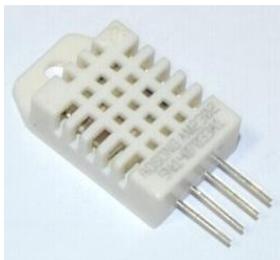


Рис. 3. Датчик am2302

4. Вывод информации

Для вывода информации решено было использовать lcd 1602 с платой расширения Keypad Shield для возможности взаимодействия пользователя с метеостанцией. LCD дисплеи размерности 1602, на базе контроллера HD44780, являются одними из самых простых, доступных и востребованных дисплеев для разработки различных электронных устройств. На дисплее имеется 16pin разъем для подключения.

Модуль lcd 1602 Keypad Shield включает в себя LCD дисплей (16x2) и шесть кнопок. Для взаимодействия контроллера Arduino и модуля, используются цифровые выводы 4, 5, 6, 7, 8, 9, 10 а для чтения состояния кнопок используется один аналоговый вывод.

Датчик BME 280 решено было подключить по интерфейсу I2C, т.к. он позволяет использовать всего 2 проводника для подключения (рис. 4) [6].

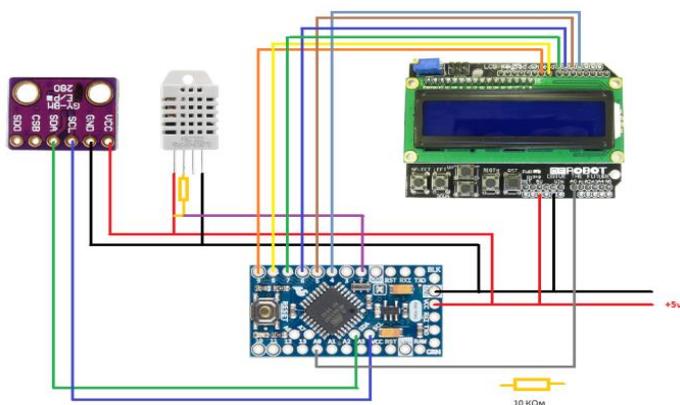


Рис. 4. Схема подключения

5. Программирование

Для программной части метеостанции потребовалось несколько сторонних библиотек [7]:

- SparkFunBME280 для работы с BME280;
- LiquidCrystal для работы с lcd1602;
- DHT для работы с m2302;
- Wire и Spi для работы с I2C интерфейсом.

Интерфейс устройства представляет 2 переключаемых страницы. На первой представлена информация с датчика BME280, на второй с датчика m2302 (рис. 5). Управление переключением страниц – автоматическое.

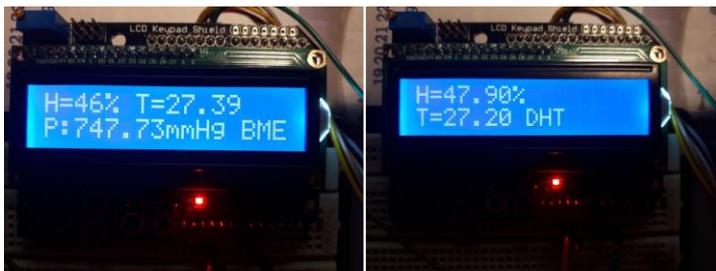


Рис. 5. Интерфейс устройства

Все скетчи хранятся по адресу:

<https://github.com/SteamDiver/ArduinoMeteo>, а также [8], [9].

Для проверки использовались эталонные приборы. Изначально датчик опускался в тающий лед, засекалась погрешность Δ_1 , затем в кипящую воду и засекалась погрешность Δ_2 . Затем вычисляли Δ , равное среднему между Δ_1 и Δ_2 . На величину Δ делали для датчика сдвиг нуля [9].

Заключение

Нам удалось создать компактную метеостанцию, позволяющую получать основную информацию о состоянии окружающей среды. При этом цена устройства достаточно низкая, а погрешность измерений сведена к минимуму. Также датчик обладает внушительным потенциалом для модернизации.

Список литературы

1. Температура [Электронный ресурс]. – Режим доступа : <https://ru.wikipedia.org/wiki/Температура>
2. Влажность [Электронный ресурс]. – Режим доступа : <https://ru.wikipedia.org/wiki/Влажность>
3. Давление [Электронный ресурс]. – Режим доступа : <https://ru.wikipedia.org/wiki/Давление>

4. Arduino Pro Mini [Электронный ресурс]. – Режим доступа : <http://arduino.ru/Hardware/ArduinoBoardUno>
5. BME280 – датчик давления, температуры и влажности [Электронный ресурс]. – Режим доступа : <https://micro-pi.ru/bme280-давления-температуры-влажности>
6. Метеодатчик BME280 – подключение к Arduino [Электронный ресурс]. — Режим доступа : <https://volti9.ru/bme280-and-arduino>
7. Сомер У. Программирование микроконтроллерных плат Arduino/Freduino / У. Сомер; пер. с нем. – 2-е изд., перераб. и доп. – СПб. : БХВ-Петербург, 2016. – 256 с.
8. Justo Di P. Atmospheric Monitoring with Arduino / P. Di Justo, E. Gertz // Sebastopol : O'Reilly Media, 2012. – 77 p.
9. Петин В.А. Проекты с использованием контроллера Arduino. / А.В. Петин. – СПб. : БХВ-Петербург, 2014. – 400 с.

МНОГОУРОВНЕВЫЙ МАСШТАБИРУЕМЫЙ ГРАФ ДЛЯ ЭЛЕКТРОННЫХ КАРТ

Н. В. Ушков
Магистрант
Н. А. Тюкачев
Доцент

Введение

В настоящее время в программировании наиболее широкое применение среди дисциплин, а также методов дискретной математики находят теории графов, а в особенности алгоритмы на графах. Это связано с тем, что теория графов использует наиболее удобный язык для описания программных моделей.

Язык теории графов дает возможность приобрести наглядность условию задач, что существенно упрощает процесс их анализа. Сами решения, полученные через этот метод, являются более простыми, нежели решения, полученные другими методами. Они не содержат утомительных и нудных вычислений. Из-за этого теория графов широко применяется в современном мире в самых различных областях науки и техники.

Существующая в теории графов система специальных терминов и обозначений позволяет описывать сложные, а порою очень тонкие вещи, достаточно просто и доступно для читателя. Особую важность

представляет собой наличие графической интерпретации самого понятия графа, которая дает наглядность.

Наличие в данной теории графических изображений дает человеку возможность сразу «увидеть» суть дела, опираясь чисто на интуитивный уровень восприятия. Изображения украшают и дополняют текстовые доказательства и утомительные, сложные для восприятия формулы.

Графы по своей сути являются наиболее абстрактной структурой, с которой мы сталкиваемся в теории ЭВМ (computerscience). Они используются как для описания алгоритмов автоматического проектирования, так и при решении задач маршрутизации потоков. Кроме того, любая система, которая предполагает наличие дискретных состояний или узлов, а также переходов между ними может быть описана графом.

Теории графов широко применяется в геоинформационных системах.

Геоинформационные системы используются повсеместно: в доставке продуктов питания, доставке товаров, отслеживании такси по карте, в навигаторах, в системах самолетов и ракет. Они являются независимым инструментом при решении транспортных и географических задач, а также задач имитационного моделирования.

Кроме рядовой справочной информации, карты открывают нам обширные рекламные и функциональные возможности. Так, например, с помощью карт можно просматривать, а также размещать различные данные о местных компаниях. К примеру, данные о адресе, контактной информации и маршруте проезда к компании. А посредством API инструмента, позволяющего разработчикам получать доступ к частям сервиса карт, их можно интегрировать в собственные сайты.

1. Постановка задачи

Основной целью современных геоинформационных систем (ГИС) и систем автоматизированного проектирования (САПР) является построение и анализ графических объектов, а также их взаимосвязей в пространстве и времени. Одним из механизмов выявления и анализа таких связей служит аппарат теории графов, который рассматривает графы, состоящие из элементов двух типов – вершин и ребер между ними.

Существует и достаточно широко используется планарный граф, представляющий собой граф, который нарисован таким образом, что его ребра не пересекаются. Можно говорить, что граф допускает плоскую укладку, в том случае, если его можно нарисовать как плоский, иначе говоря планарный.

Несмотря на частое использование планарного графа, его применение в геоинформационных системах невозможно, так как в них рассматриваются существующие или вновь проектируемые дома, кварталы и прочие сооружения как вершины, а соединяющие их дороги,

инженерные сети, линии электропередачи и т.п. как ребра, а соответственно данные «ребра» будут пересекаться. Поэтому в своей работе мы будем использовать многоуровневый (3D) граф, то есть граф, который в каждом узле может содержать еще один граф.

Цель работы – разработка приложения, в котором пользователь может создавать и использовать масштабируемый многоуровневый граф для электронных карт.

Применение различных вычислений, производимых на таком графе, позволяет, найти кратчайший объездной путь, ближайший продуктовый магазин или спланировать оптимальный маршрут.

В ходе анализа разрабатываемого приложения было выявлено, что оно должно иметь следующие варианты использования (рис. 1).



Рис. 1. Диаграмма use-case

2. Реализация

Для реализации разрабатываемого приложения необходимо усовершенствовать алгоритм Дейкстры для того, чтобы находить кратчайший путь от одного узла до другого, который может находиться на любом из имеющихся уровней.

Разрабатываемое приложение должно быть десктопным, то есть приложением, логика работы которого подразумевает наличие оператора (пользователя), содержащим полную функциональность, а также способное работать отдельно от других приложений.

Для реализации приложения понадобятся такие инструменты как:

- Visual Studio 2017;
- Язык программирования C# [1].

Visual Studio 2017 была выбрана из-за удобства и широкого ее использования для разработки десктопных приложения под операционную систему Windows.

Как было сказано выше, для реализации создаваемого нами приложения необходимо усовершенствовать алгоритм Дейкстры, который имеет следующую структуру:

Шаг 1. Первой вершине присваивается вес равный 0, а всем остальным вершинам - равный бесконечности.

Шаг 2. Все имеющиеся вершины остаются не выделенными.

Шаг 3. Первой вершине присваивается статус текущей.

Шаг 4. Всем невыделенным вершинам вычисляется вес по формуле: вес невыделенной вершины есть минимальное число из старого веса данной вершины, суммы веса текущей вершины и веса ребра, соединяющего текущую вершину с невыделенной.

Шаг 5. После вычисления веса всех невыделенных вершин среди них ищется та, которая имеет минимальный вес. Она становится текущей и выделяется. Если же такая вершина не обнаружена, то строящийся маршрут не существует. Следовательно, выход.

Шаг 6. Путь считается найденным, если текущая вершина оказывается конечной. Вес данного пути равен весу конечной вершины.

Шаг 7. Переход на шаг 4 [2].

Для того, чтобы использовать описанный выше алгоритм для поиска кратчайшего пути (маршрута) были реализованы специальные классы и методы, которые проходят по всем уровням данного графа. Данные методы проходят по всем узлам верхнего уровня и ищут все подграфы, после чего поднимают все найденные подграфы на самый верхний уровень. На рис. 2 продемонстрировано конечный результат выполнения алгоритма.



Рис. 2. Результат выполнения алгоритма поиска кратчайшего пути (маршрута)

Заключение

В ходе выполнения данной работы было реализовано десктопное приложения, которое позволяет полностью создавать многоуровневые масштабируемые (3D) графы для электронных карт.

Реализованное приложение позволяет находить кратчайший путь между двумя узлами, расположенными на разных уровнях графа.

Данное приложение можно применять в современных геоинформационных системах таких как доставка продуктов, навигационные системы, предназначенные для авто- и авиатранспорта, в военной промышленности для навигации ракет и тому подобное.

