



Научно-технический журнал

Технологии инженерных и информационных систем

Выпуск 3

2018 г.

Technologies of engineering and information systems

Издательство

Межрегиональная общественная
организация «Московское научно-
техническое общество
радиотехники, электроники и связи
Имени А.С. Попова»
МНТОРЭС им. А.С. Попова
Тел.:(7-499)948-0655
E-mail: mntores@mail.ru

Состав редколлегии

Главный редактор – д-р техн. наук, ст. науч. сотр. Слукин Г.П.;
отв. секретарь – канд. техн. наук Шумов А.В.;
д-р техн. наук, доц. Нефедов С.И.; д-р техн. наук, проф. Зимин В.Н.;
д-р техн. наук, проф. Чернышев С.Л.; канд. техн. наук Белоус В.В.;
академик МАС Смольская Н.Н.

Научно-технический журнал «Технологии инженерных и информационных систем» является периодическим рецензируемым научным изданием, которое отражает оригинальные научные результаты теоретических и прикладных исследований, проводимых по широкому кругу проблем в области машиностроения и приборостроения. Тематика издания также включает проблемы информатики, управления техническими системами и математического моделирования технических систем, а также общие и комплексные проблемы естественных и точных наук.

Содержание

	стр.
Баранова Е.А., Ничушкина Т.Н. Проблемы формирования отчетов в системе ИнфраМенеджер.....	3
Самарёв Р.С., Иванов А.Г. Анализ архитектур сетей нейронных для сегментации изображений	9
Минитаева А.М., Петров И.Д. Структурирование исходных данных археологических раскопок	18

Гуренко В.В., Бычков Б.И. Особенности преобразования тригонометрических спектров в базис обобщенных функций Хартли	24
Ничушкина Т.Н., Круглов Д.С. Анализ возможности применения технологии дополненной реальности для навигации в торговом центре	33
Попов А.Ю., Евдокимов А.А. Классификация критических ситуаций летательного аппарата на основе автоматического анализа данных	40
Гуренко В.В., Гузий Е.А., Федоренков В.В. Анализ моделей классификации текстовой информации.....	54
Ким Т.А., Гуренко В.В., Сотников А.А. Обработка цифровых сигналов в Уолше-подобном базисе.....	64
Иванова Г.С., Головков А.А., Лоншакова К.А. Анализ методов предобработки рентгеновских снимков.....	79
Попов А.Ю., Колотовкин М.И., Куклина Н.И. Система управления виртуальной моделью инвалидной коляски в 3D-тренаже при помощи интерфейса «мозг-компьютер».....	86
Хартов В.Я., Белкина Т.А. Акустический процессор формантного синтезатора речи с фонемным возбуждением.....	99
Пугачев Е.К., Лапина Н.А. Подход к проектированию механизма обновления баз правил.....	110
Ничушкина Т.Н., Гузий Е.А., Федоренков В.В. Сравнительный анализ способов конвертирования структур данных между базами данных платформы «1С: Предприятие».....	115
Пугачев Е.К., Хорунжина К.С. Сравнение мирового и российского опыта противодействия отмывания денежных средств и финансового терроризма.....	123
Гуренко В.В., Букша К.В., Цаплин С.Т. Сравнительный анализ контейнеров библиотек Qt и STL по критерию времени.....	129

Проблемы формирования отчетов в системе ИнфраМенеджер

¹ Баранова Е.А., ² Ничушкина Т.Н.

МГТУ им. Н.Э. Баумана (НИУ), Москва, Россия

bevgeniyaa@mail.ru

tnich@rambler.ru

Аннотация: В статье выполнен анализ структуры системы ИнфраМенеджер и возможностей ее составных частей. Проанализированы возможности и особенности системы при создании статистических отчетов. Выявлены проблемы, не позволяющие генерировать отчеты определенных видов и форм. Для решения возникающих проблем предложено разработать специальную подсистему и сформулированы требования к ней. Обозначен круг решаемых подсистемой задач. Проанализированы сложности реализации взаимодействия подсистемы с базой системы ИнфраМенеджер и предложены пути их устранения.

Ключевые слова: база данных, обмен файлами, удаленный вызов, асинхронный обмен сообщениями, компонентная объектная модель.

Problems of formation of reports in the Inframanager system

Baranova E.A., Nychyshcina T.N.

Bauman Moscow State Technical University, Russia

bevgeniyaa@mail.ru

tnich@rambler.ru

Annotation: In article the analysis of structure of the Inframenedzher system and opportunities of her components is made. Opportunities and features of system are analysed during creation of statistical reports. The problems which aren't allowing to generate reports of certain types and forms are revealed. It is offered to develop a special subsystem for the solution of the arising problems and requirements to her are formulated. The circle of the tasks solved by a subsystem is designated. Difficulties of realization of interaction of a subsystem with base of the Inframenedzher system are analysed and ways of their elimination are offered.

Keywords: database, file exchange, remote call, asynchronous exchange of messages, component object model.

¹Баранова Евгения Александровна, студентка кафедры «Компьютерные системы и сети» (ИУ6) МГТУ им. Н.Э. Баумана

²Ничушкина Татьяна Николаевна кандидат технических наук, доцент кафедры «Компьютерные системы и сети» (ИУ6) МГТУ им. Н.Э. Баумана. SPIN-код 8681-8433

Введение

Для эффективности работы комплекса информационных технологий по обслуживанию пользователей предприятия важную роль играет оперативное реагирование руководства, ответственных подразделений и сотрудников на инциденты. Для этого необходимо своевременно оповещать их посредством информационных отчетов в реальном времени и преподносить информацию в определенном формате с заданными характеристиками. Поэтому на предприятиях существуют различные специальные ИТ системы, которые призваны решать все эти вопросы. Одной из таких систем является информационная система ИнфраМенеджер, которая может предоставлять руководству и ответственным сотрудникам статистику по системам вычислительной техники, по заявкам и так далее. Однако, в некоторых случаях возникают проблемы при использовании встроенной в систему статистики.

1. Анализ возможностей системы ИнфраМенеджер

Для обслуживания пользователей, учета и управления ИТ-активами, комплексного управления ИТ на предприятии используется система ИнфраМенеджер. Решение этих задач система ИнфраМенеджер осуществляет по трем направлениям:

- ITSM(IT Service Management) для комплексной автоматизации ИТ-деятельности и постановки ИТ-процессов;
- ИТ-активы для учета ИТ-имущества и управления ИТ-инфраструктурой.
- Service desk для автоматизации всего цикла обслуживания пользователей на всех линиях поддержки;

ITSM система ИнфраМенеджер предназначена для комплексной автоматизации ИТ-деятельности и постановки ИТ-процессов в соответствии с мировой методологией ITSM. ITIL(IT Infrastructure Library) – общепризнанная практика постановки ИТ-процессов в компании, основанная на ITSM [1].

ИнфраМенеджер ИТ-активы предназначен для учета ИТ-имущества и управления ИТ-инфраструктурой на предприятии. ИнфраМенеджер ИТ-активы обеспечивает автоматизированную поддержку процессов IT Asset Management (ITAM) и управление ИТ-ресурсами на всем их жизненном цикле [2].

ИнфраМенеджер Service Desk – уникальная российская Сервис деск система для всех линий поддержки, которая не просто обеспечивает регистрацию обращений, а заметно сокращает время диагностики и устранения причин инцидентов. ИнфраМенеджер Service Desk построен на базе методологии ITIL/ITSM. В отличие от традиционных help desk систем, ИнфраМенеджер Service Desk охватывает все линии поддержки, объединяя их в единое информационное пространство [3]. Кроме того, статистика Сервис деск дает ИТ-руководителям возможность принимать взвешенные управленческие решения в контексте затрат на ИТ-инфраструктуру и перераспределения ИТ-ресурсов:

- какие ресурсы вовлечены в обслуживание каких сервисов / подразделений и т.п.;
- какие сотрудники перегружены;
- какие трудоемкие мероприятия требуют перераспределить нагрузку между сотрудниками Сервис деск и т.д.;
- какое оборудование требует немедленной замены /сервисного обслуживания.

On-line статистика в виде настраиваемых инструментальных панелей («виджетов») призвана управлять ИТ-специалистами через ключевые показатели эффективности. Она включает в себя более 40 встроенных отчетов, таких как загрузка специалистов, распределение заявок между специалистами и пр. Различные панели статистики предоставляют новые возможности для проектирования и просмотра панелей статистики в системе ИнфраМенеджер

Конструктор панелей имеет понятный интерфейс, позволяющий определять источник данных и привязывать данные к элементам, определять формат представления данных и структуру панели. Многие из этих обычно сложных задач могут выполняться через простые drag-and-drop операции.

В конструкторе панелей ИнфраМенеджер можно разделить элементы панелей статистики на следующие группы:

- Элементы визуализации данных – элементы используются для показа данных на панелях. Можно использовать элементы график, таблица, карточки, сводная таблица, диаграмма, фильтр диапазона, Image и Text Box для показа статических элементов панели;
- Элементы фильтрации – можно использовать раскрывающийся список, список, дерево выбора;
- Элементы группирования – элементы организуют пространство панели и позволяют управлять взаимодействием элементов внутри и вне группы.

2. Обзор проблем формирования статистических отчетов

Однако, как показал анализ групп элементов статистики, при использовании конструктора панелей нет возможности использовать поля для ввода требуемой для поиска или сортировки пользователю информации. Поэтому, при создании отчета необходимо предусматривать все возможные варианты для изменения данных выводимых в панели статистики. Для создания отчета, в котором необходимо представить информацию по часам, то для задания времени можно использовать поля таблиц с датой и временем, таким образом, представление привязывается к дате и времени, указанными в таблицах базы данных.

В базах данных системы ИнфраМенеджер присутствует определенное количество представлений (view), которые могут быть полезны при создании новых панелей статистики. Представления упрощают этап создания источника данных в части

формирования запроса к базе данных. По сути, данные представления являются преднастроенными запросами, которые возвращают информацию из различных таблиц базы данных, и объединенные по разным тематикам. Представления доступны в меню «Конструктора запросов», под списком доступных таблиц [4].

В ходе анализа представлений было выяснено, что иногда необходимых полей в них нет. Тогда приходится искать необходимые поля в таблицах баз данных системы, которых насчитывается более 500 штук.

В настоящее время разработано много программ для сбора статистики и формирования отчетов, но большинство из них ориентированы на сбор статистики и формирование отчетов в финансовой сфере или не подходит для применения на промышленном предприятии.

Кроме того, в процессе работы с конструктором панелей системы были выявлены некоторые недостатки, которые затрудняют создание статистики и отчетов:

- Для формирования более сложных отчетов нужно знания языка SQL-запросов, что требует определенной квалификации пользователя.
- Можно использовать только данные, занесенные в базу данных системы.
- Представление данных в панелях должно соответствовать структуре базы данных в виде таблиц, определенных столбцами, что требует использовать поля столбца таблицы в качестве заголовков для представления данных в панелях статистики.

Например, для создания отчета, в котором необходимо представить количество заявок, находящихся в работе, по часам. В этом случае, для задания времени можно использовать поля таблиц с датой и временем (время регистрации заявки, время закрытия заявки или время изменения заявки). Следовательно, для формирования отчета можно использовать, например, время регистрации заявки. Однако, возникает сложность подсчета заявок, так как в приведенном примере, заявка будет учитываться только один раз, в тот час, когда она была зарегистрирована. Однако, в следующий час система уже не включает эту заявку в подсчет, хотя она все еще в работе.

3. Анализ путей решения проблем формирования отчетов

Для устранения указанных недостатков и обеспечения возможности формирования универсальных отчетов, было принято решение разработать дополнительную подсистему формирования отчетов на базе системы ИнфраМенеджер.

Разрабатываемая подсистема должна:

- быть простой в использовании и не требовать для формирования отчета знаний языков программирования.
- обеспечивать выбор в меню вид отчета или шаблон для его формирования.
- иметь средства ввода необходимых параметров и их диапазона для выбранных форм отчета;

- обеспечивать сохранение отчета в виде шаблона или в виде документа необходимого типа;
- предусматривать возможность печати отчета по желанию пользователя.

Реализация подсистемы, обеспечивающей указанные возможности, требует решения некоторых проблем. Одной из таких проблем является взаимодействие разрабатываемой подсистемы с системой ИнфраМенеджер. Разрабатываемая подсистема должна использовать данные хранящиеся в базе данных системы ИнфраМенеджер, но при этом она не может вносить изменения в данные базы данных.

Для взаимодействия приложений обычно используются такие методы, как обмен файлами, общая база данных, удаленный вызов и асинхронный обмен сообщениями [5].

В настоящее время выделяют следующие наиболее распространенные технологии реализации метода удаленного вызова процедур в рамках систем обмена сообщениями:

- компонентная объектная модель Component Object Model (COM), разработанная корпорацией Microsoft;
- общая архитектура брокеров объектных запросов Common Object Request Broker Architecture (CORBA) [6];
- веб-сервисы, в том числе, ESB и SOAP.

Выбор конкретной архитектуры зависит от специфики связываемых информационных систем, например, в случае веб-приложений наиболее эффективным будет последний вариант. Однако если сопрягаемые информационные системы функционируют как настольные приложения, что часто распространено среди корпоративных информационных систем, наиболее уместным будет их интеграция на основе взаимодействующих компонент, каждая из которых может использоваться во многих информационных системах одновременно. Поскольку информационные системы предприятия, которые необходимо интегрировать между собой, функционируют на базе операционной системы MS Windows, для их программного сопряжения целесообразно использовать методологию и средства COM-технологии. Данный вывод обусловлен тем, что технология CORBA, главным образом, предназначена для сопряжения систем в гетерогенных средах, когда сопрягаемые приложения функционируют на базе различных операционных систем (Linux, Windows, Mac OS и т.д.)

Для решения поставленных задач необходимо:

- выполнить анализ структур данных базы ИнфраМенеджер,
- провести анализ особенностей взаимодействия с базой;
- разработать технологию такого взаимодействия
- разработать интерфейс для формирования отчетов разного вида и формата;
- реализовать систему.

Заключение

В настоящее время с целью решения поставленной задачи был проведен анализ структур данных базы ИнфраМенеджер и особенностей взаимодействия с базой, в ходе которого были проанализированы возможные варианты взаимодействия приложений, определена модель данных, которая лежит в их основе, базовое программное обеспечение — СУБД, сервер приложений, модель бизнес-процессов.

В ходе исследования особенностей работы системы ИнфраМенеджер было выяснено, что для создания отчетов встроенным конструктором панелей статистики системы ИнфраМенеджер нет возможности настроить панель статистики для создания автоматизированных универсальных отчетов. Также в конструкторе необходимо знать SQL-язык для формирования запросов к базе данных для более удобной выборки данных в отчет.

В результате анализа отчетов была выделена часть видов отчетов, которая может быть сформирована при помощи подсистемы формирования автоматизированных универсальных отчетов. Для этих видов отчетов проведен анализ необходимых таблиц из базы данных и полей из них. Также были рассмотрены возможные интерфейсы отчетов.

Список литературы

1. ITSM система ИнфраМенеджер. [Электронный ресурс]. Режим доступа: <http://www.inframanager.ru/solutions/itsm/> (Дата обращения: 24.03.2018)
2. ИнфраМенеджер ИТ-активы. [Электронный ресурс]. Режим доступа: <http://www.inframanager.ru/solutions/itam/> (Дата обращения: 24.03.2018)
3. Service Desk ИнфраМенеджер. [Электронный ресурс]. Режим доступа: <http://www.inframanager.ru/solutions/service-desk/> (Дата обращения: 24.03.2018)
4. Документация ИнфраМенеджер. [Электронный ресурс]. Режим доступа: <http://www.inframanager.ru/download/documents/> (Дата обращения: 28.03.2018)
5. Добровольский А. Интеграция приложений: методы взаимодействия, топология, инструменты. // Журнал «Открытые системы. СУБД». 2006. № 09
6. CORBA. [Электронный ресурс]. Режим доступа: <http://www.corba.org>. (Дата обращения: 28.03.2018)
7. Хоп Г., Вульф Б. Шаблоны интеграции корпоративных приложений. Издательство: Вильямс. 2007. 672 с.
8. Боркус В. Методы и инструменты интеграции корпоративных приложений. М.: RC Group. 2005. 215 с.
9. Вичугова А.А., Вичугов В.Н., Дмитриева Е.А., Цапко Г.П. Методы и средства интеграции информационных систем в рамках единого информационного пространства проектирования // Вестник науки Сибири. 2012. № 5 (6). С. 113-115

Анализ архитектур нейронных сетей для сегментации изображений

³Самарёв Р.С., ⁴Иванов А.Г.

МГТУ им. Н.Э. Баумана (НИУ), Москва, Россия

samarev@acm.org

Ya.fakt@yandex.ru

Аннотация: для эффективного решения задач, связанных с сегментацией изображений, широкое применение получили нейронные сети. По сравнению с традиционными методами, применение нейронных сетей позволяет значительно увеличить скорость и, часто, точность сегментации. В настоящее время существует множество различных архитектур нейронных сетей для сегментации изображений. В работе приводится описание наиболее популярных архитектур и проведен их сравнительный анализ. Определены области предпочтительного применения рассмотренных нейронных сетей.

Ключевые слова: сегментация изображений, нейронные сети, Unet, DenseNet, ReSeg, SegNet, FCN-FF.

Analysis of neural network architectures for image segmentation

Samarev R.S., Ivanov A.G.

Bauman Moscow State Technical University, Russia

samarev@acm.org

Ya.fakt@yandex.ru

Abstract: neural networks are widely used for effective solution of problems related to image segmentation. Compared with traditional methods, the use of neural networks can significantly increase the speed and, often, the accuracy of segmentation. Currently, there are many different neural network architectures for image segmentation. The paper describes the most popular architectures and their comparative analysis. The areas of preferred application of the considered neural networks are defined.

Keywords: image segmentation, neural networks, Unet, DenseNet, ReSeg, SegNet, FCN-FF.

³Самарев Роман Станиславович, к.т.н., доцент кафедры «Компьютерные системы и сети» (ИУ6) МГТУ им. Н.Э. Баумана

⁴Иванов Алексей Геннадьевич, студент кафедры «Компьютерные системы и сети» (ИУ6) МГТУ им. Н.Э. Баумана

Введение

Сегментация изображений занимает значимое место в области анализа изображений, используется для выделения объектов и границ между ними на изображениях и находит применение в медицинской, военной, транспортной и многих других сферах. С точки зрения компьютерного зрения, сегментация является наиболее детальным методом анализа изображений. При анализе изображений простейшей задачей является классификация, которая позволяет определить отношение изображение к определенному классу. Решение задач классификация и локализация объекта, дополнительно к предыдущей задаче, определяет примерное место нахождения и тип объекта на изображении. Решение задачи поиска объектов позволяет найти несколько объектов, относящихся к различным классам и определить их примерное положение на картинке. Сегментация же находит все искомые объекты и их точные границы, позволяя получить о них такую информацию как цвет, форма, площадь и так далее. Наглядное сравнение различных задач анализа изображений показано на рисунке 1.

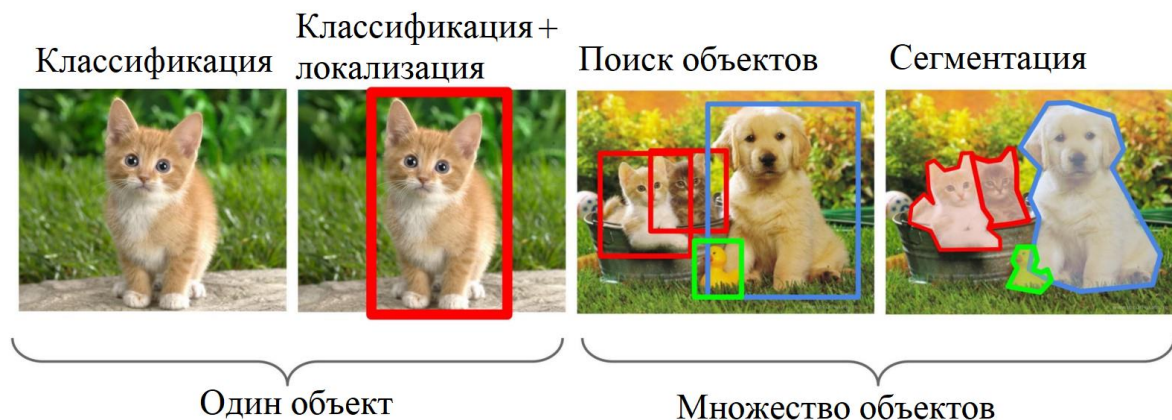


Рис. 1. Сравнение задач анализа изображений

Существует множество видов методов сегментации: основанные на порогах, границах, зонах, кластерах, графах [1]. Однако, в последнее время, большую популярность имеют решения, основанные на нейронных сетях, имеющие лучшую точность и скорость сегментации, по сравнению с традиционными методами [2].

В настоящее время имеется множество различных архитектур нейронных сетей для сегментации изображений, наиболее популярные из которых это SegNet, ReSeg, Unet, FCN-FF, DenseNet [4-9].

1. Анализ нейронных сетей для сегментации изображений

Перед проведением анализа стоит дать пояснение основным слоям, используемым в рассматриваемых нейронных сетях, так как термины одинаковы для разных сетей.

Свёрточный слой: операция свёртки, это, по сути, матричный фильтр, который трансформирует исходное изображение в какое-то другое, эту операцию можно повторять множество раз. Имеется матрица, которая называется ядром фильтра. Эта матрица поочередно накладывается на все части изображения, соответствующие элементы просто перемножаются, результаты складываются и записываются в центральную точку.

Слой субдискретизации: Операция субдискретизации — это просто усреднение либо взятие максимума. Операция работает на небольших квадратах, например, 2×2 . Это квадрат накладывается на изображение и, например, выбирается максимальный элемент из этого квадрата 2×2 , отправляется на выход. Таким образом уменьшается изображение. Взятие максимума дает небольшую инвариантность к смещениям. То есть становится не важно, какой-то признак нашелся в этой позиции или на 2 пикселя вправо. Эта операция позволяет нейросети быть чуть более устойчивой к сдвигам изображения.

1.1. Нейронная сеть DenseNet

Сверточные нейронные сети стали доминирующим подходом машинного обучения для визуального распознавания объектов. Хотя они были впервые представлены более 20 лет тому назад [3], улучшения в аппаратном обеспечении компьютеров и структурах сетей позволили тренировку по-настоящему глубоких свёрточных нейронных сетей только недавно. По мере того, как свёрточные нейронные сети становятся все более глубокими (то есть имеющими большее количество слоёв), возникают новые исследовательские задачи. По мере того как поданная на вход информация проходит через много слоев, она может быть утеряна к тому времени, когда она достигнет конца (или начала) сети.

Недавние работы показали, что сверточные сети могут быть значительно глубже, точнее и эффективнее в обучении если они содержат более короткие связи между слоями, близкими ко входу и выходу нейронной сети. Нейронная сеть DenseNet (Dense Convolutional Network) использует эту закономерность и соединяет каждый слой с каждым последующим слоем по ходу распространения сигнала [4]. Тогда как традиционные сверточные сети с количеством слоёв L имеют L связей — поочередно по одной между текущим и следующим за ним слоем, DenseNet имеет $L(L+1)/2$ связей. Для каждого слоя карты признаков всех предыдущих слоев используются в качестве входных данных, а собственные карты признаков используются в качестве входных данных во все последующие слои. На рисунке 2 представлено схематичное изображение уплотняющего блока.

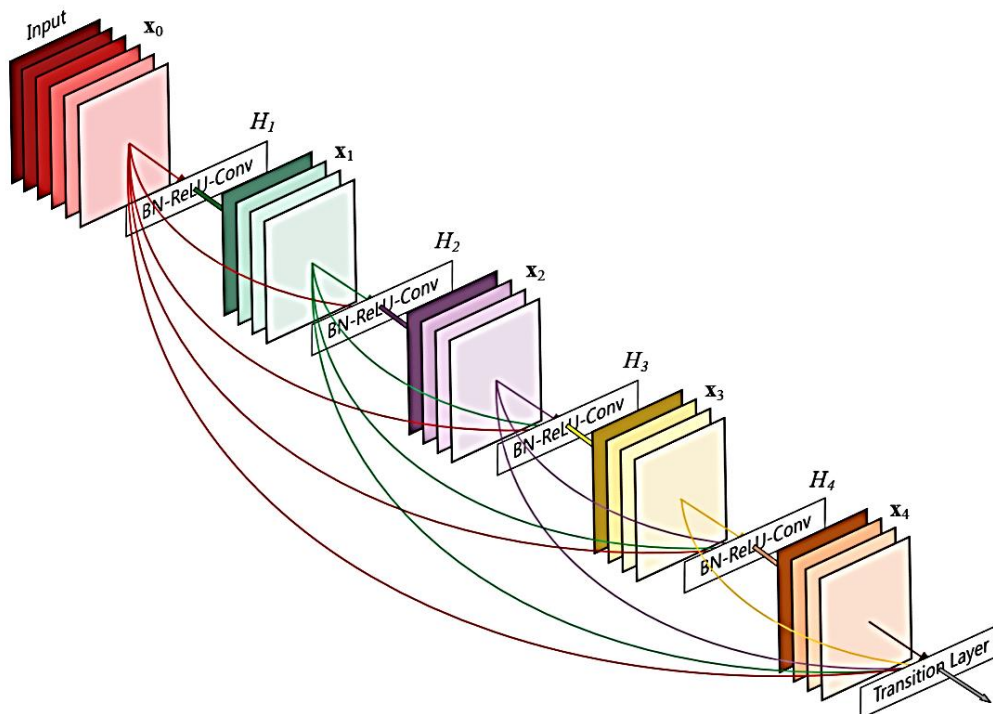


Рис. 2. Пятислойный уплотняющий блок. Каждый слой получает карты признаков всех предыдущих слоёв на вход

Нейронная сеть DenseNet состоит из нескольких слоёв свёртки, субдискретизации и уплотняющих (dense block – набор слоёв, каждый из которых имеет связь со всеми последующими) блоков. На рисунке 3 представлена DenseNet из трёх блоков.

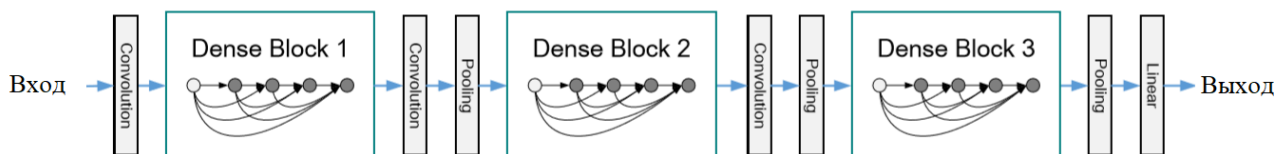


Рис. 3. Архитектура DenseNet с тремя блоками. Слои между двумя соседними блоками служат для изменения размеров карт признаков с помощью свёртки и субдискретизации

Основное преимущество DenseNet заключается в том, что она позволяет добиться очень высоких показателей точности распознавания. Так же, DenseNet проста в настройке, так как требуется выбрать малое число гиперпараметров (число слоёв L и темп роста K). Однако, из-за большого количества связей, DenseNet требует достаточно больших вычислительных мощностей.

1.2. Нейронная сеть Unet

Архитектура сети Unet представляет собой последовательность слоёв свёртки + субдискретизации, которые сначала уменьшают пространственное разрешение

картинки, а потом увеличивают его, предварительно объединив с данными картинки и пропустив через другие слои свёртки [5]. Таким образом, сеть выполняет роль своеобразного фильтра. Одно из главных преимуществ данной нейронной сети – она способно эффективно обучаться на небольшой обучающей выборке (150-200 изображений). Устройство нейронной сети Unet показано на рисунке 4.

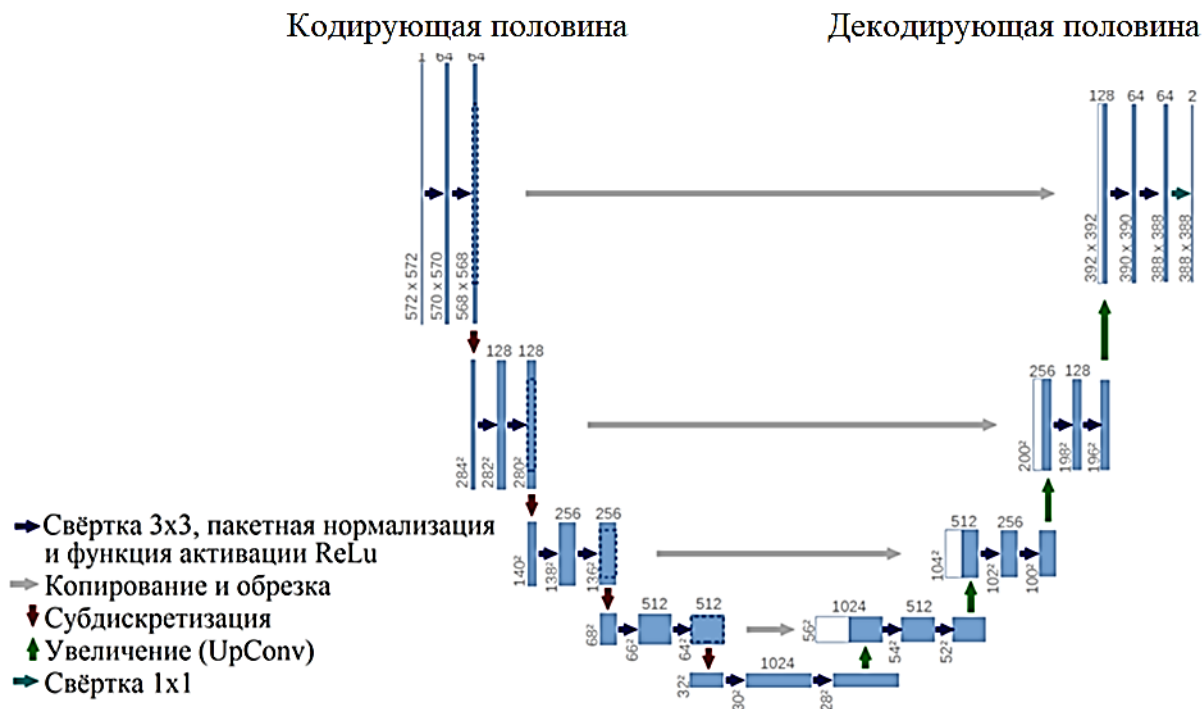


Рис. 4. Архитектура нейронной сети Unet

Unet состоит из кодирующей части (левая половина), и декодирующей (правая половина). Сжимающая часть идет по пути типичной сверточной нейронной сети. Она состоит из повторяемого применения двух сверток 3x3 и операции субдискретизации 2x2 с шагом 2. В каждом шаге субдискретизации мы удваиваем количество каналов признаков. Каждый шаг расширяющей части состоит из обратной операции субдискретизации и затем свертки 2x2, которая уменьшает количество каналов признаков вдвое, склеивания с соответствующим образом обрезанной картой признаков из сжимающей части, и двух сверток 3x3. Обрезка необходима из-за потери граничных пикселей в каждой свертке. На последнем уровне свертка 1x1 используется для сопоставления каждого 64-х компонентного вектора признаков с требуемым количеством классов. В сумме сеть имеет 23 сверточных слоя. Входные изображения и их соответствующие карты сегментации используются для обучения сети с реализацией стохастического градиентного спуска.

Основным преимуществом Unet является малое время сегментации, что позволяет применять данную нейронную сеть в системах с распознаванием в реальном времени. Из недостатков можно выделить низкую по сравнению с другими архитектурами точность сегментации.

1.3. Нейронная сеть SegNet

SegNet состоит из трёх частей: кодирующая часть, имеющая 13 слоёв свёртки, декодирующая часть и финальный слой классификации [6]. Отличительная особенность SegNet состоит в том, что дешифратор увеличивает поданные на его вход карты признаков низкого разрешения. Алгоритм работы SegNet следующий:

- изображение проходит через традиционный шифратор;
- дешифратор расширяет входное изображение, используя переданные индексы субдискретизации от соответствующего шифратора, чтобы создать разреженную карту признаков;
- дешифратор выполняет свертку для уплотнения карты признаков.
- окончательные карты признаков дешифратора подаются в классификатор softmax для классификации по пикселям.

Схема нейронной сети SegNet приведена на рисунке 5.