Список литературы

1. Майо Дж. Самоучитель Microsoft Visual Studio 2010 / Дж. Майо. – СПб. : БХВ-Петербург, 2011. – 464 с.
2. Дасгупта С. Алгоритмы / С. Дасгупта, Х. Пападимитриу, У. Вазиран; пер. с англ. под ред. А. Шеня. – М. : МЦНМО, 2014. – 320 с.

РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ЭКСПЕРТНОЙ СИСТЕМЫ НА ОСНОВЕ МЕХАНИЗМА НЕЧЕТКОГО ВЫВОДА

С. В. Черепанова
Магистрант
Н. А. Тюкачев
Доцент

Введение

С развитием информационной деятельности человека и быстрым ростом объемов информации становится необходимым создание и разработка информационных технологий и эффективных интеллектуальных продуктов, способных справиться с обработкой больших объемов информации.

Экспертные системы обеспечивают поддержку принятия решений в ситуациях, для которых алгоритм заранее не известен и формируется по исходным данным в виде цепочки рассуждений (правил) из базы знаний.

В настоящее время в интеллектуальных информационных системах широко применяются нестрогие рассуждения, принятие решений в условиях неопределенности, неполной и нечеткой информации и осуществляются попытки моделирования человеческих приемов мышления на основе здравого смысла [1].

1. Точное земледелие

«Точное земледелие» — это система управления продуктивностью посевов, основанная на использовании комплекса спутниковых и компьютерных технологий. Вместо того, чтобы пахать, сеять, вносить удобрения «на глаз», как это делалось на протяжении всей предыдущей истории сельского хозяйства, сегодня фермеры могут точно рассчитать количество семян, удобрений и других ресурсов для каждого участка поля с точностью до метра.

Суть точного земледелия в том, что обработка полей производится в зависимости от реальных потребностей, выращиваемых в данном месте культур. Эти потребности определяются с помощью современных информационных технологий. При этом средства обработки дифференцируются в пределах различных участков поля, давая максимальный эффект при минимальном ущербе окружающей среде и снижении общего расхода применяемых веществ. Наиболее важным вопросом, решенным в последнее время в европейских странах, было нахождение оптимального уровня использования удобрений и химикатов в растениеводстве, а также определение доз их внесения, исключающих негативное воздействие на почву, растения и окружающую среду.

Накопление статистики обработки (куда и сколько внесли каждого вещества) и получаемых результатов (урожайность) позволяет применять различные виды анализа с тем, чтобы в дальнейшем корректировать применяемые дозы для получения максимума отдачи на каждую единицу вкладываемых денег в обработку.

Использование аппарата нечёткой логики при разработке базы знаний и механизмов вывода экспертной системы позволяет формализовать некоторые процедуры, например, оценку и выбор посева наиболее объективного сорта культуры на базе фрагментарной, ненадёжной и возможно неточной информации или принятие обоснованного решения по нужному количеству требуемых удобрений.

Нечёткие экспертные системы позволяют не только учитывать неопределенность, но и моделировать рассуждения, что с большим трудом поддается реализации в системах на классической логике. Таким образом, основной целью использования нечёткой логики является создание аппарата, способного моделировать человеческие рассуждения и объяснять приемы принятия решений.

Накопление статистики обработки (куда и сколько внесли каждого вещества) и получаемых результатов (урожайность) позволяет применять различные виды анализа с тем, чтобы в дальнейшем корректировать применяемые дозы для получения максимума отдачи на каждую единицу вкладываемых денег в обработку.

Если резюмировать достоинства данной системы, то получается следующее:

1. Оптимизация (минимизация) затрат сырья – семян, удобрений, воды и т. д.
2. Повышение урожайности используемых полей.
3. Улучшение качества получаемой продукции.
4. Повышение качественных характеристик используемой земли.
5. Снижение негативного влияния на окружающую среду.

Однако на пути внедрения данных технологий стоит несколько препятствий, такие как дороговизна, техническая сложность, отсутствие практического опыта.

Для минимизации и сокращения недостатков точечного земледелия предлагается разработка такого продукта, который будет основываться на многолетнем опыте агрономов.

Нет нужды вкладывать огромные средства в дорогостоящую инновационную технику, оцифровку километров полей сельскохозяйственных угодий и обучению агрономов сложной и интуитивно абсолютно им непонятной техники. Все что требуется для современного и качественного ведения точечного земледелия – это уже имеющийся многолетний опыт ведения сельского хозяйства агрономами из всех областей Российской Федерации.

Основная задача – это структурировать данные от агронома в интеллектуальную экспертную систему и применить методы нечеткой логики для полноценного формирования системы правил выводов [2].

2. Системы нечеткого вывода Мамдани-Заде

Элементы теории нечетких множеств, правила импликации и нечетких рассуждений образуют систему нечеткого вывода. В ней можно выделить множество нечетких правил, базу данных, содержащую описания функций принадлежности, а также механизм вывода и агрегирования. В технической реализации для обеспечения взаимодействия множеств входных и выходных значений вводится нечеткая система с так называемыми фаззификатором (преобразователем множества входных данных в нечеткое множество) на входе и дефаззификатором (преобразователем нечетких множеств в конкретное значение выходной переменной) на выходе. Структура такой системы представлена на рис. 1 [3].

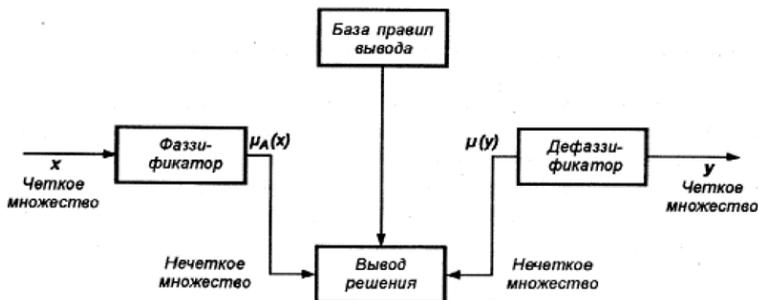


Рис. 1. Структура нечеткой системы с фаззификатором и дефаззификатором

Фаззификатор преобразует N -мерный входной вектор $x = [x_1, x_2, \dots, x_N]$ в нечеткое множество A , характеризуемое функцией принадлежности $\mu_A(x)$ с четкими переменными.

Общая форма гауссовской функции для переменной x с центром c и вариацией σ для множества F имеет вид

$$\mu_A(x) = \exp \left[- \left(\frac{x - c}{\sigma} \right)^2 \right]. \quad (1)$$

На рис. 2 представлена форма типовых гауссовских функций в случае фиксированного положения центра нечеткого множества c при различных значениях параметра σ .

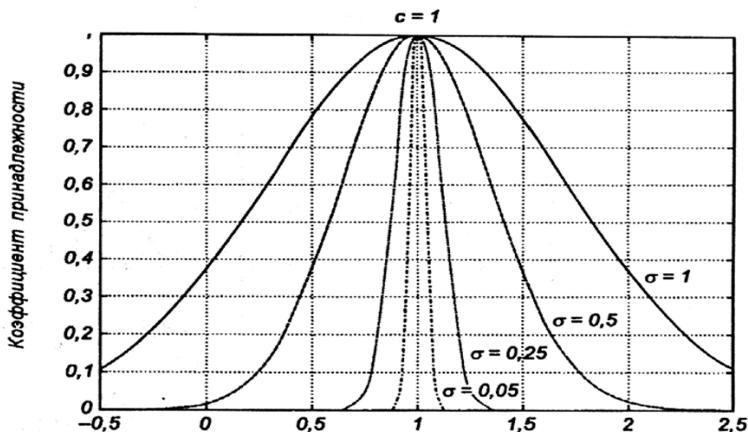


Рис. 2. Влияние значения параметра σ на форму гауссовской функции

Симметричная треугольная функция принадлежности, которую можно записать в виде

$$\mu_A(x) = 1 - \frac{|x - c|}{d}, x \in [c - d, c + d]. \quad (2)$$

$$\mu_A(x) = 0, x \in [c - d, c + d]. \quad (3)$$

Интерпретация центральной точки c и ширины d для треугольной функции представлена на рис. 3.

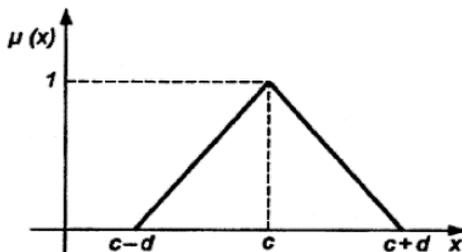


Рис. 3. Треугольная форма функции принадлежности

Обобщением треугольной функции является трапециевидальная функция принадлежности, форма и обозначения которой показаны на рис. 4.

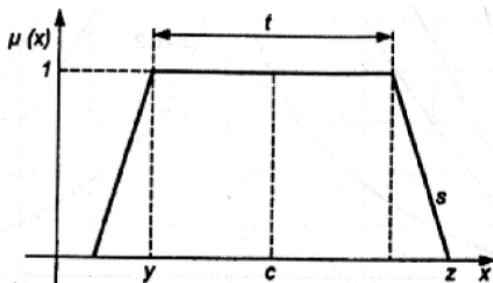


Рис. 4. Трапециевидальная форма функции принадлежности

Если определить $y = c - t/2 - 1/s$, $z = c + t/2 + 1/s$, где s обозначает угол наклона, то трапециевидальная функция описывается зависимостью. На рис. 5 представлен способ агрегирования двух правил нечеткого вывода при существовании двух значений переменных x_1 и x_2 .

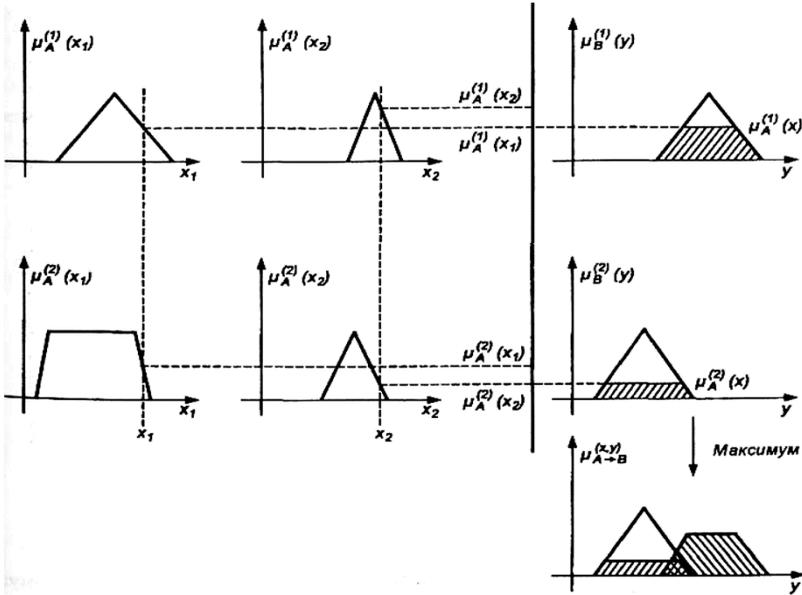


Рис. 5. Иллюстрация примера системы вывода Мамдани-Заде

$$\mu_A(x) = 0, \quad x > z \text{ or } x < y, \quad (4)$$

$$\mu_A(x) = 0, \quad c - \frac{t}{2} \leq x \leq c + \frac{t}{2}, \quad (5)$$

$$\mu_A(x) = s(z - x), \quad c + \frac{t}{2} \leq x \leq z, \quad (6)$$

$$\mu_A(x) = s(z - y), \quad y \leq x \leq c - \frac{t}{2} \quad (7)$$

Логическое произведение (оператор Min) используется как для агрегирования нечетких правил относительно конкретных переменных x_i ($i = 1, 2$), образующих вектор x , так и на уровне импликации $A \rightarrow B$ для одиночных правил вывода. Агрегирование импликаций, касающихся правил 1 и 2, проводится с использованием логической суммы (оператор Max). В правой нижней части рисунка представлен нечеткий результат в виде функции принадлежности переменной y . Получение четкого значения y , соответствующего также четким значениям входных переменных x_1 и x_2 , требовало бы в этом случае применения процедуры дефаззификации [4].

3. Объектно-ориентированный подход к реализации системы нечеткой логики

Базовым классом для типов фаззификации будет абстрактный класс Variant (листинг).

Листинг

Базовый класс

```
abstract class Variant {
    public ShortString name;
    public MyPoint[] points;
    public TypeVariant type;
    public int id;
    public int idVariable;
    public abstract void BegPoint(float x1, float x2);
    public abstract void MidlPoint(int m, int n, float x1,
float x2);
    public abstract void EndPoint(int n, float x1, float x2);
    public abstract float SetFunc(float x0); }
```

Он обладает общими свойствами, такими как: name класса ShortString (хранит в себе лингвистическую переменную), массивом точек points класса MyPoint[], типом type класса TypeVariant (тип фаззификации: треугольник, трапеция, гаусс), идентификатором id, idVariable и абстрактными методами, предназначенные для определения координат:

- неподвижных точек (BegPoint(), EndPoint());
- подвижных точек (MidlPoint());
- центральных точек, определяемых с помощью функции принадлежности (SetFunc());

Общая диаграмма абстрактного класса представлена на рис. 6, реализация осуществлена на языке C# с помощью Visual Studio 2015.

Заключение

В данной статье представлено краткое описание инновационной технологии точного земледелия и подробное описание процесса нечеткого вывода. Показан один из вариантов создания программного кода при помощи принципов объектно-ориентированного программирования. В итоге данного эксперимента получилась экспертная система на базе нечеткой логики. Применение решения использования многолетнего опыта агрономов для создания экспертной системы помогает решить большинство недостатков технологии точного земледелия, что является несомненным преимуществом.

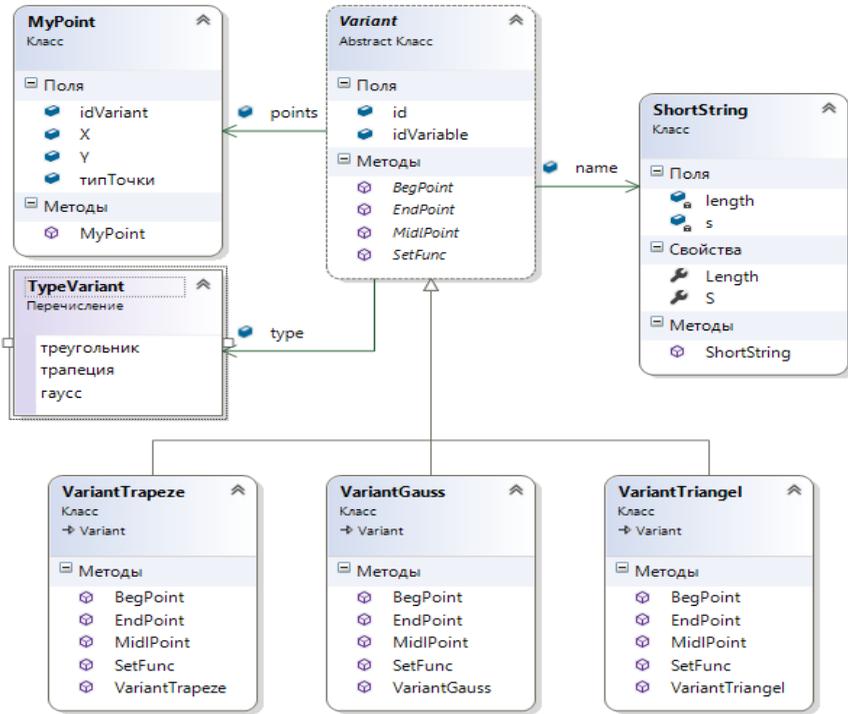


Рис. 6. Диаграмма абстрактного класса Variant

Список литературы

1. Джексон П. Введение в экспертные системы / П. Джексон. – Вильямс, 2001. – 397 с.
2. Сельхоз портал. Точное земледелие: принцип работы и перспективы [Электронный ресурс] / Новицкий И. – Режим доступа : <https://xn--80ajgpcpbhks4a4g.xn--p1ai/articles/tochnoe-zemledelie>
3. Джордж Ф. Люгер Искусственный интеллект. Стратегии и методы решения сложных проблем / Люгер Ф. Джордж. – 4-е изд. – М. : Издательский дом «Вильямс», 2003. – 864 с.
4. Спицын В. Г. Разработка экспертных систем на основе нечетких правил вывода / В. Г. Спицын. – Томск : Изд-во ТПУ, 2011. – 33 с.

НАТУРНОЕ МОДЕЛИРОВАНИЕ ЭЛЕМЕНТОВ БЕСПРОВОДНОЙ ОДНОРАНГОВОЙ ЯЧЕЙСТОЙ СЕНСОРНОЙ СЕТИ НА БАЗЕ ARDUINO

Д. Е. Шевцов

Магистрант

А. В. Стромов

Ассистент

Введение

В настоящее время распределенные системы все шире входят в нашу жизнь. Одним из направлений развития современных распределенных систем являются беспроводные сенсорные сети.

Беспроводная сенсорная сеть – это распределённая, самоорганизующаяся сеть множества датчиков (сенсоров) и, в некоторых случаях, исполнительных устройств, объединенных между собой посредством радиоканала. Каждый узел сети является ретранслятором, благодаря этому область покрытия подобной сети может составлять от нескольких метров до нескольких километров даже при небольшом радиусе действия отдельных узлов сети.

БСС применимы во многих областях; таких как мониторинг окружающей среды, здравоохранение, безопасность, сельское хозяйство и многие другие.

Узел беспроводной сенсорной сети состоит из вычислительного модуля, датчиков (сенсоров), модуля беспроводной связи и модуля электропитания [1].

1. Моделирование беспроводных сенсорных сетей

Одна из основных сложностей при моделировании БСС заключается в том, что топология сети, как правило, неизвестна заранее. Узлы сети могут располагаться произвольно, а расстояние между ними неизвестно заранее. К тому же, расстояние между различными узлами сети может значительно отличаться. Осложняется ситуация так же разнообразием сфер применения БСС. Создать универсальную сеть невозможно, поскольку для различных вариантов использования необходимы различные параметры сети.

Проектирование беспроводной сенсорной сети трудоёмкий процесс, требующий хорошего знания предметной области и большого количества времени. Чтобы упростить этот процесс, разработчики стараются использовать универсальные протоколы работы сети, успешно функционирующие при любой конфигурации сети. Однако разработка

таких протоколов требует проверки их характеристик, поэтому большое значение уделяется моделированию работы протоколов беспроводных ячеистых сетей. Моделирование БСС можно разделить на аналитическое, имитационное и натурное. Аналитическое моделирование основано на использовании методов математической физики и теории вероятностей, даёт обобщённую информацию и применимо только к некоторым протоколам на начальном этапе разработки. Имитационное моделирование выполняется в специализированном программном обеспечении на компьютере. Оно позволяет исследовать сети с большим количеством узлов и в различных конфигурациях, однако данный подход ограничен точностью и адекватностью используемых моделей. Как правило, тщательно моделируется только один уровень семиуровневой модели сети, а остальные имитируются упрощённо. Поэтому необходимо натурное моделирование, проводимое на сети, состоящей из реальных узлов. Это позволяет выполнить комплексную проверку всех используемых протоколов, программного обеспечения и, полностью или частично, аппаратного обеспечения. В качестве узлов могут использоваться как узлы, разработанные для данного проекта, так и узлы, предназначенные для моделирования. Примером такого узла является Mica node [2]. Проблема использования готовых решений заключается в их узкой направленности, то есть приспособленности для решения некоторой конкретной проблемы. Для исследования свойств сети наиболее подходящим решением является создание узла, параметры которого могут быть легко изменены, а работа с ним не потребует больших временных затрат. Поэтому актуальным является самостоятельная сборка подобного узла и исследование его параметров. Тем более что в последние годы рынок электронных компонент типа «сделай сам» развивается очень активно, а успешное внедрение в электронные компоненты принципов программно-определяемых радиосредств (SDR) обеспечивает простую настройку параметров приёмопередатчика. Актуальна также задача оценки характеристик такого узла.

2. Аппаратное обеспечение узла сети

В настоящее время широкую популярность имеет программно-аппаратная платформа Arduino. Основная часть плат Arduino построена на базе процессоров Atmel AVR, мощности которых будет вполне достаточно для выполнения поставленных целей. Выбор данной платформы обусловлен удобством работы с ней. Платы Arduino имеют встроенный загрузчик, что позволяет без проблем программировать контроллер, а так же выполнены в удобном форм-факторе, который даёт возможность собрать узел, который можно легко переконфигурировать или модернизировать. Для создаваемой модели были выбраны

микроконтроллеры Arduino Nano на базе процессора ATmega328, который имеет тактовую частоту 16 МГц и 32 Кб флеш-памяти для загрузки программ. Кроме того, он имеет 2 Кб ОЗУ и 1 Кб энергонезависимой памяти.

В качестве приёмопередатчика был выбран радиомодуль на базе микросхемы SI4432, который позволяет изменять частоту и мощность передатчика. Все настройки осуществляются программно. Так же имеется возможность изменить способ модуляции сигнала. Доступно три вида модуляции: FSK, GFSK и OOK. Работа возможна на частотах от 240 до 960 МГц, а максимальная мощность передатчика составляет +20 dBm

Общий вид собранного узла представлен на рис. 1.



Рис. 1. Общий вид узла сети

Собранный узел содержит следующие компоненты:

- микроконтроллер,
- радиомодуль,
- два индикаторных светодиода для отслеживания состояния узла,
- преобразователь логических уровней, который необходим для конвертирования напряжений 5В и 3.3В для взаимодействия микроконтроллера и приёмопередатчика,
- кнопку для управления режимами передачи,
- интерфейс для подключения внешнего дисплея,
- датчик температуры и влажности воздуха,
- гнездо для подключения питания.

К узлу могут быть подключены любые датчики из чрезвычайно широкой номенклатуры, предлагаемой для Arduino. Помимо этого микроконтроллер может работать с интерфейсами SPI и I2C, что позволяет подключать различные радиомодули и другие устройства.

3. Изучение свойств полученной сети

Изучение свойств созданной сети было начато с определения максимального радиуса действия сети на различных частотах. Для оценки качества передачи был выбран параметр RSSI, который показывает уровень входящего сигнала. Устойчивая передача с минимальным количеством потерь наблюдается при значениях RSSI не ниже -90 dBm. Это значение и было выбрано как пороговая величина, при которой сеть работает в нормальном режиме. Во всех опытах использовалась частотная модуляция. Результаты измерений изображены на рис. 2. Не трудно заметить, что дальность передачи сигнала зависит от частоты.

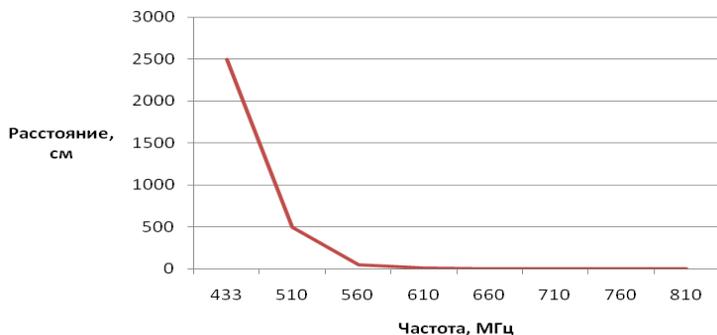


Рис. 2. Зависимость расстояния передачи от частоты

К сожалению, реальные условия, в которых эксплуатируются БСС, зачастую далеки от идеальных. Одним из факторов, отрицательно влияющих на качество передачи данных между узлами, являются радиопомехи, то есть передача сигналов на той же, или близкой частоте. Это могут быть как какие-либо другие сети, так и передача данных между узлами той же сети. Следующим экспериментом было определение влияния помехи на передачу данных. В качестве источника помех использовался ещё один узел сети, который постоянно передавал данные с очень близкими параметрами. Такие сообщения игнорировались основными узлами, но при этом занимали радиоканал. Передача осуществлялась на частоте 610 МГц, мощность передатчика 1 dBm. Было посчитано количество потерянных сообщений при различной мощности помехи. График изображён на рис. 3.