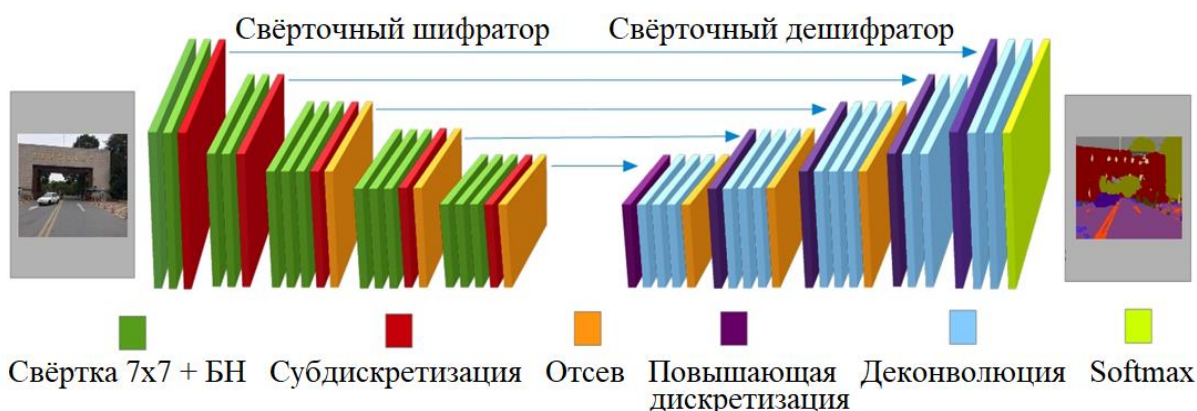


Рис. 5. Архитектура нейронной сети SegNet

Основными преимуществами SegNet является быстрое обучение, архитектура нетребовательна к ресурсам и позволяет получить хорошие результаты при небольшом обучающем наборе. Минусами является большое время сегментации, что делает невозможным применение в системах реального времени, и достаточно невысокая точность сегментации.

1.4. Нейронная сеть ReSeg

ReSeg – рекуррентная нейронная сеть. В рекуррентных нейросетях нейроны обмениваются информацией между собой: вдобавок к новой порции входных данных нейрон также получает некоторую информацию о предыдущем состоянии сети. Таким образом в сети реализуется «память» [7].

В нейронная сеть ReSeg основана на модели ReNet. Каждый слой ReNet состоит из четырёх рекуррентных нейронных сетей, которые проходят изображение по горизонтали справа налево и слева направо, и по вертикали, сверху вниз и снизу вверх [8].

После слоёв ReNet идет слой увеличения, чтобы восстановить оригинальное разрешение изображения, и слой классификации.

Схема сети ReSeg показана на рисунке 6. Первые два рекуррентных слоя (синий и зеленый) в ReSeg применяются на части изображения размером $2 \times 2 \times 3$, их карты признаков размером $16 \times 16 \times 256$ проходят свертку и подаются на вход следующим двум рекуррентным слоям (красный и желтый), которые формируют выход слоя. В конце идет слой увеличения и классификатор softmax.

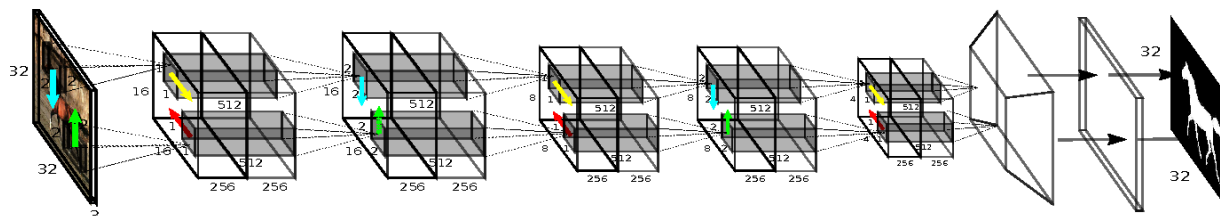


Рис. 6. Нейронная сеть ReSeg

Благодаря рекуррентной архитектуре, ReSeg обладает высокой точностью сегментации. Однако, данная архитектура обладает большим временем обучения и требует мощное оборудование.

1.5. Нейронная сеть FCN-FF

Нейронная сеть FCN-FF (deep fully convolutional network with feature fusion – глубокая полносвёрточная сеть со слиянием признаков) извлекает иерархические признаки из изображений, а затем сливает высокоуровневые и низкоуровневые признаки из глубокого и поверхностного слоёв для сегментации изображения [9]. Слияние различных уровней признаков повышает точность распознавания на сложных изображениях. Схема нейронной сети показана на рисунке 7.

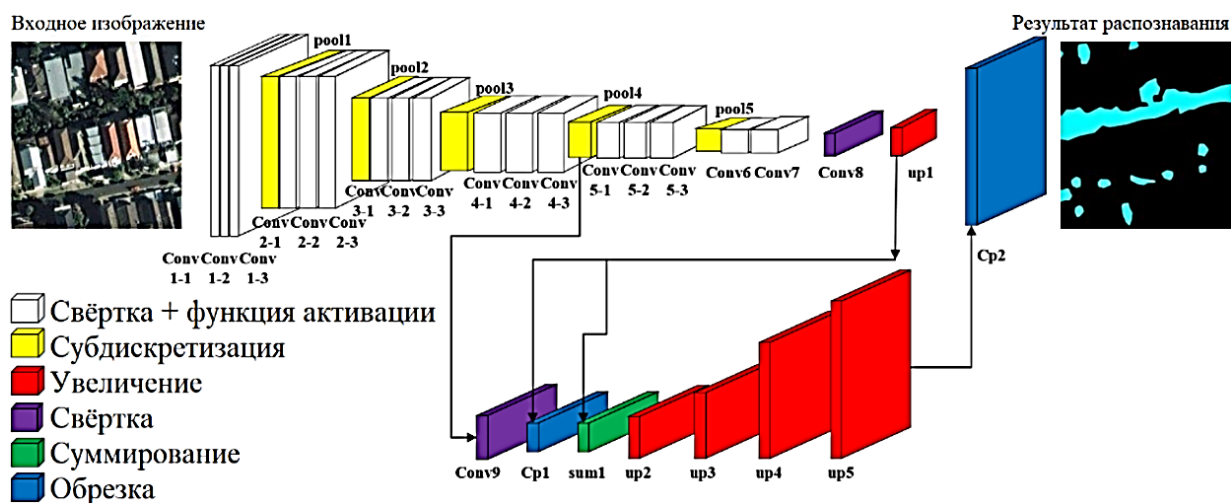


Рис. 7. Архитектура нейронной сети FCN-FF

Нейронная сеть FCN-FF обладает высокой точностью сегментации и хорошо справляется с изображениями, на которых находится большое количество объектов. Ещё одним преимуществом можно назвать высокую скорость работы. Однако, для обучения данной нейронной сети требуется достаточно большая обучающая выборка.

Заключение

Проведённый анализ позволяет сделать выводы о сильных и слабых сторонах каждой из рассмотренных архитектур. Для оценки используются баллы от 1 до 5, оцениваются время обучения (5 баллов – самое маленькое время обучения), скорость работы (5 баллов – самая быстрая), точность сегментации (5 баллов – максимальная точность). Результаты сравнения представлены в таблице 1.

Таблица 1.

Результаты сравнения архитектур нейронных сетей

Нейронная сеть	Время обучения	Скорость работы	Точность сегментации
DenseNet	2	3	5
Unet	4	5	2
SegNet	5	2	3
ReSeg	1	3	5
FCN-FF	2	4	4

По результатам исследования нельзя выявить однозначно лучшую нейронную сеть. Предпочтение одной из архитектур следует отдавать исходя из специфики конкретной задачи и её требований (скорость сегментации, точность, размер обучающей выборки и так далее). В частности, в случае, когда нужна максимальная точность сегментации при приемлемой скорости работы, к примеру, при анализе медицинских изображений, следует использовать ReSeg или DenseNet. U-Net хороша в тех случаях, когда можно немного пожертвовать точностью ради максимально быстрой работы, это будет полезно в системах, в которых распознавание происходит в реальном времени, таких как камеры анализа дорожной ситуации. Для этой задачи так же подойдёт FCN-FF, она будет работать чуть медленнее, но точнее выполнять сегментацию. SegNet очень медленна в работе, что делает её непригодной для решения задач распознавания в реальном времени, но благодаря самому лучшему времени обучения хороша, если нет возможности обучать сеть на мощном оборудовании.

Список литературы

1. Khan A.M. Image Segmentation Methods: A Comparative Study // International Journal of Soft Computing and Engineering. 2013. N 3. С. 2231-2307.
2. Long J., Shelhamer E. Fully Convolutional Networks for Semantic Segmentation //

IEEE Transactions on Pattern Analysis and Machine Intelligence. 2017. N 39. C. 640 – 651.

3. LeCun, Boser B, Denker J. S., Henderson D., Howard R. E. Backpropagation applied to handwritten zip code recognition // Neural computation. 1989. N 1. P. 541-551.

4. Huang G., Liu Z. Densely Connected Convolutional Networks // IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2017.

5. U-Net: Convolutional Networks for Biomedical Image Segmentation [Электронный ресурс] // Computer Science Department and BIOS Centre for Biological Signalling Studies, University of Freiburg. URL: <https://arxiv.org/pdf/1505.04597.pdf> (дата обращения: 19.04.2018).

6. Badrinarayanan V., Kendall A. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation // IEEE Transactions on Pattern Analysis and Machine Intelligence. 2017. N 39. C. 2481-2495.

7. Medsker L.R., Jain L.C. Recurrent neural networks design and application // CRC Press. 2001.

8. Visin F., Romero A., Cho K., Matteucci M. ReSeg: A Recurrent Neural Network-based Model for Semantic Segmentation // IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). 2016.

9. Fan S., Li Y., Guo L., Wang X. Vegetation Recognition based on Deep Learning with Feature Fusion // ICAIP Proceedings of the International Conference on Advances in Image Processing. 2017. P. 19-23.

Структурирование исходных данных археологических раскопок

⁵Минитаева А.М., ⁶Петров И.Д.

МГТУ им. Н.Э. Баумана (НИУ), Москва, Россия

alinaflower@rambler.ru

pivanetrov@gmail.ru

Аннотация: Статья посвящена исследованию материалов археологических раскопок и получению структуры исходных данных, которое является одной из задач прикладной исторической информатики. Предложен подход к созданию информационных систем, обрабатывающих данные археологических раскопок, отвечающие в должной мере критериям полноты и достоверности информации, а также репрезентативной визуализации, с учетом особенностей археологических материалов. Исследован спектр различных данных, которые получают в ходе и результате археологических раскопок. Результаты работы применимы как основа для крупномасштабной разработки информационных систем обработки материалов археологических раскопок.

Ключевые слова: археологические раскопки, памятники, историческая информатика, цифровая история, структура, исходные данные

Structuring of the initially obtained data of the archaeological excavations

Minitaeva A.M., Petrov I.D.

Bauman Moscow State Technical University, Moscow, Russia

alinaflower@rambler.ru

pivanetrov@gmail.ru

Abstract: This article is devoted to the research of archaeological excavation materials and to the developing of the structure of the initially obtained data, which is one of the problems of the digital history. An approach is proposed for the development of the digital archaeological data processing systems, which adequately measure up the criteria for integrity, completeness and reliability of information, as well as representativity of the visualization, taking into account the peculiar properties of archaeological materials. The spectrum of various data obtained in the course and results of archaeological excavations is studied. The results of the work are applicable as a basis for the large-scale development of digital archaeological data processing systems.

⁵Минитаева Алина Мажитовна – к.т.н., доцент кафедры "Компьютерные системы и сети" МГТУ им. Н.Э. Баумана, SPIN-код 8014-8367.

⁶Петров Иван Дмитриевич – бакалавр кафедры "Компьютерные системы и сети" МГТУ им. Н.Э. Баумана.

Keywords: archaeological excavations, artefacts, history and computing, digital history, structure, initially obtained data

Введение

В прикладном междисциплинарном направлении исторической информатики, методы и способы использующиеся для повышения наглядности исторических материалов и для обработки данных археологических раскопок, выделяют следующие направления разработок, приведенные на рис. 1.

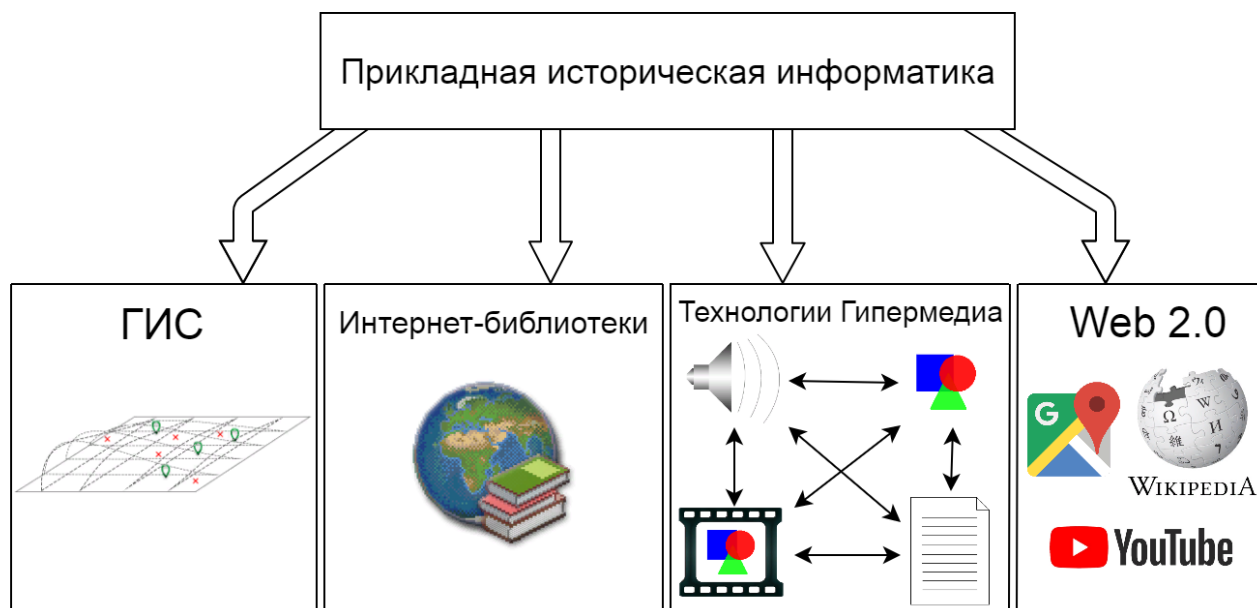


Рис. 1. Направления прикладной исторической информатики

Наиболее соответствующим решением задачи обработки археологических материалов является объединение методик создания географических информационных систем (ГИС) и крупных интернет-ресурсов, представляющих собой библиотеки данных, доступных большому кругу пользователей. Данные направления были выбраны ввиду того, что необходимо явно визуализировать область археологических раскопок, а также ее элементы и составляющие, и обеспечить масштабный охват аудитории. В свою очередь, технологии гипермедиа и Web 2.0, при всех их достоинствах, таких как высокая визуальная репрезентативность и оригинальность решений в первом случае и возможность участия в добавлении и редактировании данных обычных пользователей, не отвечают в должной мере критериям полноты и достоверности информации, особенно учитывая особенности археологических материалов.

В итоге, выбрав использование методов ГИС и крупных интернет-ресурсов, возникает проблема решения задачи структурирования исходных данных археологических раскопок. Создание определенной структуры исходных данных требуется как в случае работы с геоинформационными системами, так и с интернет-библиотеками. Решение задачи структурирования археологических материалов для

удобства их визуализации и представления в информационных системах является целью данной работы.

1. Формализация археологических материалов при помощи чертежей

Раскопки — это метод определения слоев, остатков сооружений и рельефов, которые скрыты под поздними слоями [2]. При помощи раскопок добывается разнообразная информация, которая тщательно записывается в определенном порядке, чтобы понять положение объектов, которые обычно представлены в виде археологических памятников, а также культурных и прочих слоев, или другими, словами, пластов почвы или породы.

Культурным слоем обычно принято называть место, на котором в течение какого-то времени жили люди и надолго сохраняющее следы и остатки человеческой жизнедеятельности [2]. На стенках любого раскопа будут видны несколько слоев характеристики которых представлены в порядке углубления в табл. 1.

Таблица 1.

Слои, которые открываются в ходе археологических работ

Номер слоя	Название	Образование	Следы жизнедеятельности	Цветовое различие
1	Современная почва	После пребывания человека	Отсутствуют, так как не происходило существенных воздействий	Определенный цвет
2	Культурный слой	Во время пребывания человека	Имеет следы жизнедеятельности, такими как:	Определенный цвет или несколько цветов
			Керамика	
			Уголь	
			Участки жилищ	
			Прочие предметы или их части	
3	Погребенная почва	До пребывания человека	Отсутствуют, так как не происходило существенных воздействий	Определенный цвет
4	Подстилающая порода	-	-	Определенный цвет

Археологическими памятниками (или артефактами) называются любые остатки древних сооружений, следы трудовой или культовой деятельности древних людей, а также сами древние вещи, которые были изготовлены человеком или приспособленные им для своих целей [4]. Как раз такие памятники и находятся в культурных слоях.