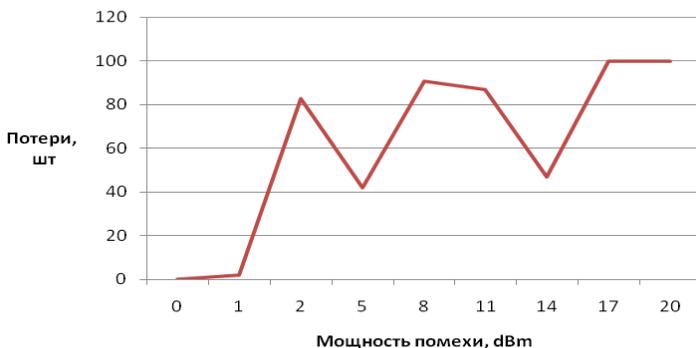


Рис. 3. Зависимость количество потерянных сообщений от мощности помехи

Получившийся график нельзя считать конечным - даже при одинаковых условиях влияние помех на передачу сообщений очень непостоянно. Например, при наблюдении за сетью в незначительно отличающихся условиях дало следующие результаты: при линейном увеличении мощности шума количество потерянных сообщений изменяется хаотично и заметно меняется от измерения к измерению. Пример такого поведения представлен на рис. 4.

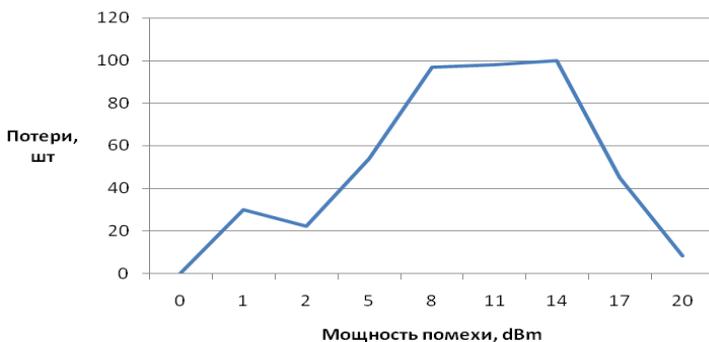


Рис. 4. Зависимость количество потерянных сообщений от мощности помехи

Заключение

Натурное моделирование беспроводных одноранговых сетей в настоящее время является актуальным и доступным направлением исследований, имеющим значительное теоретическое и практическое значение, и не требующим существенных затрат.

Использование готовых компонентов «экосистемы» Arduino позволяет создавать гибко конфигурируемые модели узлов беспроводных сетей.

В реальных условиях характеристики узлов меняются не в строгом соответствии с теоретическими предположениями, но со случайными отклонениями, которые могут быть вызваны помехами, условиями распространения сигнала и т. д. Впрочем, в реальных сетях будет наблюдаться то же самое, так что это не является недостатком, но скорее достоинством натурального моделирования, позволяющим оценить работу исследуемого протокола в действительно реальных условиях.

Список литературы

1. Dargie W. Fundamentals of wireless sensor networks: theory and practice / W. Dargie, C. Poellabauer. – New York : John Wiley and Sons, 2010. – 330 p.
2. Hill J.L. Mica: A Wireless Platform for Deeply Embedded Networks / J.L. Hill, D.E. Culler // Proceedings of the 35th Annual International Symposium on Microarchitecture. – 2002. – P. 12-24.

РАЗРАБОТКА ПЛАГИНОВ ДЛЯ РЕАЛИЗАЦИИ НОВЫХ ТИПОВ ТЕСТОВ В СИСТЕМЕ MOODLE

М. А. Шмелев

Студент

А. П. Толстобров

Доцент

Введение

В настоящее время в образовательном процессе все чаще используются виртуальные обучающие среды. Одной из самых известных обучающих платформ в наше время является электронная система Moodle. Она используется в университетах, колледжах, школах, училищах [1].

В настоящее время более ста тысяч учебных организаций по всему миру используют образовательную систему Moodle, чтобы предоставить обучающимся онлайн-курсы и дополнить традиционные аудиторные способы обучения. Важно выделить ключевые особенности системы Moodle [2]:

1. Бесплатная система с открытым исходным кодом;
2. Модульность;
3. Удобство обучения;
4. Всемирное сообщество пользователей Moodle.

1. Понятие игровых плагинов. Цели создания

Эффективным способом контроля знаний обучающихся в электронной среде является использование тестов. Стандартная реализация тестов в системе Moodle подразумевает использование в тесте фиксированного количества вопросов.

Игровой плагин – это особый вид теста, нацеленный на повышение интереса обучающихся к учебе. Особенности данного плагина (интерактивные элементы, порядок выдачи вопросов) должны способствовать формированию у обучающихся представления, что они проходят игру. В то же время с помощью игрового плагина проверяются знания испытуемого.

На основе данного определения можно выделить две основные цели создания плагинов:

- Повышение эффективности проверки знаний обучающихся;
- Улучшение интереса к учебе.

2. Игровой плагин «Кто хочет стать миллионером?»

Данный игровой плагин реализован на основе известной телевизионной версии игры «Кто хочет стать миллионером?».

Правила моделируемой игры состоят в следующем. Участник последовательно отвечает на 15 вопросов. При неправильном ответе игра прекращается. В процессе выполнения теста-игры участник может воспользоваться одной из 4 подсказок: «50-50», «Звонок другу», «Помощь зала» и «Двойной шанс».

В данном игровом плагине подсказка «50-50» означает, что из 4 вариантов ответа остается только два, один из которых правильный. Аналог подсказки «Звонок другу» реализован таким образом, что испытуемому будет подсказан как-бы правильный вариант ответа, но вероятность того, что он действительно является правильным, равна 80%.

Следующая подсказка «Помощь зала» воплощена в коде таким образом, что каждому предлагаемому ответу присваивается определенное численное значение (аналог числа голосов за данный ответ). При этом правильный ответ будет иметь большее численное значение по сравнению с другими неправильными ответами лишь с вероятностью 80%.

Реализация плагина приближена к современной телевизионной версии игры, в которой изменены суммы выигрыша за каждый отдельный

вопрос и добавлен новый вид подсказки «Двойной шанс», в которой участнику сообщается, что один из двух ответов правильный.

На рис. 1 представлен интерфейс реализованного плагина.

Иконки кнопок для выбора подсказок «50-50», «Звонок другу», «Помощь зала» и «Двойной шанс» располагаются слева направо в левом верхнем угле интерфейса. Правее подсказки «Двойной шанс» расположена кнопка для выхода из игры. Также представлены поля для текста вопроса и для четырех вариантов ответов на него. В правой части интерфейса обозначены суммы (очки), которые можно получить за правильные ответы. Поле в правом нижнем углу предназначено для отображения информации от подсказок.



Рис. 1. Первый вопрос в игровом плагине

На следующем рис. 2 показано, что при правильном ответе на 15 вопросов игра завершается. Испытуемый признается победителем и получает максимальное количество баллов при оценивании.

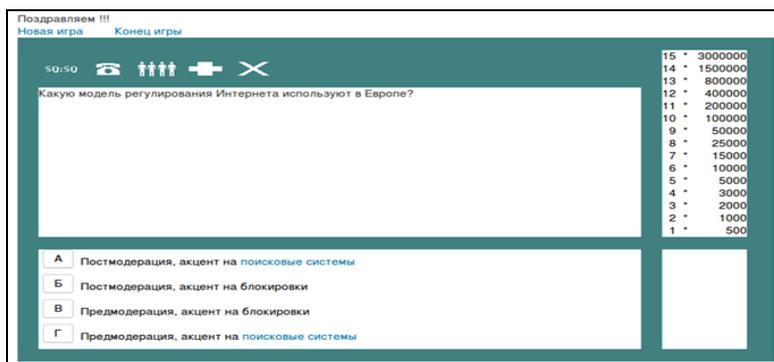


Рис. 2. Успешное прохождение игры

3. Игровой плагин «Вершина»

Данный игровой плагин не имеет воплощения в виде телевизионной версии. Его правила были предложены в рамках данной разработки. Они состоят в следующем.

При выполнении теста его участник должен последовательно ответить на 20 вопросов. В тесте-игре имеется 3 уровня сложности вопросов: легкий, средний и тяжелый. Присутствует возможность получения испытуемым в процессе выполнения теста нескольких подсказок. Переход от одного уровня сложности к другому происходит после ответа на определенное количество вопросов. Если обучающийся начинает отвечать на более сложные вопросы, это означает его переход на более высокий уровень. Если обучающийся начинает отвечать на более простые вопросы, то это означает, что он перешел на уровень ниже.

Важно отметить, что в игровом плагине «Вершина» проверяемый отвечает на конкретный вопрос только один раз и следующие вопросы подбираются для него в зависимости от результатов ответа на серию предыдущих вопросов. То есть явно выражена блочная концепция.

Данный игровой плагин обладает важным свойством адаптивности. Здесь испытуемый отвечает на конкретный вопрос только один раз, а состав и количество следующих вопросов подбираются в зависимости от результатов ответа на серию предыдущих вопросов.

Важным достоинством свойства адаптивности теста является сокращение времени тестирования, по сравнению с традиционными тестами с фиксированным количеством вопросов, без потери, при этом, качества тестирования, то есть точности оценки уровня подготовки испытуемых [3].

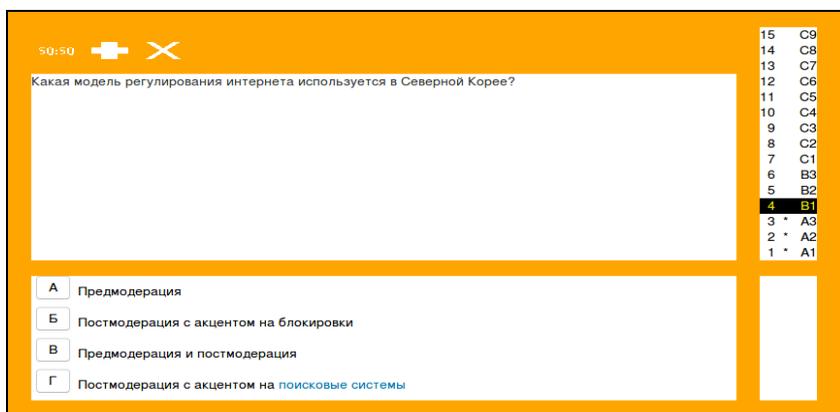


Рис. 3. Прохождение игрового плагина «Вершина»



Рис. 4. Иллюстрация использования подсказки

4. Возможные вариации создания игровых плагинов

С изменением нескольких параметров могут получаться другие игровые плагины. Перечислим данные параметры:

А) Число уровней вопросов;

В игровом плагине «Вершина», например, 3 уровня сложности. А в плагине «Кто хочет стать миллионером?» по сути один.

Б) Число правильных ответов для перехода на другой уровень;

В) Число неправильных ответов для перехода на другой уровень;

В игре Вершина после 3 правильных/неправильных ответов возможен переход на другой уровень. В Миллионере после 1 неправильного ответа игра заканчивается.

Г) Число вопросов в игре.

В игре Миллионер – 15 вопросов, а в игре Вершина – 20 вопросов.

Следовательно, возможно большое количество различных вариаций реализации игровых плагинов.

Заключение

Данная работа посвящена созданию игровых плагинов для реализации игровых типов тестов, расширяющих возможности системы электронного обучения Moodle. Использование второго плагина позволяет, по сравнению с обычным тестом, сократить время тестирования без потери его качества. Реализация данных плагинов позволила достичь поставленных целей:

- Повышение эффективности проверки знаний обучающихся;
- Улучшение интереса к учебе.

Список литературы

1. Официальный сайт Moodle [Электронный ресурс]. – Режим доступа : https://docs.moodle.org/archive/ru/История_развития
2. Пиший С.А. Система управления курсами Moodle [Электронный ресурс]. – Режим доступа : <http://www.scienceforum.ru/2016/1584/23359>
3. Голанова А. В. Адаптивное тестирование как одна из форм компьютерного тестирования [Электронный ресурс] / А. В. Голанова, Е. И. Голикова // Царскосельские чтения. – 2010. – v. XIV. – Режим доступа : <https://cyberleninka.ru/article/n/adaptivnoe-testirovanie-kak-odna-iz-form-kompyuternogo-testirovaniya>

АВТОМАТИЗИРОВАННАЯ СИСТЕМА ВЕДЕНИЯ ТЕМПОРАЛЬНОЙ БАЗЫ ДАННЫХ СИСТЕМЫ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ ПО ИДЕНТИФИКАЦИИ БЕСПИЛОТНЫХ ЛЕТАТЕЛЬНЫХ АППАРАТОВ

А. В. Юшин
Студент
В. Ю. Храмов
Профессор

Введение

В магистерской диссертации Копыловой Д.В. «Система поддержки принятия решений в интересах идентификации беспилотных летательных аппаратов» [1] обоснованы состав и структура системы поддержки принятия решений (СППР) в интересах идентификации беспилотных летательных аппаратов (БЛА), а также разработаны алгоритмы её функционирования и осуществлена программная реализация.

Однако, формирование входной нечеткой ситуации, характеризующей идентифицируемый БЛА, на сегодняшний день при работе с системой осуществляется оператором в ручном режиме. В реально функционирующей системе данное формирование должно осуществляться в автоматическом режиме по результатам поступления информации, идентифицирующей БЛА, от различных источников. Причем данная информация будет приходиться в различные моменты времени и характеризовать отдельные идентификационные признаки БЛА. После поступления данной информации СППР должна сформировать на их

основе единое описание входной нечеткой ситуации и уже по ней осуществить идентификацию БЛА.

Таким образом, в системе в явном виде прослеживается связь идентификационных признаков БЛА, поступающих от различных источников информации, с временными метками времени. Это в свою очередь требует разработки подходов к представлению временной информации и способов ее обработки в СППР.

В статье проведен анализ существующих подходов к представлению темпоральной (временной) модели данных, обоснован выбор темпоральной модели данных для её реализации в СППР с помощью реляционной системы управления базами данных (СУБД), разработан алгоритм функционирования и выполнена программная реализация автоматизированной системы на языке программирования C# с помощью СУБД PostgreSQL.

Состав и структура СППР для идентификации БЛА представлены на рис. 1.

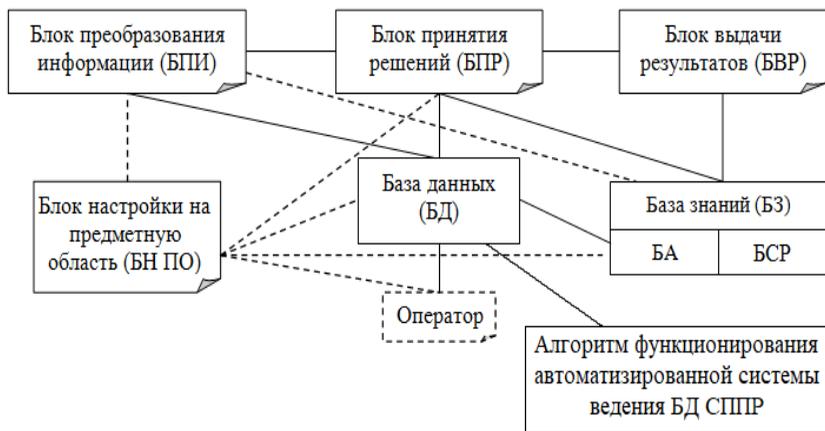


Рис. 1. Состав и структура СППР для идентификации БЛА

На рис. 1 пунктирными линиями отображены взаимосвязи блоков при настройке системы на предметную область, а сплошными – при ее функционировании [1].

Блок преобразования информации (БПИ) предназначен для преобразования типовых и входных ситуаций к нечеткому виду путем проецирования «четких» (количественных и качественных) значений описывающих их признаков на соответствующие им функции принадлежности, построенные в блоке настройки на предметную область

(БН ПО). Преобразованные таким образом признаки входной ситуации в блоке принятия решений (БПР) сравниваются с признаками типовых ситуаций, хранящимися в базе знаний СППР, и определяется типовая ситуация, наиболее "близкая" к нечеткой входной ситуации, которой ставится в соответствие определенное решение. Блок выдачи результатов (БВР) обеспечивает выдачу оператору информации, связанной с распознанной входной ситуацией и соответствующей ей типовой ситуации. База данных (БД) предназначена для хранения постоянных и переменных данных, используемых алгоритмами принятия решений, а также данных для формирования нечеткой входной ситуации. База знаний (БЗ) состоит из двух компонентов: БА – библиотеки алгоритмов принятия решений; БСР – библиотеки ситуаций и решений, в которой хранятся формализованные описания всех типовых ситуаций.

В отличие от традиционных моделей данных, обеспечивающих хранение лишь мгновенного снимка объектов предметной области, темпоральные модели данных позволяют хранить информацию об эволюции объектов: для любого объекта, который был создан в момент времени T_1 и закончил свое существование в момент времени T_2 , в БД будут сохранены все его состояния на временном интервале $[T_1, T_2]$.

В соответствии с устоявшимися понятиями темпоральная модель данных (ТМД) – это модель данных, ориентированная на хранение темпоральных данных, все аспекты которой также должны быть темпоральными. Традиционная модель данных – $M=(DS, OP, C)$ – состоит из трех компонент: структура данных DS , операции OP и ограничения целостности C , а темпоральная – $MT=(DST, OPT, CT)$ – должна поддерживать все понятия, входящие в каждый из трех компонент, с учетом изменений данных во времени. Структура данных должна быть адаптирована таким образом, чтобы имелась возможность хранить темпоральные данные. Алгебру и операции модификации следует переопределить, используя темпоральную семантику. Дополнительно для каждого ограничения целостности в нетемпоральной модели данных M темпоральная – MT – должна поддерживать темпоральный аналог нетемпорального ограничения. Семантика темпоральных ограничений целостности также должна быть переопределена.

Таким образом, разработка ТМД предполагает развитие следующих темпоральных составляющих: структура данных, ограничения целостности и ключи, запросы и модификации, алгебра [2].

Темпоральные данные могут быть связаны как с дискретным представлением времени – моментом, так и с интервальным. Преимущество модели, основанной на дискретном представлении, заключается в ее простоте с точки зрения поддержки реляционных СУБД.

1. Анализ существующих подходов к представлению темпоральной модели данных

Рассматривались пять подходов к представлению темпоральных данных [3].

1. Модель представления темпоральных данных, предложенная Р. Снодграсом. Пусть битемпоральное отношение R имеет набор атрибутов (A_1, \dots, A_n, T) , где T – битемпоральный атрибут, определенный на множестве битемпоральных элементов. Тогда R можно записать в следующем виде: $R=(A_1, \dots, A_n, Ts, Te, Vs, Ve)$.

Дополнительные атрибуты Ts, Te, Vs, Ve – это атомарные темпоральные атрибуты времени, содержащие дату начала и окончания транзакционного и модельного времени. Данное представление отношений является самым естественным и наиболее часто используемым способом представления битемпоральных отношений.

2. Модель представления темпоральных данных, предложенная К. Дженсеном. Особенность данного представления в том, что исторические кортежи никогда не обновляются, то есть доступны только для чтения. Таким образом, это представление данных хорошо подходит для основанного на архивах хранения битемпоральных отношений. Битемпоральное отношение R с набором атрибутов A_1, \dots, A_n может быть представлено в следующем виде: $R=(A_1, \dots, A_n, Vs, Ve, T, Op)$.

Как и в предыдущей схеме представления данных, атрибуты Vs и Ve хранят даты начала и окончания актуальности факта в моделируемой реальности соответственно. Атрибут T хранит информацию о времени внесения кортежа в журнал изменений. Запросы на создание и удаление кортежей обозначаются в атрибуте Op соответствующими символами – I (Insert) и D (Delete). Модификация данных представляет собой пару запросов – удаление и создание записи – с одинаковым атрибутом времени T .

3. Модель представления темпоральных данных, предложенная С. Гадией. Данный подход предполагает наличие битемпоральных меток у каждого из атрибутов кортежа, что обеспечивает возможность более гибкого моделирования реальности. Пусть битемпоральное отношение R имеет атрибуты (A_1, \dots, A_n, T) , где T – темпоральный атрибут, определенный на множестве битемпоральных элементов. Тогда битемпоральное отношение R может быть представлено в виде отношений, где каждый из атрибутов имеет свою темпоральную метку:

$$R=\{([Ts, Te] [Vs, Ve] A_1)\}, \dots, \{([Ts, Te] [Vs, Ve] A_n)\}.$$

Кортеж состоит из n элементов. Каждый элемент представляет собой тройку значений: транзакционное время $[Ts, Te]$, модельное время $[Vs, Ve]$ и значение атрибута A_i .

4. Модель представления темпоральных данных, предложенная Е. МакКензи. В данной модели битемпоральное отношение – это последовательность состояний в модельном времени, проиндексированная транзакционным временем. В кортежах с модельным временем атрибуты имеют свои темпоральные метки. Битемпоральное отношение R с набором атрибутов A_1, \dots, A_n может быть представлено в виде отношения, в котором каждый атрибут помечается временной меткой: $R=(VR, T)$, где VR – отношение в модельном времени; T – транзакционное время. Схема состояний отношения модельного времени имеет вид: $VR=(A_1V_1, \dots, A_nV_n)$, где A_1, \dots, A_n – набор атрибутов; V_i – атрибут модельного времени, связанный с каждым атрибутом A_i и обозначающий время актуальности значения атрибута A_i в моделируемой реальности.

5. Модель представления темпоральных данных, предложенная Дж. Бен-Зви. Пусть битемпоральное отношение R состоит из набора атрибутов (A_1, \dots, A_n, T) , где T – темпоральный атрибут, определенный на множестве битемпоральных элементов. Тогда R может быть представлено в модели Бен-Зви следующим образом: $R=(A_1, \dots, A_n, Tes, Trs, Tee, Tre, Td)$.

В кортеже значение атрибута Tes (effective start) – это время, когда значение атрибута кортежа становится актуальным. Атрибут Trs хранит информацию о том, когда Tes было сохранено в БД. Аналогично Tee хранит информацию о том, когда факт перестает быть актуальным в моделируемой реальности, а Tre – когда Tee было зафиксировано в БД. Атрибут Td указывает на время, когда запись была логически удалена из БД.

В ходе реализации системы ведения БД, опираясь на требования к СППР, из рассматриваемых подходов была выбрана модель Бен-Зви. Данная модель рассматривает дискретные значения времени и обеспечивает темпоральность на уровне кортежей.

Признаки, используемые для представления типовых ситуаций, хранятся базе данных системы. При этом в нетемпоральной таблице хранится вся текущая информация по признакам идентифицируемого БЛА (рис. 2).

Однако, с учетом ограничения по дополнительной обработке, темпоральную модель следует модифицировать в части использования только транзакционного времени, так как действительное время в большинстве случаев не будет известно. Кроме того, для дальнейшей обработки при функционировании СППР оно не требуется.