На раскопках главной задачей является формализация полученной в ходе наблюдений информации, которая извлекается в ходе полевых работ. Большая часть информации, получаемая на раскопках, формализуется в виде разнообразных археологических чертежей, использующиеся в качестве исходных данных для

информационной системы. Для полноценного отражения всех типов работ составляются следующие виды чертежей: ситуационные планы местности; топографические планы памятников; планы места проведения исследований; планы пластов; стратиграфические профили; сечения.

Необходимо использовать все типы археологических чертежей и данные о том в каком культурном слое был найден тот или иной археологический памятник.

2. Документирование археологических памятников

Для решения задачи структурирования исходных данных необходимо выбирать такие характеристики артефактов, которые обладают достаточно высоким уровнем объективности и непротиворечивости. Первичное документирование наиболее репрезентативно с точки зрения объективности информации, так как сразу после обнаружения находки невозможно дать ее подробное описание, поэтому используются характеристики, которые полноценно идентифицируют находку. Также подобным свойством обладает лабораторный учет предметов. В табл. 2 приведена классификация находок и соответствующий им учет.

Таблица 2.

Классификация находок и их первичное и лабораторное документирование

Находки	Учет	Первоначальный учет		Лабораторный учет
Индивидуальные	Отдельный учет	Запись находки	Название	Опись предмета
			+	Порядковый номер
Массовые	Каждая находка включается в комплекс	Запись комплекса находок	Место работы	Полевой номер
			Год раскопок	Материал
			Номер раскопа	Размеры
			Пласт культурного слоя	Описание
			Глубина	Координаты
			Квадрат	Рисунок
			Сооружение или погребение	Примечание
			Номер	

Главное же отличие массовых находок от индивидуальных заключается в том, что индивидуальные археологические памятники иллюстрируются, фотографируются или сканируются. С помощью этих методов получают двухмерное или трехмерное изображение, которое потом можно использовать в цифровых структурах данных. То есть совершается первичный процесс информатизации. А при условии объединения остальных данных каждой находки и затем всех находок в комплекс получается полная картина археологических раскопок.

3. Формирование структуры исходных данных археологических раскопок

Структура исходных данных должна включать те материалы археологических раскопок, которые относятся к составлению информационной базы для описания и исследования культурных слоев и выведенной из них хронологии жизнедеятельности человека на участке раскопок. Такие сведения содержатся в археологических чертежах и описях, которые составлены по результатам полевых работ.

На основе проведенных исследований данных при помощи нотации Джексона, которые относятся к материалам раскопок, составлена схема структуры пакета археологических материалов, то есть исходных данных, имеющая следующий вид, приведенный на рис. 2.



Рис. 2. Структура исходных археологических данных (диаграмма Джексона)

Конструкции повторения, последовательности и выбора между элементами, показанная на диаграмме Джексона соответствуют соотношениям характеристик объектов в структуре исходных данных археологических материалов.

Заключение

В результате проведенного исследования преобразования получена структура на основе характеристик археологических памятников, которая позволяет выбрать конкретный способ организации данных в информационных системах, обрабатывающие материалы археологических раскопок. Разработка подобных систем является актуальной

задачей, предусматривающая цель усовершенствования работы археологов на основе информационных технологий и облегчения доступа пользователям к оцифрованным историческим ресурсам, повысив степень их наглядности и визуальной репрезентации, а также развития прикладной исторической информатики в целом.

Список литературы

1. Иванова Г.С. Технология программирования: учебник для вузов. М.: КНОРУС. 2016. 334 с.
2. Мартынов А.И., Шер Я.А. Методы археологического исследования. М.: Высш. шк. 2002. 240 с.
3. Шакиров З.Г. Методы фиксации в археологии. Казань: Казан. Ун-т. 2015. 114 с.
4. Мартынов А.И. Археология. М.: Высш. шк. 2005. 447 с.
5. Иванова Г.С., Ничушкина Т.Н. Проектирование программного обеспечения: Учебное пособие. М.: Изд-во МГТУ им. Н.Э. Баумана. 2013. 104 с.
6. Бородкин Л.И. Digital History: Применение цифровых медиа в сохранении историко-культурного наследия? // Историческая информатика. 2012. №1. С.14-21.

Особенности преобразования тригонометрических спектров в базис обобщенных функций Хартли

⁷Гуренко В.В., ⁸Бычков Б.И.

МГТУ им. Н.Э. Баумана (НИУ), Москва, Россия

wgurenko@bmstu.ru

b102009@yandex.ru

Аннотация: Работа посвящена задаче преобразования спектра дискретного сигнала из классического базиса Фурье в обобщенный базис Хартли в системе счисления с произвольным основанием. Рассмотрены особенности анализа спектров в базисных системах Фурье и Хартли. Определена структура и сформулированы основные свойства матричного ядра преобразования тригонометрических спектров и спектров Хартли. Получены аналитические правила выделения независимых групп спектральных коэффициентов с целью снижения вычислительной сложности спектральных преобразований. Сформулированы алгоритмы преобразования гармонического спектра Фурье в обобщенный спектр Хартли. Выполнена аналитическая оценка вычислительной сложности алгоритмов. Результаты работы могут быть востребованы при решении широкого класса задач, связанных с имитацией и цифровой обработкой сигналов.

Работа выполнена в рамках проекта 2.7782.2017/БЧ. осуществляемого при поддержке Минобрнауки России

Ключевые слова: базисные системы Фурье и Хартли, анализ спектра, преобразование спектра, ядро Фурье, система счисления, вычислительная сложность.

Trigonometric spectrum to generalized Hartley basis transformation specifics

Gurenko V.V., Bychkov B.I.

Bauman Moscow State Technical University, Moscow, Russia

wgurenko@bmstu.ru

b102009@yandex.ru

Abstract: This article is devoted to a problem of converting a discrete signal spectrum from the classical Fourier basis to the generalized Hartley basis in the number system with an arbitrary base. Features of the spectrum analysis in Fourier and Hartley basis systems are considered. The structure and main properties of the transformation matrix core for trigonometric and Hartley spectrums are determined. Analytical rules for separation spectral

⁷Гуренко Владимир Викторович – доцент кафедры "Компьютерные системы и сети" МГТУ им. Н.Э. Баумана, SPIN-код 2675-4796.

⁸Бычков Борис Игоревич – магистрант кафедры "Компьютерные системы и сети" МГТУ им. Н.Э. Баумана, SPIN-код 6970-0210.

coefficients to independent groups are obtained with the aim of reducing spectrum transformation computational complexity. The algorithms for transforming harmonic Fourier spectrums into generalized Hartley spectrums are formulated. Analytical computational complexity estimation for the algorithms is performed. The results of the work can be used for solving a wide class of problems related to simulation and digital signal processing.

Keywords: Fourier and Hartley basic systems, spectrum analysis, spectrum transformation, Fourier kernel, number system, computational complexity.

Введение

В процессе разработки моделей систем реального времени широкого назначения возникают задачи, связанные со спектральным подходом к имитации и обработке сигналов различной формы и длительности. Для анализа спектров и воспроизведения сигналов применяются классические методы дискретного преобразования Фурье (ДПФ) в различных базисных системах [1–5], в том числе тригонометрической. При этом возникает необходимость взаимного преобразования спектров из одной базисной системы в другую с целью снижения вычислительной сложности алгоритмов имитации. Особый практический интерес при этом представляют обобщенные параметрические базисные системы и, в частности, система обобщенных функций Хартли (ОФХ).

Целью настоящей работы является исследование особенностей взаимного преобразования спектров сигналов из тригонометрических систем Фурье в системы ОФХ. Из трех основных систем упорядочения функций в обобщенных базисах – Адамара, Пэли и Хармута – в работе использовано наиболее употребимое упорядочение Адамара [6].

1. Тригонометрические системы Фурье и обобщенные системы Хартли

Тригонометрические системы Фурье являются составными базисными системами и образуются из тригонометрических четных $\cos\left(\frac{2\pi}{N} \nu i\right)$ и нечетных $\sin\left(\frac{2\pi}{N} \nu i\right)$ дискретных функций, расположенных в порядке возрастания их номера ν , где $\nu = 0, 1, \dots, N/2$ для четных значений N и $\nu = 0, 1, \dots, (N-1)/2$ для нечетных N [4]. Данные функции определены на дискретном интервале $[0, N)$ из N точек. Обратное и прямое ДПФ в данных базисах имеют вид:

– для четного N

$$x(i) = X_{\text{ч}}(0) + X_{\text{ч}}\left(\frac{N}{2}\right)\cos(\pi i) + \sum_{\nu=1}^{\frac{N}{2}-1} \left[X_{\text{ч}}(\nu)\cos\left(\frac{2\pi}{N} \nu i\right) + X_{\text{н}}(\nu)\sin\left(\frac{2\pi}{N} \nu i\right) \right], \quad (1)$$

$$\left. \begin{aligned} X_{\text{ч}}(0) &= \frac{1}{N} \sum_{i=0}^{N-1} x(i), X_{\text{ч}}\left(\frac{N}{2}\right) = \frac{1}{N} \sum_{i=0}^{N-1} x(i) \cos(\pi i), \\ X_{\text{ч}}(v) &= \frac{2}{N} \sum_{i=0}^{N-1} x(i) \cos\left(\frac{2\pi}{N} vi\right), X_{\text{н}}(v) = \frac{2}{N} \sum_{i=0}^{N-1} x(i) \sin\left(\frac{2\pi}{N} vi\right), \\ v &= 1, 2, \dots, \frac{N}{2} - 1 \end{aligned} \right\}; \quad (2)$$

– для нечетного N

$$x(i) = X_{\text{ч}}(0) + \sum_{v=1}^{\frac{N-1}{2}} \left[X_{\text{ч}}(v) \cos\left(\frac{2\pi}{N} vi\right) + X_{\text{н}}(v) \sin\left(\frac{2\pi}{N} vi\right) \right], \quad (3)$$

$$\left. \begin{aligned} X_{\text{ч}}(0) &= \frac{1}{N} \sum_{i=0}^{N-1} x(i), \\ X_{\text{ч}}(v) &= \frac{2}{N} \sum_{i=0}^{N-1} x(i) \cos\left(\frac{2\pi}{N} vi\right), X_{\text{н}}(v) = \frac{2}{N} \sum_{i=0}^{N-1} x(i) \sin\left(\frac{2\pi}{N} vi\right), \\ v &= 1, 2, \dots, \frac{N-1}{2} \end{aligned} \right\}. \quad (4)$$

Общий тригонометрический спектр $\{X_{\text{T}}(m)\}$ дискретного сигнала при этом представляется следующим образом [3]:

– для четного N

$$\{X_{\text{T}}(m)\} = \left\{ X_{\text{ч}}(0), X_{\text{н}}(1), X_{\text{ч}}(1), \dots, X_{\text{н}}\left(\frac{N}{2} - 1\right), X_{\text{ч}}\left(\frac{N}{2} - 1\right), X_{\text{ч}}\left(\frac{N}{2}\right) \right\}, \quad (5)$$

– для нечетного N

$$\{X_{\text{T}}(m)\} = \left\{ X_{\text{ч}}(0), X_{\text{н}}(1), X_{\text{ч}}(1), \dots, X_{\text{н}}\left(\frac{N-1}{2}\right), X_{\text{ч}}\left(\frac{N-1}{2}\right) \right\}. \quad (6)$$

Дискретные обобщенные функции Хартли (ОФХ) $Cas(k, i)$ образуются путем сложения обобщенных функций косинуса и синуса [6, 7]:

$$Cas(k, i) = \cos\left(\frac{2\pi}{p} \sum_{\lambda=1}^n k_{\lambda} i_{\lambda}\right) + \sin\left(\frac{2\pi}{p} \sum_{\lambda=1}^n k_{\lambda} i_{\lambda}\right), \quad (7)$$

где k_{λ} и i_{λ} являются λ -ми разрядами n -разрядного p -ичного кода чисел k и i соответственно и лежат в целочисленном диапазоне $[0, p-1]$. Объединение

$$N = p^n, n = 1, 2, \dots, \text{K} \quad (8)$$

первых ОФХ приводит к полной ортонормированной базисной системе.

Базисная система (7) является обобщенной системой Хартли–Адамара. Обобщенное прямое преобразование Хартли (ОПХ) имеет вид:

$$X_{\text{X}}(k) = \frac{1}{N} \sum_{i=0}^{N-1} x(i) Cas(k, i). \quad (9)$$

2. Обобщенный анализ спектра в базисах Фурье и Хартли

Задача обобщенного спектрального анализа состоит в определении спектра $\{X_X(k)\}$ в базисе ОФХ по известному тригонометрическому спектру $\{X_q(v), X_H(v)\}$ в базисе тригонометрических функций Фурье (ТФФ). Ее решение в зависимости от четности или нечетности N сводится к совместному решению уравнений (9) и (1)–(4), которое в результате приводит к следующим уравнениям:

– для четного N

$$X_X(k) = \sum_{v=0}^{\frac{N}{2}} X_q(v) \Phi_q(k, v) + \sum_{v=1}^{\frac{N-1}{2}} X_H(v) \Phi_H(k, v), \quad k = 0, 1, \dots, N-1; \quad (10)$$

– для нечетного N

$$X_X(k) = \sum_{v=0}^{\frac{N-1}{2}} X_q(v) \Phi_q(k, v) + \sum_{v=1}^{\frac{N-1}{2}} X_H(v) \Phi_H(k, v), \quad k = 0, 1, \dots, N-1. \quad (11)$$

Эти уравнения определяют линейный оператор преобразования спектров Фурье и Хартли, математическую основу которого составляет двумерная функция Φ , которая по своим четной Φ_q и нечетной Φ_H составляющим записывается в виде:

$$\Phi_q(k, v) = \frac{1}{N} \sum_{i=0}^{N-1} \cos\left(\frac{2\pi}{N} vi\right) \text{Cas}(k, i), \quad (12)$$

$$\Phi_H(k, v) = \frac{1}{N} \sum_{i=0}^{N-1} \sin\left(\frac{2\pi}{N} vi\right) \text{Cas}(k, i). \quad (13)$$

Функция Φ , представляемая зависимостями (12) и (13), носит название ядра Фурье [3,8]. На практике наиболее удобно пользоваться матричным представлением ядра. В силу сложности ТФФ и ОФХ получение аналитических выражений для элементов ядра Фурье при общих значениях параметров N , p и n затруднительно. Поэтому используется численный метод анализа ядра, основанный на расчетах матриц Φ различных порядков с определением их структурных особенностей.

3. Уравнения ядра преобразования спектров

Элементы ядра Фурье в базисах ТФФ и ОФХ–Адамара могут быть вычислены по формулам (12–13) и зависят от четности или нечетности N . Согласно (8), четность N однозначно определяется четностью p , поэтому структуру ядра можно рассматривать для четных и нечетных значений p . Проведенный анализ показывает, что в общем случае для нечетных p матрица Φ будет содержать один элемент, равный 1, $2(p-1)$ элементов, равных ± 0.5 , $(2N^2 + p - 1)/(p + 1)$ иных ненулевых элементов и $(N^2 - 1)(p - 1)/(p + 1)$ элементов, равных нулю. Из ненулевых элементов и

соответствующих им спектральных коэффициентов можно образовать $n(p-1)/2+1$ независимых групп согласно правилу, сформулированному в табл. 1.

Таблица 1.