Поэтому темпоральную модель данных можно представить в виде

$$R=(A_1, \dots, A_n, Trs, Tre, Td).$$

№ п/п	Наименование атрибутов
1	Диапазоны частот, МГц
1.1	Линия «вверх»
1.2	Линия «вниз»
2	Род работы каналов
2.1	Линия «вверх»
2.2	Линия «вниз»
3	Вид модуляции
3.1	Линия «вверх»
3.2	Линия «вниз»
4	Разделение каналов передачи и приема
5	Ширина полосы сигнала, МГц
6	Скорость передачи данных, кбит/с
6.1	Команды, телеметрия
6.2	Видео
7	Формат сжатия видео
8	Канальное кодирование
9	Расстояние НСУ-БЛА, км
10	Скорость полета, км/ч
11	Высота полета, м

Рис. 2. Нетемпоральная таблица

Атрибут Trs хранит информацию о том, когда значение было сохранено в БД, а Tre – когда информация о том, что факт перестает быть актуальным в моделируемой реальности, была зафиксирована в БД. Атрибут Td указывает на время, когда запись была логически удалена из БД.

2. Алгоритм функционирования автоматизированной системы ведения БД СППР

Алгоритм функционирования автоматизированной системы ведения БД СППР представлен на рис. 3.

Алгоритм включает следующие шаги.

1. Оператор вводит в специальную форму параметр Δ .
2. Система случайным образом выбирает одну из темпоральных таблиц для записи данных.
3. Производится расчет параметра $Tre = Trs + \Delta$.
4. Параметр Tre записывается в другие таблицы.
5. Происходит генерация значений в других темпоральных таблицах на интервале Δ .
6. Полученная информация записывается в нетемпоральную таблицу.
7. Происходит вывод информации об идентифицируемом БЛА из нетемпоральной таблицы оператору.

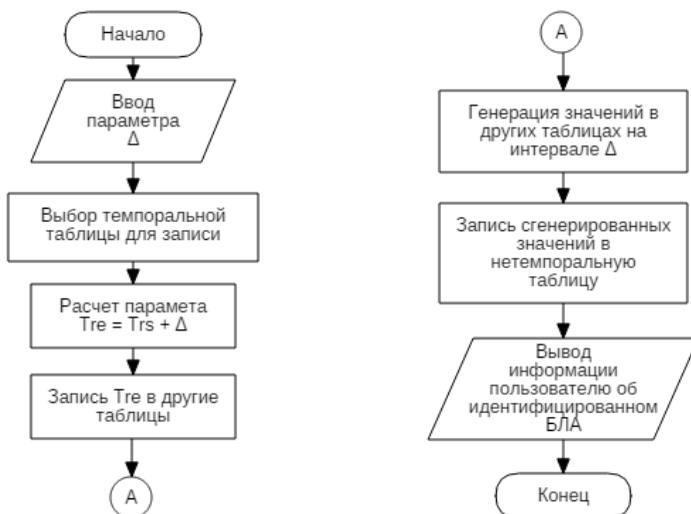


Рис. 3. Алгоритм функционирования автоматизированной системы ведения БД СППР

3. Программная реализация

Автоматизированная система ведения БД СППР была реализована на языке программирования C# с использованием СУБД PostgreSQL [4].

На рис. 4 представлен результат формирования нетемпоральной таблицы, содержащей значения признаков идентифицируемого БЛА, из трех темпоральных таблиц на интервале $\Delta = 5$ секунд.

Заключение

В статье проведен анализ существующих подходов к представлению темпоральной модели данных, включая 5 моделей, обоснован выбор модели данных для её реализации в СППР. В качестве темпоральной модели данных предложено использовать модель Дж. Бен-Зви. Разработан алгоритм функционирования автоматизированной системы ведения базы данных системы поддержки принятия решений и осуществлена его программная реализация на языке программирования C# с использованием СУБД PostgreSQL.

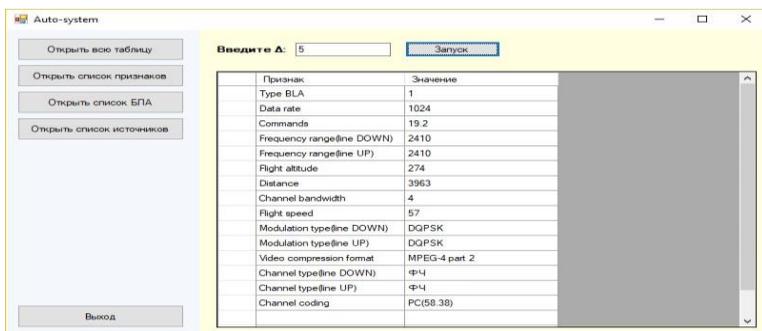


Рис. 4. Формирование нетемпоральной таблицы

Список литературы

1. Копылова Д.В. Система поддержки принятия решений в интересах идентификации беспилотных летательных аппаратов / Магистерская диссертация. – Воронеж : ВГУ, 2016. – 81 с.
2. Саймон А.Р. Стратегические технологии баз данных / А.Р. Саймон. – М. : Финансы и статистика, 1999. – 479 с.
3. Snodgrass R. Developing Time-Oriented Database Applications in SQL / R. Snodgrass. – Berlin : Morgan Kaufmann Publishers, 2000. – 544 p.
4. Храмов В.Ю. Методы и средства проектирования баз данных. Монография / В.Ю. Храмов, Э.Б. Ханов, А.И. Кустов. – Воронеж : Воронежский ЦНТИ-филиал ФГБУ «РЭА» Минэнерго России, 2015. – 188 с.

ПЛАГИН ДЛЯ НАГЛЯДНОЙ ВИЗУАЛИЗАЦИИ ИЗМЕНЕНИЯ ПЕРЕМЕННЫХ И СТРУКТУР ДАННЫХ ВО ВРЕМЯ ОТЛАДКИ

Ю. А. Яковлева

Студент

Д. И. Соломатин

Старший преподаватель

Введение

Обучение программированию включает в себя изучение различных алгоритмов и структур данных, при этом возникает потребность в средствах их визуализации в связи с тем, что для многих учащихся

изучение алгоритмов без такой визуализации является крайне затруднительным [1]. Даже при выполнении средством визуализации своей задачи на высоком уровне без значительных усилий со стороны преподавателя и хорошей организации преподаваемого материала добиться значительных улучшений результатов учащихся представляет достаточно большую сложность [2].

Стоит отметить, при изучении основ программирования, как правило, какой-то индивидуальной визуализации не требуется, можно обойтись универсальным подходом в представлении переменных и структур данным, главное, чтобы такое представление было применимо для большинства программ, которые реализуют учащиеся, без каких-либо дополнительных усилий с их стороны.

В данной работе предпринята попытка разработки такого инструмента для наглядной визуализации изменений переменных и структур данных во время отладки.

1. Постановка задачи

Учитывая, что в настоящий момент изучение программирования на первом курсе факультета компьютерных наук ВГУ для большинства студентов начинается с языка C#, было принято решение о разработке такого инструмента в виде плагина для IDE Visual Studio.

К разрабатываемому средству выдвинуты следующие требования:

- отображение состояния выполняемой программы в отдельном окне IDE Visual Studio во время отладки;
- отображение переменных текущего стекового кадра;
- наглядное отображение структур данных (например, массив, список, множество);
- возможность перемещения элементов визуализации пользователем.

2. Интеграция с IDE Visual Studio

Учитывая, что необходимо разработать плагин для IDE Visual Studio, необходимо провести анализ возможных способов реализации расширения для этой среды разработки. Расширение функциональности IDE Visual Studio возможно следующими способами:

- создание макроса – применяется при необходимости автоматизировать небольшие повторяющиеся задачи;
- создание Visual Studio Package – используется при необходимости интегрирования больших компонентов, например, нового языка программирования;
- создание проекта VSIX – применяется при необходимости создать дополнительные пункты меню команд Visual Studio, окна панелей инструментов и прочего, а также для обертки нескольких расширений в

одно; VSIX-файл представляет собой устанавливаемое расширение для Visual Studio [3].

Принимая во внимание выдвигаемые к проекту требования, наиболее подходящим способом для расширения среды разработки является VSIX-проект.

Для разработки расширения для IDE Visual Studio был использован набор средств Visual Studio Software Development Kit, разработка велась на языке C#.

Для отрисовки локальных переменных и структур данных была использована библиотека с открытым исходным кодом Microsoft Automatic Graph Layout, предоставляющая большое количество способов настройки представления отображаемых объектов и возможность их перемещения.

3. Обзор существующих решений для визуализации

В настоящее время существует множество средств визуализации, из которых были рассмотрены следующие:

- стандартные окна визуализации IDE;
- Pythontutor [4];
- Data Display Debugger (DDD) [5].

В большинстве современных средствах разработки присутствуют окна, отображающие состояния локальных переменных во время отладки. Но, не смотря на то, что в таких окнах представлена максимально полная информация о переменных, порой данное отображение не является наглядным, т.к. перегружено информацией, среди которой быстрый поиск необходимой информации затруднен, особенно для только начинающих изучать программирование.

Интересным проектом является Pythontutor – это веб-сайт, специально разработанный для визуализации структур данных и демонстрации пошагового выполнения программ. Несмотря на название, в настоящий момент проект поддерживает несколько языков программирования, но, к сожалению, не C#. Также у данного инструмента можно отметить следующий значительный недостаток – он позволяет работать только с очень маленькими демонстрационными проектами.

Data Display Debugger представляет собой графический визуализатор для отладчиков с интерфейсом командной строки. Даже учитывая то, что в большинстве случаев DDD представляет переменные достаточно наглядно, при наличии большого количества связей между отображаемыми объектами наглядность визуализации сильно падает.

4. Отображение стандартных структур данных

Ниже описаны и продемонстрированы результаты отображения разработанным плагином различных структур данных и локальных переменных во время отладки.

Отображение массива, создание которого описано в листинге 1, представлено на рис. 1.

Листинг 1

Создание и заполнение массива

```
string[] arr = new string[3];  
arr[0] = "cat";  
arr[1] = "waffles";  
arr[2] = "toad";
```

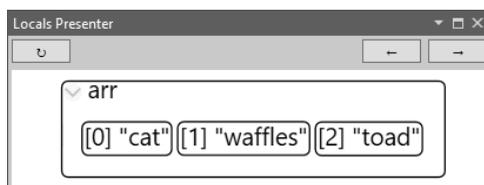


Рис. 1. Отображение массива

В окне отображения переменных можно наблюдать графическое представление массива `arr` в виде прямоугольника, внутри которого содержатся другие прямоугольники – элементы данного массива. Все три строки уже были добавлены в массив, поэтому мы видим полностью заполненный массив. Внутри каждого прямоугольника в квадратных скобках также присутствует число, обозначающее индекс элемента в массиве.

Как и во многих языках программирования, в стандартной библиотеке для языка C# есть несколько реализаций интерфейса списка: на основе массива (`List<T>`) и в виде связанного списка (`LinkedList<T>`). Именно этим различием и обусловлена разница в представлении этих двух типов списков, код создания которых представлен в листингах 2 и 3, а отображение можно видеть на рис. 2 и 3.