Правило образования групп для нечетного p

Номер группы	Число элементов в группе	Обобщенные коэффициенты Хартли–Адамара	Тригонометрические коэффициенты Фурье
0	1	$X_A(0)$	$X_q(0)$
$k, k=1,2,\dots,\frac{p-1}{2}$	2	$X_A(k), X_A(p-k)$	$X_H(kp^{n-1}), X_q(kp^{n-1})$
$\frac{p-1}{2} + (m-1)(n-1) + \lambda,$ $m=1,2,\dots,\frac{p-1}{2},$ $\lambda=1,2,\dots,n-1$	$2p^\lambda$	$X_A(mp^\lambda + j),$ $X_A[(p-m)p^\lambda + j],$ $j=0,1,\dots,p^\lambda-1$	$X_H\left[\frac{N}{p^{\lambda+1}}(m+pj)\right], X_q\left[\frac{N}{p^{\lambda+1}}(m+pj)\right],$ $j=0,1,\dots,\frac{p^\lambda-1}{2}$ $X_H\left[\frac{N}{p^{\lambda+1}}(p-m+pj)\right], X_q\left[\frac{N}{p^{\lambda+1}}(p-m+pj)\right],$ $j=0,1,\dots,\frac{p^\lambda-3}{2}$

С учетом данного правила аналитическое описание оператора преобразования тригонометрического спектра Фурье в обобщенный спектр Хартли–Адамара для нечетного $p \geq 3$ записывается следующим образом:

$$X_A(0) = X_q(0), \quad (14)$$

$$\left. \begin{aligned} X_A(k) &= X_H(kp^{n-1})\Phi_H(k, kp^{n-1}) + X_q(kp^{n-1})\Phi_q(k, kp^{n-1}), \\ X_A(p-k) &= X_H(kp^{n-1})\Phi_H(p-k, kp^{n-1}) + X_q(kp^{n-1})\Phi_q(p-k, kp^{n-1}), \\ k &= 1, 2, \dots, \frac{p-1}{2}; \end{aligned} \right\} \quad (15)$$

$$\begin{aligned} X_A(mp^\lambda + j) &= \sum_{i=0}^{\frac{p^\lambda-1}{2}} \left\{ X_H\left[\frac{N}{p^{\lambda+1}}(m+pi)\right] \Phi_H\left[mp^\lambda + j, \frac{N}{p^{\lambda+1}}(m+pi)\right] + \right. \\ &+ X_q\left[\frac{N}{p^{\lambda+1}}(m+pi)\right] \Phi_q\left[mp^\lambda + j, \frac{N}{p^{\lambda+1}}(m+pi)\right] \left. \right\} + \\ &+ \sum_{i=0}^{\frac{p^\lambda-3}{2}} \left\{ X_H\left[\frac{N}{p^{\lambda+1}}(p-m+pi)\right] \Phi_H\left[mp^\lambda + j, \frac{N}{p^{\lambda+1}}(p-m+pi)\right] + \right. \\ &+ X_q\left[\frac{N}{p^{\lambda+1}}(p-m+pi)\right] \Phi_q\left[mp^\lambda + j, \frac{N}{p^{\lambda+1}}(p-m+pi)\right] \left. \right\}, \end{aligned} \quad (16)$$

$$\begin{aligned}
X_A((p-m)p^\lambda + j) &= \sum_{i=0}^{\frac{p^\lambda-1}{2}} \left\{ X_H \left[\frac{N}{p^{\lambda+1}}(m+pi) \right] \Phi_H \left[(p-m)p^\lambda + j, \frac{N}{p^{\lambda+1}}(m+pi) \right] + \right. \\
&+ X_q \left[\frac{N}{p^{\lambda+1}}(m+pi) \right] \Phi_q \left[(p-m)p^\lambda + j, \frac{N}{p^{\lambda+1}}(m+pi) \right] \left. \right\} + \\
&+ \sum_{i=0}^{\frac{p^\lambda-3}{2}} \left\{ X_H \left[\frac{N}{p^{\lambda+1}}(p-m+pi) \right] \Phi_H \left[(p-m)p^\lambda + j, \frac{N}{p^{\lambda+1}}(p-m+pi) \right] + \right. \\
&+ X_q \left[\frac{N}{p^{\lambda+1}}(p-m+pi) \right] \Phi_q \left[(p-m)p^\lambda + j, \frac{N}{p^{\lambda+1}}(p-m+pi) \right] \left. \right\}, \tag{17}
\end{aligned}$$

$$m = 1, 2, \dots, \frac{p-1}{2}; \lambda = 1, 2, \dots, n-1; j = 0, 1, \dots, p^\lambda - 1.$$

Такие же рассуждения позволяют получить аналогичные уравнения для четных значений p :

$$X_A(0) = X_q(0), X_A\left(\frac{p}{2}\right) = X_q\left(\frac{N}{2}\right); \tag{18}$$

$$\left. \begin{aligned}
X_A(k) &= X_H(kp^{n-1})\Phi_H(k, kp^{n-1}) + X_q(kp^{n-1})\Phi_q(k, kp^{n-1}), \\
X_A(p-k) &= X_H(kp^{n-1})\Phi_H(p-k, kp^{n-1}) + X_q(kp^{n-1})\Phi_q(p-k, kp^{n-1}), \\
k &= 1, 2, \dots, \frac{p}{2} - 1;
\end{aligned} \right\}, \tag{19}$$

$$\begin{aligned}
X_A(mp^\lambda + j) &= \sum_{i=0}^{\frac{p^\lambda}{2}-1} \left\{ \begin{aligned} &X_H \left[\frac{N}{p^{\lambda+1}}(m+pi) \right] \Phi_H \left[mp^\lambda + j, \frac{N}{p^{\lambda+1}}(m+pi) \right] + \\ &+ X_{\text{ч}} \left[\frac{N}{p^{\lambda+1}}(m+pi) \right] \Phi_{\text{ч}} \left[mp^\lambda + j, \frac{N}{p^{\lambda+1}}(m+pi) \right] + \\ &+ X_H \left[\frac{N}{p^{\lambda+1}}(p-m+pi) \right] \Phi_H \left[mp^\lambda + j, \frac{N}{p^{\lambda+1}}(p-m+pi) \right] + \\ &+ X_{\text{ч}} \left[\frac{N}{p^{\lambda+1}}(p-m+pi) \right] \Phi_{\text{ч}} \left[mp^\lambda + j, \frac{N}{p^{\lambda+1}}(p-m+pi) \right] \end{aligned} \right\}; \\
X_A((p-m)p^\lambda + j) &= \sum_{i=0}^{\frac{p^\lambda}{2}-1} \left\{ \begin{aligned} &X_H \left[\frac{N}{p^{\lambda+1}}(m+pi) \right] \Phi_H \left[(p-m)p^\lambda + j, \frac{N}{p^{\lambda+1}}(m+pi) \right] + \\ &+ X_{\text{ч}} \left[\frac{N}{p^{\lambda+1}}(m+pi) \right] \Phi_{\text{ч}} \left[(p-m)p^\lambda + j, \frac{N}{p^{\lambda+1}}(m+pi) \right] + \\ &+ X_H \left[\frac{N}{p^{\lambda+1}}(p-m+pi) \right] \Phi_H \left[(p-m)p^\lambda + j, \frac{N}{p^{\lambda+1}}(p-m+pi) \right] + \\ &+ X_{\text{ч}} \left[\frac{N}{p^{\lambda+1}}(p-m+pi) \right] \Phi_{\text{ч}} \left[(p-m)p^\lambda + j, \frac{N}{p^{\lambda+1}}(p-m+pi) \right] \end{aligned} \right\}; \tag{20}
\end{aligned}$$

$$m = 1, 2, \dots, \frac{p}{2} - 1; \lambda = 1, 2, \dots, n-1; j = 0, 1, \dots, p^\lambda - 1;$$

$$X_A\left(\frac{p^{\lambda+1}}{2} + j\right) = \sum_{i=0}^{\frac{p^\lambda}{2}-1} \left\{ \begin{aligned} &X_H \left[\frac{N}{p^{\lambda+1}}\left(\frac{p}{2} + pi\right) \right] \Phi_H \left[\frac{p^{\lambda+1}}{2} + j, \frac{N}{p^{\lambda+1}}\left(\frac{p}{2} + pi\right) \right] + \\ &+ X_{\text{ч}} \left[\frac{N}{p^{\lambda+1}}\left(\frac{p}{2} + pi\right) \right] \Phi_{\text{ч}} \left[\frac{p^{\lambda+1}}{2} + j, \frac{N}{p^{\lambda+1}}\left(\frac{p}{2} + pi\right) \right] \end{aligned} \right\};$$

$$\lambda = 1, 2, \dots, n-1; j = 0, 1, \dots, p^\lambda - 1.$$

4. Оценка вычислительной сложности ядра Фурье

О временной вычислительной сложности ядра проведем для нечетных значений p . Нулевая группа в табл. 1 состоит из одного элемента, равного 1. Вычисление спектральных коэффициентов второй группы потребует выполнения двух операций умножения и двух операций сложения. Принимая равными длительности умножения и сложения, получим общее число операций для этих групп:

$$O_1 = 2(p-1).$$

Для вычисления коэффициентов последующих групп потребуется следующее количество операций:

$$O_2 = 8 \sum_{\lambda=1}^{n-1} p^{2\lambda} \left(\frac{p-1}{2} \right) = \frac{4(p^{2n} - p^2)}{p+1} = \frac{4(N^2 - p^2)}{p+1}.$$

Общее число вычислительных операций образуется суммой значений O_1 и O_2 , которая после эквивалентных преобразований запишется следующей формулой:

$$O = \frac{2(2N^2 - p^2 - 1)}{p + 1}. \quad (21)$$

Из выражения (21) следует, что по сравнению с прямым методом преобразования спектра по формулам (10) – (11), где число вычислительных операций равно $4N^2$, использование предложенного способа позволит более чем в $(p + 1)$ раз уменьшить объем необходимых операций. С увеличением p эффективность предложенного способа линейно возрастает.

Заключение

В результате исследования особенностей преобразования тригонометрического спектра Фурье в обобщенный спектр Хартли в упорядочении Адамара сформулированы основные свойства ядра Фурье, состоящие в образовании определенного числа независимых групп элементов. Благодаря этому расчет спектральных коэффициентов и обобщенный анализ спектров представлен в виде системы уравнений, более простых для программной реализации. Выполненная оценка вычислительной сложности ядра преобразований спектров показывает сокращение общего числа вычислительных операций, необходимых для расчета спектра сигнала. С увеличением основания системы счисления эффективность предложенного способа возрастает. Результаты работы могут быть использованы при решении задач имитации детерминированных и псевдослучайных сигналов и в ряде других областей цифровой обработки сигналов [9].

Список литературы

1. Oppenheim A., Schaffer R. Digital Signal Processing. Prentice-Hall, Englewood Cliffs. 2012. 1120 p.
2. Ifeachor E., Jervis B. Digital Signal Processing. A Practical Approach. Pearson Education. 2002. 992 p.
3. Сюезв В.В. Основы теории цифровой обработки сигналов. М.: Издательство «РТСофт». 2014. 752 с.
4. Golovin Yu.M., Zavelevich F.S., Nikulin A.G., Kozlov D.A., Monakhov D.O., Kozlov I.A., Arkhipov S.A., Tselikov V.A., Romanovsky A.S. Spaceborne Infrared Fourier-Transform Spectrometers for Temperature and Humidity Sounding of the Earth's Atmosphere // *Izvestiya, Atmospheric and Oceanic Physics*. 2014. Vol. 50. Is. 9. P. 1004–1015.
5. Шахтарин Б.И., Сидоркина Ю.А., Никифоров А.А. Оценка фазы псевдослучайной последовательности в системах передачи информации с расширенным спектром // *Вестник МГТУ им. Н.Э. Баумана. Сер. Приборостроение*. 2015. №3. С. 92–103.

6. Сюзев В.В. Обобщенные функции и преобразования Хартли в системах счисления с постоянным основанием // Вестник МГТУ им. Н.Э. Баумана. Сер. Приборостроение. 2014. №2. С. 63–79.
7. Bracewell R. The Hartley Transform. Oxford University Press. 1990. 175 p.
8. Трахтман А.М., Трахтман В.А. Основы теории дискретных сигналов на конечных интервалах. М.: Сов. радио. 1975. 208 с.
9. Сюзев В.В., Гуренко В.В. Гармонические алгоритмы имитации сигналов в рамках корреляционной теории // Вестник МГТУ им. Н.Э. Баумана. Сер. Приборостроение. 2017. №4. С. 98–117. DOI: 10.18698/0236-3933-2017-4-98-117

Анализ возможности применения технологии дополненной реальности для навигации в торговом центре

⁹Ничушкина Т.Н., ¹⁰Круглов Д.С.

МГТУ им. Н.Э. Баумана (НИУ), Москва, Россия

agitatore@yandex.ru

tnich@rambler.ru

Аннотация: В статье выполнен анализ перспективной технологии объединения изображений магазина в торговом центре с подробным описанием этого магазина, загруженным из базы данных, с целью создания системы дополненной реальности для ориентирования в торговом центре. Описана архитектура системы. Рассмотрены функциональные возможности и взаимодействие с системой мобильного приложения для пользователей, и веб-приложения для администраторов магазинов. Разъясняются технологии, позволяющие реализовать данную систему. Описаны технология и процесс создания двухуровневой кластеризации – структуры данных, для выполнения быстрого поиска по изображению.

Ключевые слова: дополненная реальность, распознавание изображений, метаданные изображений, кластер, поиск маршрута.

The analysis of the possibility of applying augmented reality technology for navigation in a mall

Nichushkina T.N., Kruglov D.S.

Bauman Moscow State Technical University, Moscow, Russia

agitatore@yandex.ru

tnich@rambler.ru

Abstract: The article analyzes the perspective technology of combining store images in a mall with a detailed description of this store, downloaded from the database, in order to create augmented reality system for shopping center orientation. The architecture of the system is described. Considered are the functionality and interaction with the mobile application system for users, and web applications for store administrators. Clarified technologies that make it possible to implement this system. The technology and the process of creating a two-level clustering - a data structure, for performing a quick search on an image are described.

Keywords: augmented reality, image recognition, image metadata, cluster, route search.

⁹**Ничушкина Татьяна Николаевна** – канд. техн. наук, доцент кафедры «Компьютерные системы и сети» МГТУ им. Н.Э. Баумана. Автор свыше 30 научных и методических работ в области вычислительной техники. Специализируется в области проектирования программных систем.

¹⁰**Круглов Дмитрий Сергеевич** — студент магистратуры кафедры «Компьютерные системы и сети» МГТУ им. Н.Э. Баумана.

Введение

Большинство приложений с технологией дополненной реальности (англ. Augmented Reality, AR) от различных магазинов сосредоточено на предоставлении клиентам возможности визуализировать и взаимодействовать с продуктами через интернет-магазины [1] или отображать некоторые сведения о продуктах, когда клиенты считывают штрих код с помощью смартфона.

Когда покупатель впервые посещает большой торговый центр, ему придется потратить достаточно много времени, чтобы узнать, что каждый магазин продает и где находится интересующий его товар. Таким покупателям могла бы помочь система, в которой покупателю нужно добавить товары в список покупок приложения, сделать снимок внутри торгового центра, на котором будут показаны детали каждого магазина и информация, связанная с его списком покупок. Аннотации могут включать имена магазинов, их ассортимент и отзывы других клиентов. Кроме того, система может предложить маршрут к интересующим магазинам, по пути демонстрируя фотографии и информацию о других магазинах, мимо которых клиент будет проходить.

Такая система упростит поиск магазинов торгового центра для новых клиентов с помощью использования средств дополненной реальности. Когда покупатель входит в магазин и делает снимок на телефон, дополнительная информация об окружающей среде появится на экране. Такая система может быть личным помощником, который имеет возможность показывать маршруты в магазины, которые ищет покупатель. Для управления своими магазинами и продвижения своего бизнеса администраторам нужно предоставить понятный веб-интерфейс. Для клиентов будет удобнее использовать приложение для смартфона. Когда клиенты делают снимок на телефон в торговом центре и запрашивают систему, соответствующая информация о магазинах и окружении будет отображаться на экране.

1. Архитектура и интерфейсы

Система должна иметь два пользовательских интерфейса, принимающих входные данные. Одним из них является веб-интерфейс для администраторов и продавцов для создания модели торгового центра путем загрузки изображений и соответствующих метаданных. Другой интерфейс является приложением для смартфона, которое используется клиентами для отправки запросов и отображения аннотаций о магазинах. Обработка запросов клиентов делится на три этапа: построение модели, сопоставление отличительных особенностей и генерация аннотаций. Первый шаг автономный, в то время как последние два шага выполняются в режиме онлайн. Во время создания модели администраторы должны делать фотографии магазинов и метаданные, используя приложение для смартфона, а затем отправлять их на сервер через веб-интерфейс. Когда клиент делает снимок с помощью приложения и отправляет на сервер, данные о

местоположении и визуальные отличительные особенности будут извлекаться из метаданных и изображений.

После нахождения соответствия отличительных особенностей из базы данных и изображения аннотации найденных магазинов будут отображены в изображении из запроса и отображены на экране пользователя. Высокоуровневое представление архитектуры системы показано на рисунке 1.

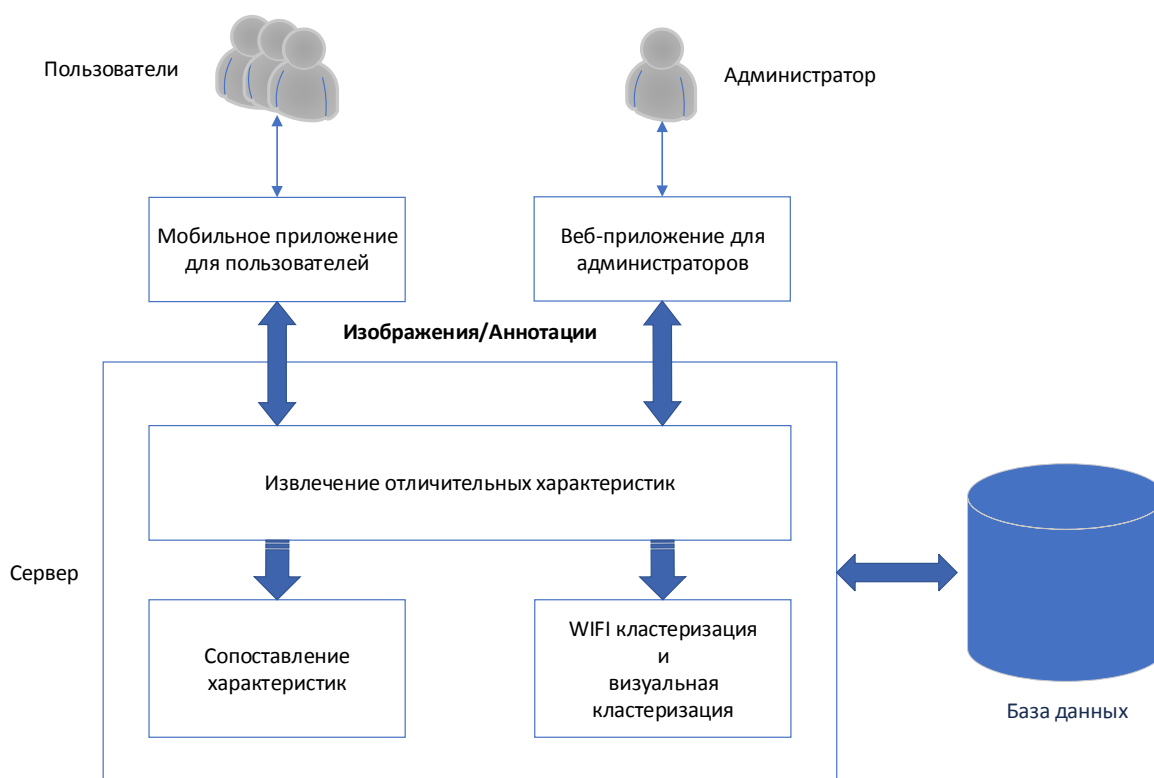


Рис. 1. Архитектура системы

1. Веб-приложение для администраторов

Построение модели торгового центра — это основной шаг, который необходимо выполнить администраторам магазинов или управляющим торговых центров. Сначала, им нужно делать фотографии с помощью приложения для смартфона, поскольку оно будет записывать метаданные, включая GPS и сигнал Wi-Fi. Фотографии должны быть сделаны под разными углами, чтобы повысить точность распознавания. Затем фотографии должны быть загружены на сервер вместе с аннотацией магазина.

2. Мобильное приложение для клиентов

Приложение для смартфонов дает возможность пользователям отправлять запросы с изображениями системе. Клиенты могут фотографировать магазины в торговом центре и просматривать возвращаемые аннотации посредством интерактивных взаимодействий, таких как перетаскивание и масштабирование, а также их описания или обзоры. Аннотации и отзывы, добавленные клиентами, также доступны для просмотра другим клиентам. Следовательно, улучшается обмен информацией и опытом покупок. В

дополнение к созданию изображений и аннотаций система также должна поддерживать такие функции, как:

- 1) просмотр изображений и аннотаций близлежащих магазинов (соседние магазины);
- 2) уведомления о ближайших магазинах, в которых продаются предметы из списка покупок клиентов (список покупок);
- 3) сопоставление списка покупок пользователя с аннотациями, созданными продавцами и получать сведения о магазинах, которые заинтересуют пользователя (предложение маршрута).

С помощью изображения из запроса и его метаданных можно определить местоположение пользователя с использованием предварительно сформированных пространственных кластеров.

2. Основные технологии

В этом разделе будет кратко рассказано о технических особенностях системы, включая двухуровневую кластеризацию, передачу аннотаций и предложение маршрута.

1. Двухуровневая кластеризация

Для оптимизации работы системы имеет смысл ввести двухуровневую кластеризацию [2], которая рассматривает как пространственные, так и визуальные характеристики (рисунок 2). Двухуровневая кластеризация нацелена на определение местоположения и ускоряет обработку запросов изображений и соответствие характеристик.

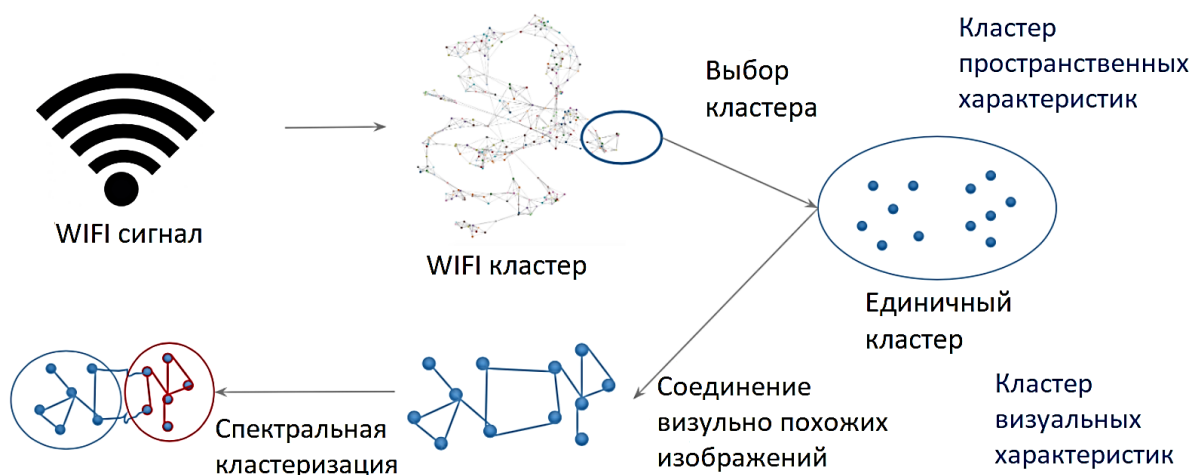


Рис. 2. Двухуровневая кластеризация

Кластеризация Wi-Fi или пространственной кластеризацией группирует все изображения на основе пространственных характеристик, извлеченных из метаданных. Необходимо использовать следующие характеристики: список доступных Wi-Fi сетей, BSSID (MAC-адрес), уровень сигнала и частоту [3]. Изображения объединяются в

разные группы на основе этих пространственных характеристик. Это первый уровень системы запросов изображений, который наполняется данными в автономном режиме, когда владельцы торговых центров создают модель торгового центра в приложении, загружая изображения. Граф кластера Wi-Fi администраторы могут просматривать и обрабатывать через веб-интерфейс. Например, администратор может выбирать область кластеров и просматривать изображения магазина внутри них. Он может создавать и изменять имена каждого кластера.

Вторым уровнем является визуальная кластеризация. Для обнаружения и извлечения отличительных особенностей из изображений обычно используется алгоритм преобразования объектов (SIFT) [6]. Отличительной особенностью алгоритма SIFT является то, что он более надежен при выполнении сопоставления изображений, взятых с разных углов обзора, что и требуется в данном случае. Метод k-ближайших [5] соседей используется для вычисления расстояния между каждой парой ключевых точек. Таким образом, создается матрица расстояний между изображениями, и к матрице применяется алгоритм иерархической спектральной кластеризации. Далее, изображения в каждом кластере Wi-Fi дополнительно группируются в соответствии с визуальными отличительными особенностями. Для просмотра результата кластеризации веб-интерфейс предоставляет специальные функции.

Когда клиенты отправляют изображение, пространственные характеристики извлекаются из метаданных, и начинается их поиск в кластере. Наиболее подходящий кластер изображений становится кандидатом на визуальное сопоставление отличительных особенностей. Таким образом, двухуровневая кластеризация делает сравнение и поиск изображений более эффективными.

2. Добавление аннотаций

Администраторам торговых центров необходима возможность для управления аннотациями. Когда администраторы должны сформировать аннотации для совершенно нового магазина, им следует обработать большое количество изображений магазина, снятых под разными углами. Обычно, это занимает много времени. Использование двухуровневой кластеризации, решает эту проблему. Сначала администратор должен выбрать прямоугольную область на самом характерном изображении, которое содержит магазин. Затем выполняется сопоставление методом k-ближайших соседей на изображениях в пределах того же визуального кластера, что и выбранное изображение. Это значительно экономит время и имеет смысл, потому что изображения, содержащие один магазин, должны быть похожи друг на друга и находиться в одном визуальном кластере. Двухуровневое сопоставление состоит из сопоставления отличительных особенностей методом SIFT, за которым следует сопоставление изображения методом k-ближайших соседей. Наиболее подходящие изображения будут возвращены и дополнительно отфильтрованы.

Окончательные результаты будут показаны администраторам как всплывающее окно в веб-интерфейсе. Кроме того, изображения магазинов можно просматривать, а область аннотации можно отрегулировать или удалить.

3. Предложение маршрута

Процедура предложения маршрута содержит два шага:

- 1) находится кратчайший путь между начальной точкой (местоположением запроса) и пунктом назначения (интересующий магазин) в графе кластера Wi-Fi;
- 2) после того, как найден кратчайший путь, выбираются и демонстрируются характерные фотографии из узлов графа в кратчайшем пути.

На первом шаге можно использовать алгоритм Дейкстры [5], чтобы найти кратчайший путь. На втором шаге предпосылка для выбора фотографии заключается в том, что на выбранных фотографиях должны быть изображены не одни и те же магазины, и выбранные фотографии не должны полностью отличаться, чтобы не показывать фотографии магазинов, которые слишком далеко друг от друга. Например, идеальным результатом является случай, когда выбираются изображения смежных магазинов. Изображения соседних магазинов имеют общие характерные черты, поэтому системе будет проще найти маршрут, а пользователю проще ориентироваться в торговом центре. Чтобы решить эту задачу, можно свести ее к задаче достижимости. Для этого нужно построить граф смежных изображений и найти в нем кратчайший маршрут от исходного изображения к целевому.

Заключение

В этой статье авторы постарались проанализировать потенциал создания программной системы с дополненной реальностью, которая позволяет упростить поиск магазинов торговом центре для посетителей.

Описанная система обеспечит необходимыми функциональными возможностями, как администраторов магазинов, так и покупателей. Она поможет новым клиентам находить товар, который они хотят купить, даст возможность делиться информацией с другими пользователями, например, отзывами о магазине или товарах. Новые решения, представленные авторами, облегчат управление моделью торгового центра и аннотациями, а также осуществлять запрос аннотаций по изображению в режиме реального времени. Кроме того, стоит отметить возможность определения текущего местоположения в торговом центре и предложение маршрута по изображениям.

Описанная система может быть использована для решения многих подобных задач. Например, художественные галереи и музеи обычно предоставляют ограниченную информацию по экспозициям. При использовании подобной системы, пользователям достаточно сделать снимок какого-то предмета, и подробная информация о нем будет отображена на экране его телефона. Этот аспект работы приложения мог бы заменить существующую систему аудиогuida в музеях и галереях.

Список литературы

1. Azuma R.T. A Survey of Augmented Reality // Teleoperators and Virtual Environments. 1997. Vol. 6. Is. 4. P. 355–385.
2. Овчинников В.А., Иванова Г.С., Методика формального синтеза комбинированных структур данных для представления графов. Вестник МГТУ им. Н. Э. Баумана. Сер. Приборостроение. 2012. Спец.вып. 4: Компьютерные системы и технологии. С. 135-145.
3. H.V. Jagadish, R.T. Ng, Beng Chin, Ooi Anthony, K.H. Tung. It compress: An iterative semantic compression algorithm. // In 20th International Conference on Data Engineering. IEEE. 2004. P. 646–657.
4. D.G. Lowe. Distinctive image features from scale-invariant keypoints. // International journal of computer vision. 2004. Vol. 60. Is. 2. P. 91–110.
5. Hall P., Park B.U., Samworth R.J. Choice of neighbor order in nearest-neighbor classification. // Annals of Statistics. 2008. Vol. 36. Is. 5. P. 2135–2152.
6. Овчинников В.А. Графы в задачах анализа и синтеза структур сложных систем. М.: Изд-во МГТУ им. Н. Э. Баумана. 2014. 423с.

Классификация критических ситуаций летательного аппарата на основе автоматического анализа данных

¹¹Попов А.Ю., ¹²Евдокимов А.А.

МГТУ им. Н.Э. Баумана (НИУ), Москва, Россия

Alexpopov@bmstu.ru

Evdokimov11.1996@gmail.com

Аннотация: Для предотвращения крушений летательных аппаратов предложен новый подход, основанный на алгоритмах машинного обучения и аппаратной платформы дискретной математики. Данная статья посвящена анализу данной платформы. Анализу метода «К Ближайших Соседей» на предмет его совместимости с данной вычислительной системой. Рассмотрены структура и особенности данного метода. Рассмотрены различные причины авиакатастроф, современные методы защиты и их недостатки, а также алгоритм обучения системы на предмет выявления возможного попадания летательного аппарата в критическую ситуацию.

Ключевые слова: Метод «К Ближайших Соседей», летательный аппарат, критическая ситуация, «LeonHard»

Classification of critical situations of an aircraft based on automatic data analysis

Popov A.Y., Evdokimov A.A.

Bauman Moscow State Technical University, Moscow, Russia

Alexpopov@bmstu.ru

Evdokimov11.1996@gmail.com

Annotation: A new approach based on machine learning algorithms and discrete mathematics hardware platform is proposed to prevent the crash of aircraft. This article is devoted to the analysis of this platform. Analysis of the method "K Nearest Neighbors" for its compatibility with this computer system. The structure and features of this method are considered. Various causes of plane crashes, modern methods of protection and their disadvantages, as well as the algorithm of training the system to identify the possible impact of the aircraft in a critical situation are considered.

Keywords: method "K nearest neighbors", aircraft, critical situation, «LeonHard»

¹¹Попов Алексей Юрьевич – к.т.н., доцент кафедры "Компьютерные системы и сети" МГТУ им. Н.Э. Баумана, SPIN-код: 7380-5537

¹²Евдокимов Александр Алексеевич – студент кафедры "Компьютерные системы и сети" МГТУ им. Н.Э. Баумана.

Введение

Ежедневно в мире совершается около 85,000 гражданских авиарейсов. В 2017 году в мире было совершено более 31 миллиона рейсов. Всего в 2017 году было перевезено 3 миллиарда пассажиров, т.е. почти половина населения земли. Речь идет только о гражданской авиации.

Но, согласно статистике [1], за последние 4 года в мире произошло 107 авиакатастроф со смертельным исходом, в которых погибло 3245 человек. Одной из причин авиакатастроф является попадание самолета в один из критических режимов полета. Попадание в такой режим не связано с какими-либо отказами техники. Во всех случаях налицо человеческий фактор и климатические условия. Так, при крушении ТУ-154 в Донецке, в котором погибло 170 человек, командир принял решение обойти грозовой фронт сверху. Подъем на высоту более 12000 метров привел к уменьшению скорости, увеличению угла атаки и, как следствию сваливанию самолета.

Таким образом, можно сделать вывод, что за много лет развития авиации, проблема безопасности летательных аппаратов не потела своей важности.

Для решения задач о предупреждении пилотов в последние годы активно используются методы машинного обучения, что значительно увеличило точность и качество решения указанных задач в сравнении со стандартными алгоритмами.

Целью данной работы является описание программного комплекса для классификации критических ситуаций летательного аппарата на основе автоматического анализа данных телеметрии и, как следствие, предупреждение летчика о ней. Программный комплекс производит классификацию состояния самолёта на основе параметров, поступающих с самолёта и выдаёт предупреждение о приближении к критической ситуации.

1. Классификация критических ситуаций

Во время проведения полёта может возникнуть ситуация, при которой летчику потребуется полная мобилизация сил, для предотвращения катастрофы. Такие опасные формы, в которые может попасть летательный аппарат, обычно называют критическими ситуациями. Как правило, данные режимы, являются последствием ошибки пилотирования, отказа в системе управления или ввиду погодных условий. Поэтому оценка границ области таких критических ситуаций и разработка способов их прекращения являются важным фактором при создании нового самолёта.