Листинг 2

Создание и заполнение списка

```
List<int> list = new List<int>();  
list.Add(1);  
list.Add(2);  
list.Add(3);
```

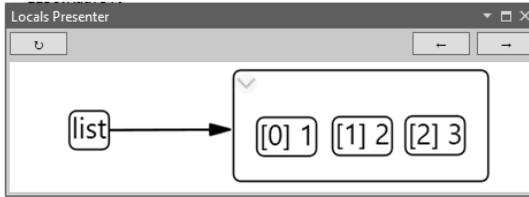


Рис. 2. Отображение списка

Листинг 3

Создание и заполнение связанного списка

```
LinkedList<int> llist = new LinkedList<int>();
llist.AddFirst(1);
llist.AddFirst(2);
llist.AddFirst(3);
llist.AddFirst(4);
llist.AddFirst(5);
```

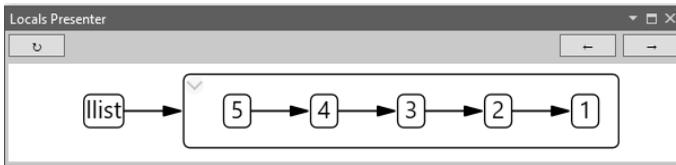


Рис. 3. Отображение связанного списка

Как можно видеть, в последнем случае связи между элементами с помощью ссылок представлены для наглядности стрелками.

Отображение множества, код создания которого приведен в листинге 4, показано на рис. 4.

Листинг 4

Создание и заполнение множества

```
HashSet<int> set = new HashSet<int>();
set.Add(3);
set.Add(12);
set.Add(6);
set.Add(45);
set.Add(8);
```

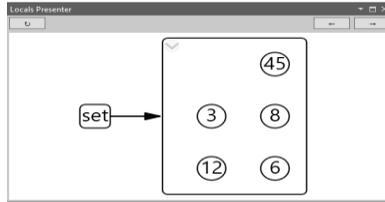


Рис. 4. Отображение множества

Несмотря на то, что внутри множества `HashSet<T>` элементы хранятся в конечном итоге в виде массива, намного нагляднее изобразить элементы неупорядоченными, без индексов.

5. Отображение пользовательских структур данных

Разработанный плагин также поддерживает отображение пользовательских классов. Ниже показана работа с классами, описание которых представлено в листинге 5, а отображение во время отладки – на рис. 5.

Листинг 5

Пример пользовательского класса

```
public class MyLList<T>{
    private MyLListNode<T> First { get; set; }
    public void Add(T data){
        //добавление нового элемента в начало списка
        ...
    }
    private class MyLListNode<T>{
        public T Data { get; set; }
        public MyLListNode<T> Next { get; set; }
    }
}
```

Рис. 5. Отображение пользовательского класса

В момент времени, представленный на рисунке, в пользовательский список уже были добавлены три элемента. Инструкция на строке 103 еще не выполнена. После её выполнения и добавления нового элемента в список, можно наблюдать состояние, представленное на рис. 6.

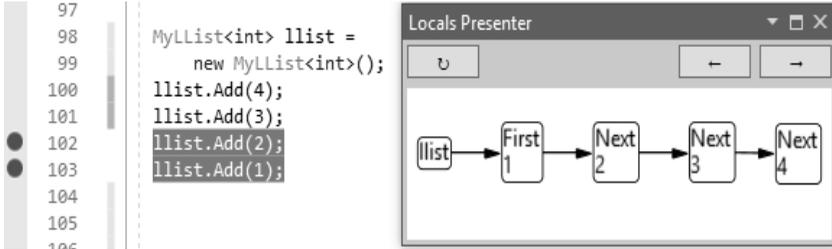


Рис. 6. Добавление нового элемента

Сравним теперь данное отображение со стандартным отображением локальных переменных, предлагаемым IDE Visual Studio. Оно представлено на рис. 7.

Локальные		
Имя	Значение	Тип
llist	{ConsoleApp2.MyLinkedList<int>}	ConsoleApp2.MyLinkedList<int>
First	{ConsoleApp2.MyLinkedList<int>.MyLinkedListNode}	ConsoleApp2.MyLinkedList<int>.MyLinkedListNode<int>
Data	1	int
Next	{ConsoleApp2.MyLinkedList<int>.MyLinkedListNode}	ConsoleApp2.MyLinkedList<int>.MyLinkedListNode<int>
Data	2	int
Next	{ConsoleApp2.MyLinkedList<int>.MyLinkedListNode}	ConsoleApp2.MyLinkedList<int>.MyLinkedListNode<int>
Data	3	int
Next	{ConsoleApp2.MyLinkedList<int>.MyLinkedListNode}	ConsoleApp2.MyLinkedList<int>.MyLinkedListNode<int>

Рис. 7. Окно локальных переменных IDE Visual Studio

По данному рисунку видно, что для того, чтобы просмотреть информацию о третьем элементе, необходимо сначала просмотреть данные всех предшествующих ему элементов, что может стать крайне неудобным при значительном увеличении количества элементов в списке. Также в данном отображении присутствует большое количество дополнительной информации, усложняющий поиск необходимых данных. Отображение, предлагаемое разработанным плагином, лишено вышеперечисленных недостатков.

Заключение

Конечно, разработанный плагин требует дальнейшего развития в плане удобства, разработки способов наглядного отображения больших структур данных, добавления различных представлений и возможностей

настройки отображения и т.д. Но уже сейчас во многих ситуациях, визуализация, выполняемая разработанным решением, нагляднее стандартных средств визуализации Visual Studio. Кроме того, решена чисто техническая, но достаточно сложная в плане изучения и тестирования задача по интеграции разработанного решения с IDE Visual Studio.

Список литературы

1. Kehoe C. Rethinking the evaluation of algorithm animations as learning aids: an observational study / C. Kehoe, J. Stasko, A. Taylor // *International Journal of Human-Computer Studies*. – 2001. – № 54. – P. 265-284.
2. Moreno A. Visualizing Programs with Jeliot 3 / A. Moreno [and all.] // *Proceedings of the working conference on Advanced visual interfaces*. – 2004. – P. 373-376.
3. Quintero C. Visual Studio – Creating Extensions for Multiple Visual Studio [Электронный ресурс]. – Режим доступа : <https://msdn.microsoft.com/magazine/mt493251>
4. Официальный сайт Pythontutor [Электронный ресурс]. – Режим доступа : <http://pythontutor.com/>
5. Официальный сайт Data Display Debugger [Электронный ресурс]. – Режим доступа : <https://www.gnu.org/software/ddd/>

СОДЕРЖАНИЕ

Алексеева М. Ю., Власов С. В. РЕАЛИЗАЦИЯ АЛГОРИТМА ОБУЧЕНИЯ БАЙЕСОВСКИХ НЕЙРОННЫХ СЕТЕЙ НА ОСНОВЕ КОРРЕКЦИИ ОШИБКИ	1
Бердников О. В., Чекмарёв А. И. РАЗРАБОТКА МОДУЛЕЙ КОРПОРАТИВНОЙ ТЕЛЕФОНИИ НА БАЗЕ ОТКРЫТЫХ ПЛАТФОРМ FREESWITCH И KAZOO 2600 NZ	8
Бобков Р. И., Зуев С. А. РАЗРАБОТКА ОКРУЖЕНИЯ РАБОЧЕГО СТОЛА АДАПТИРОВАННОГО ДЛЯ ЛЮДЕЙ ПОЖИЛОГО ВОЗРАСТА	13
Бобков Р. И., Зуев С. А. GARBAGE COLLECTION – ОБЛЕГЧЕНИЕ РАБОТЫ ПРОГРАММИСТА?.....	18
Бобков Р. И., Зуев С. А. СЛОЖНОСТЬ РАЗРАБОТКИ ASIC ПОД АЛГОРИТМЫ SCRYPTONIGHT И ETHASH И ЧЕМ ГРОЗИТ РЕАЛИЗАЦИЯ ПОДОБНЫХ УСТРОЙСТВ	25
Бойкова Н. А., Гаршина В. В. АНАЛИЗ КОРПОРАТИВНЫХ ДАННЫХ НА ОСНОВЕ ТЕХНОЛОГИЙ DATA MINING В СРЕДЕ RSTUDIO	31
Борисов А. Д., Соломатин Д. И. РАЗРАБОТКА ПРИЛОЖЕНИЯ ДЛЯ РАЗМЕТКИ КОНТУРОВ ЛИЦ	38
Бородин А. В., Крыловецкий А. А. ПРИМЕНЕНИЕ СВЕРТОЧНЫХ НЕЙРОННЫХ СЕТЕЙ ДЛЯ РЕШЕНИЯ ЗАДАЧИ РАСПОЗНАВАНИЯ ИЗОБРАЖЕНИЙ	43
Вишнякова В. В., Атанов А. В. СООТВЕТСТВИЕ КЛАСИФИКАЦИЙ ГОЛОМОРФНО-ОДНОРОДНЫХ ПОВЕРХНОСТЕЙ В S^2 И ТРЁХМЕРНЫХ АЛГЕБР ЛИ	49
Войтенко С. Р., Нечаев Ю. Б., Стромов А. В. ОЦЕНКА ПРОПУСКНОЙ СПОСОБНОСТИ БЕСПРОВОДНОЙ СЕТИ С РЕГУЛЯРНОЙ ТОПОЛОГИЕЙ С ИСПОЛЬЗОВАНИЕМ БУФЕРИЗАЦИИ СООБЩЕНИЙ.....	54
Высоцкая Д. А., Илларионов И. В. СРАВНЕНИЕ МОДЕЛЕЙ МАССОВОГО ОБСЛУЖИВАНИЯ НА ПРИМЕРЕ ЧАТ В ПРОГРАММНОЙ СРЕДЕ ANYLOGIC	60
Вычиков Д. Д., Соломатин Д. И. ПОДХОДЫ К РАСПОЗНАВАНИЮ ИНДЕНТИФИКАТОРОВ В ЗАДАЧЕ НАПИСАНИЯ КОДА ПРИ ПОМОЩИ ГОЛОСА	66
Галыгина М. А., Киселев Е. А. НОВЫЕ СПОСОБЫ ИНТЕРПОЛЯЦИИ С ПОМОЩЬЮ ОКОННЫХ СИСТЕМ ФУНКЦИЙ	71

Гетманова К. Г., Матвеев М. Г. ФОРМИРОВАНИЕ ИНТЕГРАЛЬНЫХ ОЦЕНОК АЛЬТЕРНАТИВ НА ОСНОВЕ АНАЛИЗА ИХ ПАРАМЕТРОВ.....	76
Гребенщикова Е. А., Клиньских А. Ф. ИССЛЕДОВАНИЕ ЭФФЕКТА ГИББСА ПРИ ВОССТАНОВЛЕНИИ РАЗРЫВНЫХ ФУНКЦИЙ.....	83
Грибкова М. А., Вялых С. А. ИССЛЕДОВАНИЕ УГРОЗ ПОВЫШЕНИЯ ПРИВИЛЕГИЙ В ВЫЧИСЛИТЕЛЬНОЙ СЕТИ ВОРОНЕЖСКОГО ГОСУДАРСТВЕННОГО УНИВЕРСИТЕТА	89
Данилова А. В., Мещеряков И. Г., Господарикова В. С., Зуев С.А. МАГНИТОМЕТР В ТЕХНИКЕ И МЕДИЦИНЕ.....	96
Данилова А. В., Мещеряков И. Г., Господарикова В. С., Зуев С.А. МАГНИТОМЕТР ДЛЯ ГЕОЛОГОРАЗВЕДКИ.....	100
Данковцева А. Ю., Гаршина В. В. СИСТЕМА ОЦЕНКИ ИНФОРМАЦИОННОЙ ЗАЩИЩЕННОСТИ КОМПАНИИ НА ОСНОВЕ ИНТЕГРАЦИИ РАЗНОРОДНЫХ ДАННЫХ И ПОСТРОЕНИЯ ВЫВОДОВ ПО ОНТОЛОГИИ СТАНДАРТА ISO/IEC 27001.....	105
Данковцева Е. С., Митрофанова Е. Ю. РАЗРАБОТКА БИОМЕТРИЧЕСКОЙ СИСТЕМЫ ДЛЯ АУТЕНТИФИКАЦИИ ПОЛЬЗОВАТЕЛЯ ПО КЛАВИАТУРНОМУ ПОЧЕРКУ	112
Добрынин Е. А., Крыловецкий А. А. АДАПТИВНЫЙ АЛГОРИТМ РАЗДЕЛЕНИЯ СОПРИКАСАЮЩИХСЯ ЗЕРЕН НА ЦИФРОВОМ ИЗОБРАЖЕНИИ.....	117
Землянхун М. В., Соломатин Д. И. ЭФФЕКТИВНЫЙ ПОИСК СООТВЕТСТВИЙ МЕЖДУ ТРЕХМЕРНЫМИ ОБЪЕКТАМИ МЕТОДОМ ТРАНСФЕРА ФУНКЦИЙ.....	124
Зимин Д. А., Борисов Д. Н. ИССЛЕДОВАНИЕ ПОМЕХОУСТОЙЧИВОСТИ АЛГОРИТМА MIMO ДЛЯ BPSK И 8- PSK МОДУЛЯЦИИ	131
Ивашкина Е. Е., Вялых С. А. ИССЛЕДОВАНИЕ И ОЦЕНКА УГРОЗ БЕЗОПАСНОСТИ БЕСПРОВОДНЫХ WI-FI СЕТЕЙ (НА БАЗЕ СТАНДАРТА IEEE 802.11).....	138
Игнатов Д. В., Илларионов И. В. МОДЕЛИРОВАНИЕ РИСКОВ СКЛАДСКИХ ПРОЦЕССОВ В ПРОГРАММНОЙ СРЕДЕ ANYLOGIC .	145
Измайлов Г. В., Борзунов С. В. РЕАЛИЗАЦИЯ АЛГОРИТМОВ ОБРАБОТКИ ИЗОБРАЖЕНИЙ В ИНФОРМАЦИОННОЙ СИСТЕМЕ ДЛЯ СОЗДАНИЯ ПАНОРАМ	151
Капустина Д. А., Матвеев М. Г. РАСПРЕДЕЛЕНИЕ ПРОГРАММНЫХ ЗАДАЧ ПО ФИЗИЧЕСКИМ МОДУЛЯМ В АВИОНИКЕ	156