Для современных самолётов характерны следующие типичные критические формы движения [2]: сваливание, штопор, срыв, кренение.

Для критических форм движения самолёта характерно существенное влияние нелинейности аэродинамических характеристик, кинематических связей и т.д. Из-за этих нелинейных эффектов развитие критических форм движения зависит от исходных условий полёта, начальных значений параметров движения и т.п.

Сваливание самолёта - произвольное быстро развивающееся движение со значительной угловой скоростью или амплитудой, обусловленное потерей устойчивости на больших околокритических или критических углах атаки. Выход на околокритические углы атаки может быть обусловлен резким восходящим порывом ветра. К выходу же на критические углы может привести, например, отказ в системе управления и, как следствие, быстрое неконтролируемое отклонение руля с высоты

Шторм - движение самолёта по нисходящей спиралеобразной траектории малого радиуса на закритических углах атаки, развивающееся в результате авторотации крыла и дестабилизирующего инерционного момента. Важно отметить, что данный вид критического режима является одним из наиболее опасных в авиации.

Срыв – форма движения, которая происходит при достаточно больших критических угловых скоростях, которые ведут к потере устойчивости самолёта и развитию движения с резким изменением углов атаки и нарастанию угловой скорости. Даная критическая ситуация происходит самопроизвольно при интенсивных маневрах по крену на малых и отрицательных углах атаки и продолжается как после установки органов управления в нейтральное положение, так и при отклонении их в обратное положение.

Ряд критических форм движения может быть связан с потерей устойчивости или управляемости самолёта из-за упругих деформаций конструкций. Наиболее характерной формой такого критического движения является неуправляемое движение крена при реверсе элеронов. Такое движение может возникнуть при превышении в полете допустимых значений скоростей самолёта, что приводит к росту аэродинамических нагрузок и упругих деформаций элементов самолёта. Нагрузки на элементы конструкции планера возрастают и при росте углов атаки.

2. Современные методы защиты летательных аппаратов от критических режимов пилотирования

Обеспечение безопасности полёта при всех возможных условиях эксплуатации является важной задачей в авиации. Данная задача приобрела особую остроту в связи с непрерывным ростом пассажироместимости самолётов и высоким требованием регулярности полётов. Рациональный выбор схемы самолёта, его компоновки и органов управления и использование автоматики создают определенные возможности для повышения безопасности полёта. Применение современных средств автоматизации управления позволяют значительно повысить уровень безопасности полёта, в частности, за счёт уменьшения роли человеческого фактора, сильно сказывающегося на уровне аварий и катастроф.

Таким образом, при решении вопроса безопасности летательного аппарата, в настоящее время, основными факторами являются [3]:

- рациональный выбор параметров самолёта его схемы и компоновки;
- широкое применение автоматизации управления.

На современных самолётах используются бортовые цифровые вычислительные машины, которые при переходе от аналоговых вычислителей обеспечивают больший возможный объем алгоритмов управления и дают более высокую точность.

Применение систем улучшения устойчивости и управляемости позволяют лучше решать вопросы аэродинамической компоновки самолёта, выбора его основных параметров.

При разработке бортовой цифровой вычислительной машины учитываются:

- продольная устойчивость и управляемость;
- боковая устойчивость и управляемость;
- устойчивость и управляемость при взлете и посадке;
- устойчивость и управляемость на критических углах атаки.

Система управления самолёта и входящие в нее автоматические подсистемы, отвечающие за продольную устойчивость и управляемость должны обеспечивать необходимые характеристики самолёта и безопасность полёта в любом эксплуатационном режиме. При разработке данной системы целесообразно разделить все области полёта (по высотам, скоростям и перегрузкам) на три области: основную, эксплуатационную и предельную.

Для магистральных самолётов эти области рационально разделить по перегрузкам:

$0,6 \leq n_y \leq 1,4$ – основная область режимов полёта;

$0,4 \leq n_y \leq 1,7$ – эксплуатационная область режимов полёта;

$n_{y \min} \leq n_y < 0,4$
 $1,7 < n_y \leq n_{y \max}$ } – предельная область режимов полёта;

Пассажирские и транспортные самолёты имеют ряд особенностей, которые оказывают заметное влияние на характеристики боковой устойчивости и управляемости:

- большие моменты инерции самолёта, особенно относительно вертикальной оси, вследствие большого удаления крыла и размещения топлива в крыле;
- излишняя степень поперечной статической устойчивости, особенно с отклоненной механизацией крыла, вследствие больших положительных углов поперечного крыла;
- большие значения перекрестных вращательных производных;
- большое влияние упругости конструкции на эффективность элеронов и ряд других особенностей.

Под взлётом принято понимать движение самолёта с момента начала движения до момента набора безопасной высоты равной 400 метров и установки конфигурации самолёта для дальнейшего набора высоты и крейсерского полёта.

Взлетная дистанция состоит из участка движения самолёта по взлетно-посадочной полосе (впп), до момента отрыва и воздушного участка высоты равной 10,7 м. Примерная схема взлета самолета представлена на рисунке 1.

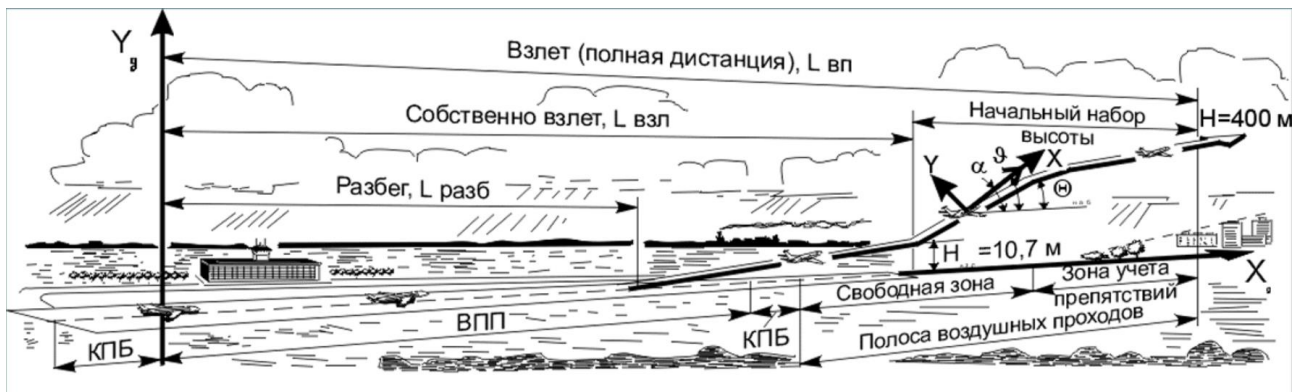


Рис. 1. Схема взлета самолёта

Во всех нормах летной годности предъявляются детальные требования к скоростям полёта на этапах взлёта, а именно:

- безопасная скорость, которая должна быть достигнута на высоте не более 10,7м;
- скорости первоначального набора высоты, которая должна быть достигнута до высоты начала уборки механизации крыла;
- скорости самолёта при полётной конфигурации;
- скорости подъема носового колеса.

Участок посадки самолёта начинается после снижения в зоне аэродрома назначения и заканчивается полной остановкой самолёта. Условно участок посадки подразделяется на несколько этапов. Примерная схема захода на посадку современных транспортных самолётов приведена на рисунке 2.

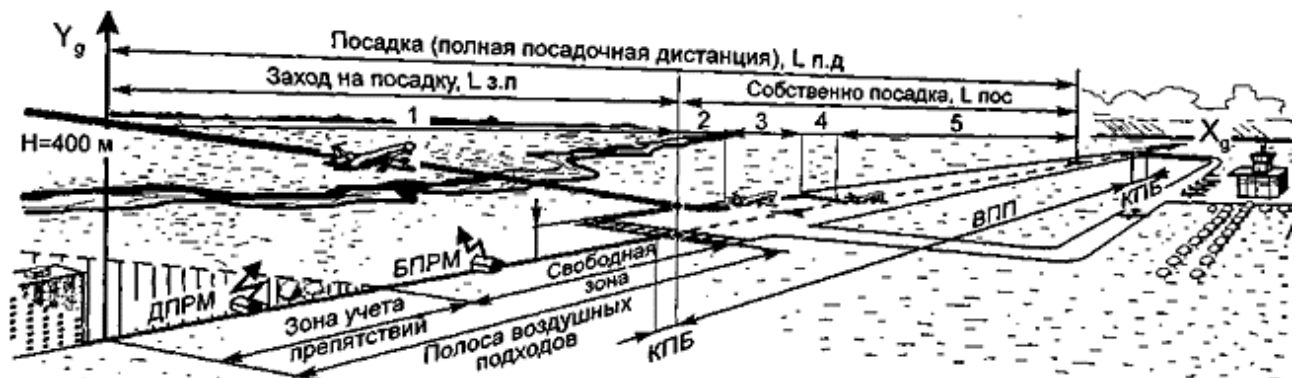


Рис. 2. Схема посадки самолёта

При посадке пилоту необходимо следить за [4]:

- пересечение самолёта кромки ВПП;
- положительным углом тангажа (3° - 4°);
- углом атаки должен быть близок к посадочному (8° - 10°);
- скорость захода на посадку;

Важным фактором при создании самолёта является обеспечение удовлетворительных характеристик устойчивости, управляемости и безопасности полёта

на больших углах атаки. Поэтому, еще на ранней стадии проектирования анализируются следующие предварительные аэродинамические характеристики:

- аэродинамические характеристики в большом диапазоне углов атаки ($-10^{\circ};60^{\circ}$) и скольжения ($-20^{\circ};20^{\circ}$)
- аэродинамические характеристики самолёта при числах M полёта
- нестационарные аэродинамические характеристики

Исходя из всего вышесказанного можно составить схему ключевых параметров, проверяемых вычислительной системой, для предотвращения попадания самолёта в различные критические ситуации. Данная схема представлена на рисунке 3.

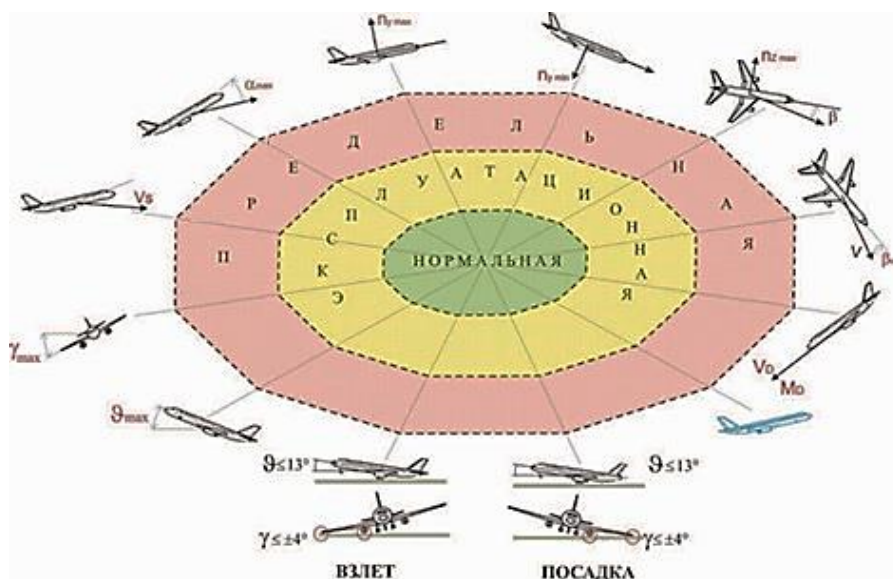


Рис. 3. Схема ключевых параметров летательного аппарата

3. Недостатки современных методов защиты летательных аппаратов от критических режимов пилотирования

Методы, описанные выше имеют ряд недостатков:

- большое количество обрабатываемой информации;
- отсутствие обрабатывания параметров как единой системы.

Важно отметить, что второй недостаток напрямую связан с первым. Так, во время полёта, вычислительные системы каждую секунду получают и обрабатывают свыше 700 параметров [5]. Для обработки такого большого объема данных необходима новейшая аппаратура, которой могут похвастаться только самолёты военной и большой авиации. В остальных же случаях, бортовой компьютер не рассматривает данные как единую систему, а проверяет параметры по-отдельности на предмет их приближения к критическому значению.

Решением этих параметров может стать вычислительная система, способная обрабатывать данные параллельно, а не один за одним, как это происходит сейчас. За счет этого, значительно снизится время обработки данных, и, как следствие, появится возможность рассматривать входящую информацию как единую систему.

4. Анализ вычислительной системы с аппаратом дискретной математики

В МГТУ им. Н.Э. Баумана разработана вычислительная система с многими потоками команд и одним потоком данных, в которой на аппаратном уровне реализован аппарат дискретной математики. Данная компьютерная система позволит сократить время обработки информации, поступающей на бортовой компьютер, и рассматривать полученные данные как единую систему. Процессор «Leonhard» новая универсальная вычислительная система. Она позволила сократить время обработки больших объемов данных за счет аппаратного ускорения действий над множествами и структурами данных. Процессор реализует набор команд высокого уровня над множествами и структурами данных. Новая архитектура позволяет более эффективно решать задачи дискретной оптимизации, основанные на моделях множеств, графов, и отношений.

Вычислительная система МКОД состоит как минимум из двух процессорных устройств [6]: Центрального процессора (ЦП) и Процессора обработки структур (СП) «Leonhard». Схема такой вычислительной системы представлена на рисунке 4. Устройства обмениваются командной информацией и данными через очереди. В МКОД системе Центральный процессор - это универсальное процессорное устройство, хранящее в своём ОЗУ команды и данные. ЦП выполняет инициализацию системы, запускает задачу на исполнение и обрабатывает результаты работы всего алгоритма. При этом основной вычислительный алгоритм разделяется на два потока команд: команды обработки данных, обрабатываемые ЦП, и команды обработки структур, обрабатываемые СП.

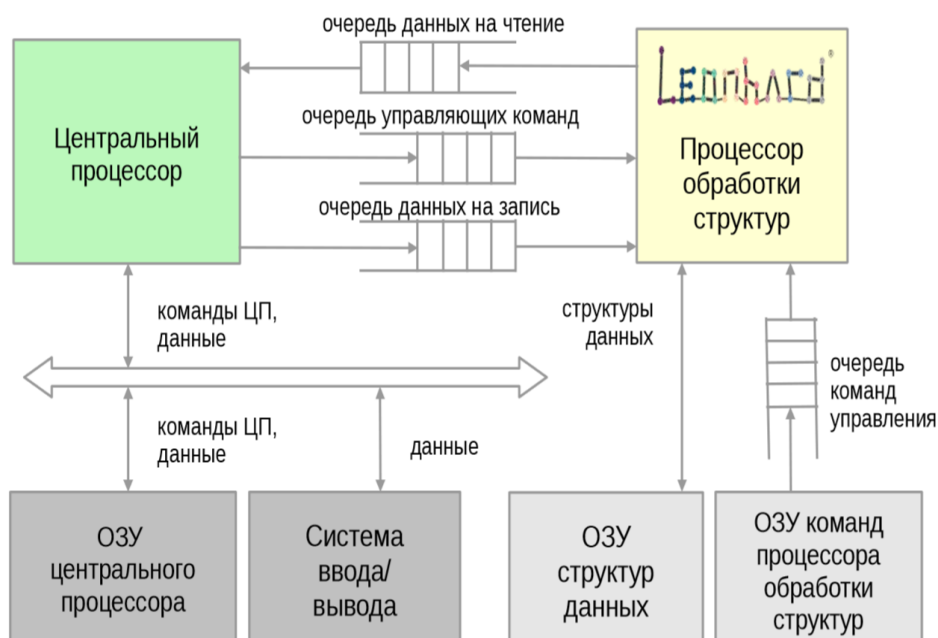


Рис. 4. Вычислительная система МКОД с хранением команд управления в ОЗУ

Процессор обработки структур подключён к локальным ЗУ, в которых хранятся структуры данных и команды их обработки. ЗУ команд представляет собой адресную

память, в которой располагаются отдельные процедуры управления СП, составляющие поток команд обработки структурной информации, такие как: поиск, добавление, удаление, объединение, поиск минимального или максимального и т.д. Доступ к содержимому адресного ЗУ выполняется с помощью уникальных ключей. Отличие СП от ЦП состоит в том, что первый содержит специальные устройства (так называемые Каталог, Операционный буфер, Устройство выборки команд, Память структур и другие), позволяющие быстро обрабатывать содержимое структур на основе ключей поиска[7]. Хранимая в СП информация представляет собой множества пар «ключ-значение», организованных в виде нескольких В+ деревьев. СП на аппаратном уровне обеспечивает все необходимые действия по обработке информации в В+ деревьях: освобождает и выделяет блоки памяти в ЗУ структур, выполняет добавление, удаление, поиск, осуществляет операции объединение, пересечение, дополнения деревьев, выполняет операции срезов. Выполнение основных команд в СП требует $\log n$ обращений к ЗУ структур, при этом применяются сильно ветвящиеся В+ деревья [8] со степенью вершин от 8 до 128, что позволяет эффективно работать с большими объемами информации. В реализованном варианте построения вычислительной системы МКОД, показанном на рисунке 5, команды СП содержатся в его локальном ЗУ команд и выбираются в естественном порядке следования. Для организации ветвлений набор команд СП включает также команды безусловного перехода, которые поступают из локального ОЗУ или от ЦП. СП не имеет собственного АЛУ и не выполняет проверки условий переходов. Управление ходом вычислений выполняет ЦП, передавая метку перехода в очередь команд управления, откуда ее извлекает СП и на основе этой информации выполняет переход. Обмен данными между ЦП и СП выполняется через две очереди данных: очередь данных на запись и очередь данных на чтение.