Караблин А. В., Стадная Н. П. РАЗРАБОТКА ПРОГРАММНОГО МОДУЛЯ ДЛЯ ИССЛЕДОВАНИЯ И АНАЛИЗА ПРОЦЕССА ОРНШТЕЙНА-УЛЕНБЕКА.....	161
Карпов Е. А., Семенов М. Е. МОДИФИЦИРОВАННАЯ СИСТЕМА ЛОРЕНЦА.....	168
Козориз Д. В., Лобода А. В. ЗАДАЧА О ДВИЖЕНИЯХ ОДНОГО КЛАССА ПЛОСКИХ ФИГУР	173
Королёв Д. А., Соломатин Д. И. ОПТИМИЗАЦИЯ МЕТОДА ОБРАТНОЙ ТРАССИРОВКИ ЛУЧЕЙ С ПОМОЩЬЮ VCN-ДЕРЕВА ...	179
Лыгин В. С., Дрюченко М. А. РАЗРАБОТКА АЛГОРИТМОВ ОЦЕНКИ ХАРАКТЕРА СОДЕРЖИМОГО РАСТРОВЫХ ИЗОБРАЖЕНИЙ.....	184
Милованова А. И., Сирота А. А. БИОМЕТРИЧЕСКАЯ ИДЕНТИФИКАЦИЯ ПОЛЬЗОВАТЕЛЯ НА ОСНОВЕ РАСПОЗНАВАНИЯ ИЗОБРАЖЕНИЙ РАДУЖНОЙ ОБОЛОЧКИ ГЛАЗА С ИСПОЛЬЗОВАНИЕМ ГЛУБОКИХ НЕЙРОННЫХ СЕТЕЙ....	190
Морозов Е. Ю., Минин Л. А. МОДЕЛИРОВАНИЕ ЭКВИДИСТАНТНОЙ ТРЕХЭЛЕМЕНТНОЙ РЕШЕТКИ В УСЛОВИЯХ ВЛИЯНИЯ ПОДСТИЛАЮЩЕЙ ПОВЕРХНОСТИ	196
Мубаракшин Д. Э., Сирота А. А. ПРИМЕНЕНИЕ ГИБРИДНЫХ И НЕЙРОСЕТЕВЫХ МОДЕЛЕЙ ДЛЯ ПОСТРОЕНИЯ РЕКОМЕНДАТЕЛЬНЫХ СИСТЕМ	202
Новиков В. А., Соломатин Д. И. РАЗРАБОТКА ТРАНСЛЯТОРА КОДА С ЯЗЫКА МАТЛАВ В КОД НА ЯЗЫКЕ RYTHON.....	207
Обручников А. В., Соломатин Д. И. ЦИФРОВОЙ АККРЕДИТИВ С ИСПОЛЬЗОВАНИЕМ ТЕХНОЛОГИИ БЛОКЧЕЙН	215
Орехов Н. С., Клиньских А. Ф. МОДЕЛЬ ПЕРЕДАЧИ ИНФОРМАЦИИ В КВАНТОВОЙ СЕТИ.....	221
Пауков Н. В., Зуев С. А. ОПРЕДЕЛЕНИЕ МЕСТОПОЛОЖЕНИЯ В ПРОСТРАНСТВЕ С ИСПОЛЬЗОВАНИЕМ ТЕХНОЛОГИИ GPS	226
Пауков Н. В., Зуев С. А. GPS-ТРЕКЕР С ИСПОЛЬЗОВАНИЕМ GSM И ARDUINO	231
Погребецкая Е. В., Абрамов И. В. ФОРМАЛИЗАЦИЯ И РЕШЕНИЕ ПРОИЗВОДСТВЕННЫХ ЗАДАЧ В КОМПАНИИ «АНАЛИТИКА. КОНСАЛТИНГ. БЕЗОПАСНОСТЬ».....	238
Полещук Х. А., Тюкачев Н. А. ЗАХВАТ ВИДЕО ПОТОКА И ПРЕОБРАЗОВАНИЕ ЦВЕТОВОЙ МОДЕЛИ ЕГО КАДРОВ	243
Пузатых М. С., Дрюченко М. А. РАЗРАБОТКА ПРОГРАММНОГО МОДУЛЯ ДЛЯ АУГМЕНТАЦИИ ДАННЫХ НА ОСНОВЕ ВНЕСЕНИЯ ИСКУССТВЕННЫХ ИСКАЖЕНИЙ НА РАСТРОВЫЕ ИЗОБРАЖЕНИЯ	249

Пывина М. В., Атанов А. В. РЕШЕНИЕ УРАВНЕНИЯ БЛЭКА-ШОУЛЗА С ИСПОЛЬЗОВАНИЕМ РАДИАЛЬНЫХ БАЗИСНЫХ ФУНКЦИЙ	256
Решетняк Е. В., Коваль А. С. ИССЛЕДОВАНИЕ САМОПОДОБИЯ ТРАФИКА БЕСПРОВОДНЫХ СЕТЕЙ	260
Рзуки А. М., Нечаев Ю. Б. ПОВЫШЕНИЕ ЭФФЕКТИВНОСТИ СОТОВЫХ СЕТЕЙ ПОДВИЖНОЙ РАДИОСВЯЗИ НА ПРИМЕРЕ СТАНДАРТА GSM	267
Рощупкина О. А., Власов С. В. АНАЛИЗ ИТ-РИСКОВ С ИСПОЛЬЗОВАНИЕМ ТЕОРИИ БАЙЕСА	276
Рыбалкин Е. А., Клинский А. Ф. АНАЛИЗ И ВИЗУАЛИЗАЦИЯ АЛГОРИТМА КВАНТОВОГО РАСПРЕДЕЛЕНИЯ КЛЮЧЕЙ ВВ84	283
Сивоплясов Т. М., Вахитова Е. В. ОБ ОДНОМ ПРИЛОЖЕНИИ ФУНКЦИИ МЕБИУСА	288
Смотров С. В., Гаршина В. В., Копыгина Е. А. ОЦЕНКА ПОПУЛЯРНОСТИ ФАЙЛОВЫХ РЕСУРСОВ КОРПОРАТИВНОЙ ИНФОРМАЦИОННОЙ СЕТИ	294
Соколова В. А., Гавшин А. Л. РАЗРАБОТКА ПЛАГИНА К ANDROIDSTUDIO ДЛЯ ГЕНЕРАЦИИ СЕРВИСНОГО СЛОЯ ANDROID-ПРИЛОЖЕНИЙ ПО СПЕЦИФИКАЦИИ API BLUEPRINT ..	301
Сотников А. Н., Коваль А. С. АРХИТЕКТУРА СИСТЕМЫ ИДЕНТИФИКАЦИИ P2P ТРАФИКА	306
Старкина М. В., Гаршина В. В. ПРИМЕНЕНИЕ VDD-ПОДХОДА В ЗАДАЧЕ КРОСС-ПЛАТФОРМЕННОГО АВТОМАТИЗИРОВАННОГО ТЕСТИРОВАНИЯ МОБИЛЬНЫХ ПРИЛОЖЕНИЙ	313
Супонев Е. Г., Киселев Е. А. УСТОЙЧИВОСТЬ СИСТЕМ РАВНОМЕРНЫХ СДВИГОВ, ПОРОЖДЕННЫХ КОМПЛЕКСАМИ ИЗ ФУНКЦИЙ ГАУССА	318
Суходолов Д. М., Крыловецкий А. А. КЛАССИФИКАЦИЯ ЭЛЕМЕНТОВ ЗЕРНОВЫХ СМЕСЕЙ МЕТОДАМИ ГЛУБОКОГО ОБУЧЕНИЯ	323
Ушаков В. А., Донских Д. П., Теплякова М. А., Зуев С. А. СОЗДАНИЕ АЛЬТИМЕТРА	329
Ушаков В. А., Донских Д. П., Теплякова М. А., Зуев С. А. СОЗДАНИЕ ДОМАШНЕЙ МЕТЕОСТАНЦИИ С НИЗКОЙ СЕБЕСТОИМОСТЬЮ ...	335
Ушков Н. В., Тюкачев Н. А. МНОГОУРОВНЕВЫЙ МАСШТАБИРУЕМЫЙ ГРАФ ДЛЯ ЭЛЛЕКТРОННЫХ КАРТ	341
Черепанова С. В., Тюкачев Н. А. РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ЭКСПЕРТНОЙ СИСТЕМЫ НА ОСНОВЕ МЕХАНИЗМА НЕЧЕТКОГО ВЫВОДА	345

Шевцов Д. Е., Стромов А. В. НАТУРНОЕ МОДЕЛИРОВАНИЕ ЭЛЕМЕНТОВ БЕСПРОВОДНОЙ ОДНОРАНГОВОЙ ЯЧЕИСТОЙ СЕНСОРНОЙ СЕТИ НА БАЗЕ ARDUINO	353
Шмелев М. А., Толстобров А. П. РАЗРАБОТКА ПЛАГИНОВ ДЛЯ РЕАЛИЗАЦИИ НОВЫХ ТИПОВ ТЕСТОВ В СИСТЕМЕ MOODLE	358
Юшин А. В., Храмов В. Ю. АВТОМАТИЗИРОВАННАЯ СИСТЕМА ВЕДЕНИЯ ТЕМПОРАЛЬНОЙ БАЗЫ ДАННЫХ СИСТЕМЫ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ ПО ИДЕНТИФИКАЦИИ БЕСПИЛОТНЫХ ЛЕТАТЕЛЬНЫХ АППАРАТОВ	363
Яковлева Ю. А., Соломатин Д. И. ПЛАГИН ДЛЯ НАГЛЯДНОЙ ВИЗУАЛИЗАЦИИ ИЗМЕНЕНИЯ ПЕРЕМЕННЫХ И СТРУКТУР ДАННЫХ ВО ВРЕМЯ ОТЛАДКИ	370