Приведенная на рисунке 5 схема построения МКОД системы обеспечивает возможность параллельной загрузки команд в оба процессора в ходе вычислительного процесса, а значит в большой степени соответствует структуре решаемых задач обработки структур данных. Это позволяет достичь большей параллельности при выполнении потоков команд обработки множеств. Вместе с тем, большое количество ветвлений в алгоритмах и связь по данным двух частей алгоритма может негативно сказаться на возможности параллельной работы СП и ЦП. Поэтому были исследованы ряд алгоритмов оптимизации на сетях и графах, которые позволили определить перспективные варианты совершенствования представленного подхода и сократить зависимости между двумя вычислительными потоками. Так, было предложено использовать устройство для генерации составных ключей, что позволит использовать результаты выполнения предыдущих команд в качестве частей новых операндов. Как показал анализ, это позволяет существенно снизить зависимость потоков по данным.

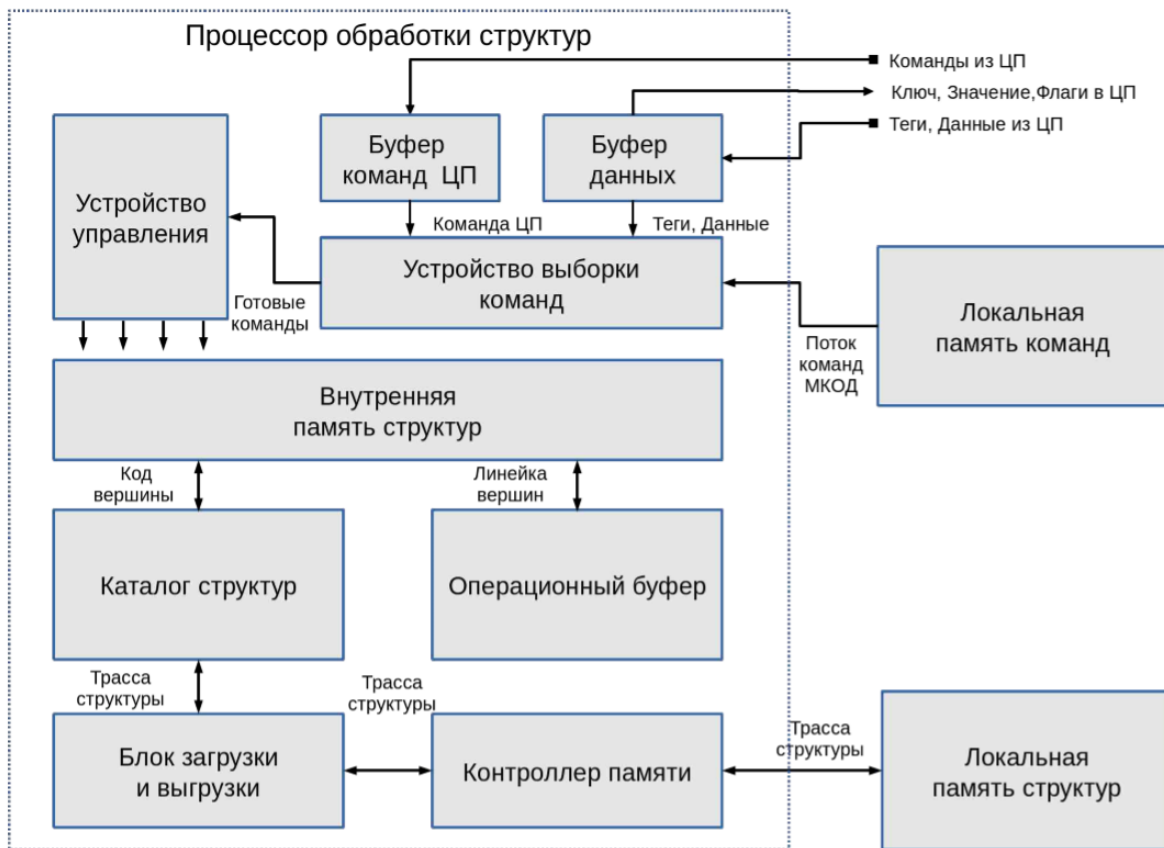


Рис. 5. Функциональная схема процессора обработки структур

Результаты тестирования на алгоритме Дейкстры показывают, что процессор обработки структур позволяет повысить эффективность вычислительной системы, так как во всех проведенных экспериментах достигнуто сокращение количества тактов для решения задачи. При этом аппаратная сложность СП примерно в 900 раз меньше, чем у микропроцессора Intel Core i5, а затраченный объем электроэнергии для решения задач в МКОД системе оказывается примерно в 5 раз ниже. Полные результаты тестирования представлены на рисунке 6.

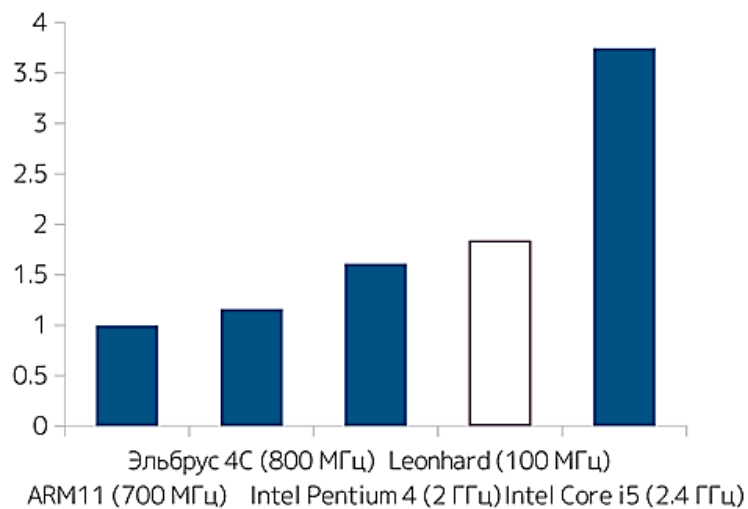


Рис. 6. Сравнение производительности микропроцессоров

В составе микропроцессора реализованы устройства для объединения, пересечения и разности множеств (И-ИЛИ-НЕ операции), а также операции срезов (LS,LSEQ,GR,GREQ). Уникальные блоки в составе микропроцессора представляют собой устройства с микроархитектурой SIMD: Каталог и операционный буфер. Команды NSM и NGR позволяют выполнить поиск ближайшего элемента множества при интерполяции данных, что существенно ускоряет алгоритмы машинного обучения. В текущей версии микропроцессора «Leonhard» реализован набор из 20 высокоуровневых машинных инструкций.

5. Алгоритм «К Ближайших Соседей»

Исходя из способностей микропроцессора «LeonHard» метод машинного обучения «К Ближайших Соседей», в основе которого лежит [9] построение графов и поиск кратчайшего расстояния до соседней вершины.

Рассмотрим пример применения алгоритма. Для простоты визуализации будут рассматриваться только 2 параметра (скорость летательного аппарата, угол тангажа) и 3 критических ситуации (штопор, сваливание, кренение). Важно отметить, что данные примеры взяты исключительно для простоты визуализации читателем и не несут под собой физического смысла.

Первым этапом необходимо обучить систему с помощью авиасимулятора на предмет распознавания той или иной критической ситуации. Результаты обучающей выборки представлены на рисунке 7.

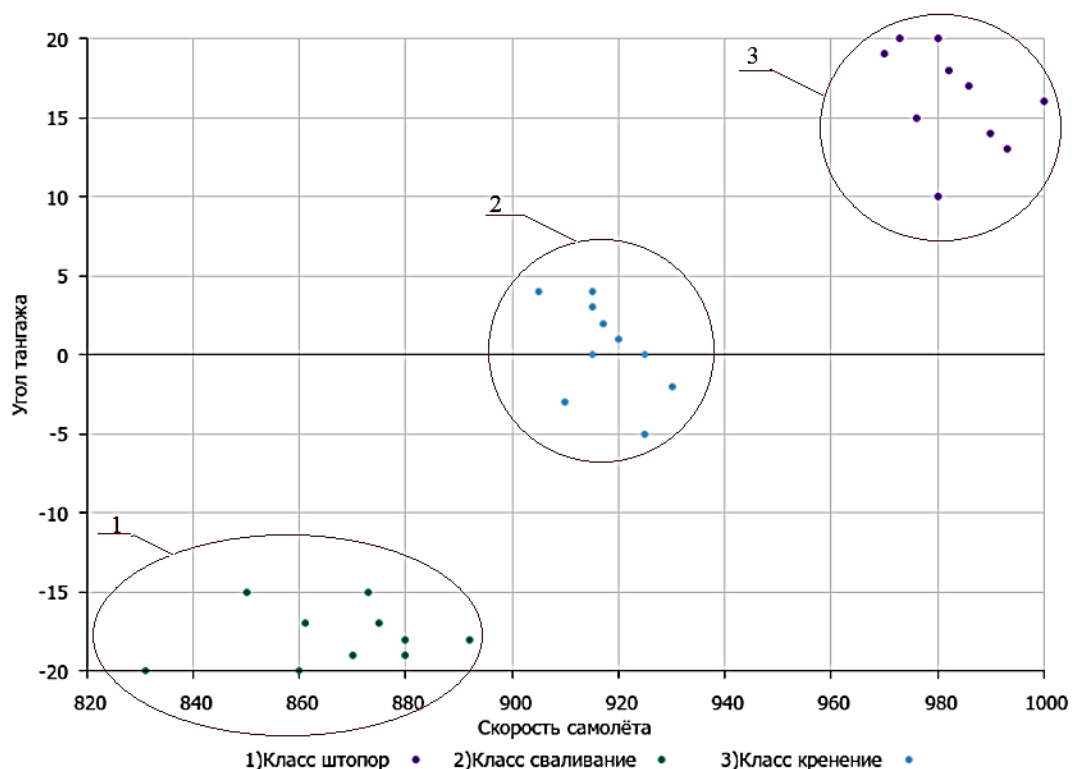


Рис. 7. Диаграмма размещения классов

Допустим, необходимо определить 2 состояния, на рисунке 8 показано размещение новых состояний среди уже известных.

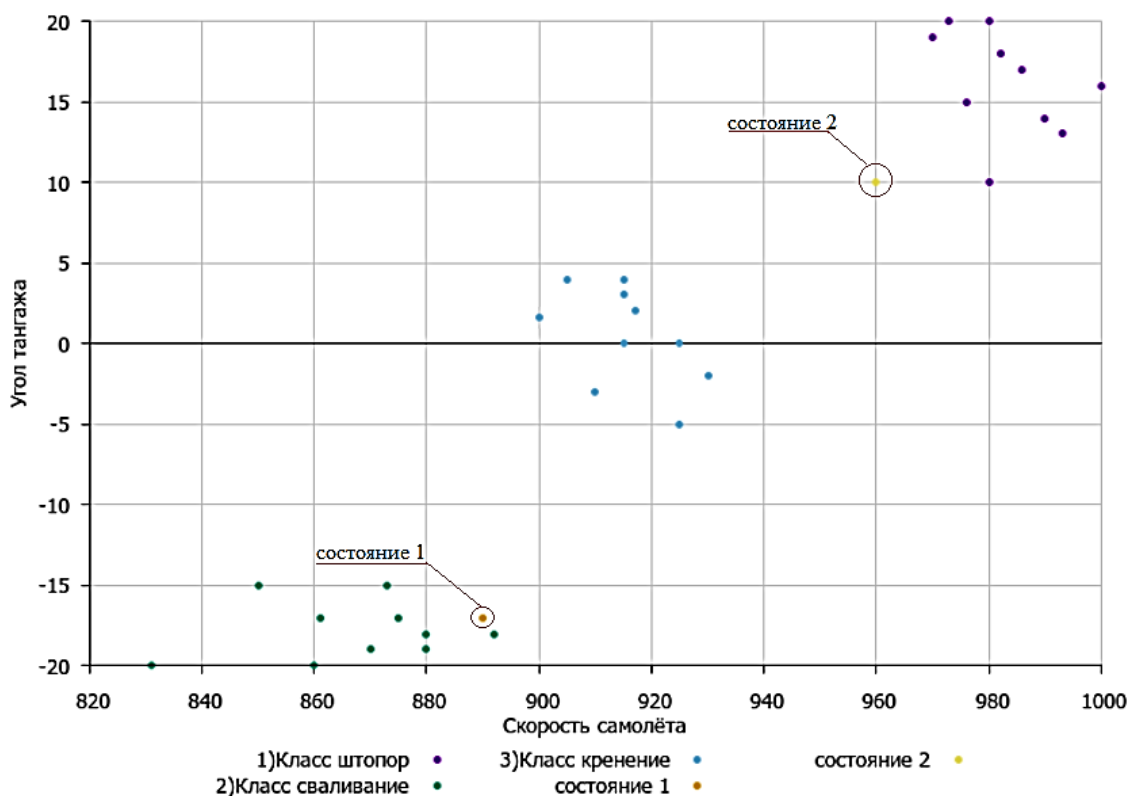


Рис. 8. Обновленная диаграмма размещения классов

Рассмотрим первое состояние с параметрами (890; -17). Установим значение $k=3$. Следовательно, нужно найти 3 ближайших «соседей». Для первого состояния, это будут состояния, со значениями: А (880; -18), В (880; -19) С (892; -18). Рассчитаем евклидово расстояние до каждого «соседа» по формуле (1):

$$D_E = \sqrt{\sum_i^n (x_i - y_i)^2}, \quad (1)$$

где n – количество атрибутов.

Соответственно расстояния будут равны:

$$D_A = 10,05$$

$$D_B = 10,20$$

$$D_C = 2,24$$

Полученные результаты представлены в таблице 1.

Таблица 1.

Параметры первого состояния и его «соседей»

Запись	Скорость	Угол тангажа	Расстояния	Состояние
Состояние 1	890	-17	-	-
A	880	-18	10,05	Сваливание
B	880	-19	10,2	Сваливание
C	892	-18	2,24	Сваливание

Очевидно, что состояние 1 также будет относиться к состоянию сваливание.

Теперь определим класс второго состояния с параметрами (960;10). Также зададим $k=3$ и по формуле (1) рассчитаем расстояния до ближайших «соседей». Здесь это будут состояния с параметрами: A (976 ;15), B(980;10), C(925;0). Соответственно расстояния до них будут равны:

$$D_A = 16,76$$

$$D_B = 20,00$$

$$D_C = 36,40$$

Полученные результаты представлены в таблице 2.

Таблица 2.

Параметры второго состояния и его «соседей»

Запись	Скорость	Угол тангажа	Расстояния	Состояние
Состояние 2	960	10	-	-
A	976	15	16,76	Штопор
B	980	10	20,00	Штопор
C	925	0	36,40	Кренение

Т.к. в данном случае не все «соседи» одного класса, то необходимо воспользоваться взвешенным голосованием. Известно, чем больше евклидово расстояние, тем менее значимый вклад вносит голос. Голоса за класс находятся по формуле (2):

$$votes(class) = \sum_{i=1}^n \frac{1}{d^2(x', y_i)} , \quad (2)$$

где $d^2(x', y_i)$ – квадрат расстояния от известной записи Y_i , до новой X' , n – количество известных записей класса, для которого рассчитываются голоса

Таким образом, рассчитав голоса для состояний получились следующие результаты:

- $votes(\text{штопор}) = 0,00606$
- $votes(\text{кренение}) = 0,00075$

Так как, значение $votes(\text{штопор})$ больше чем $votes(\text{кренение})$, то можно сделать вывод, что второе состояние относится к классу штопор.

Данный метод считается одним из наиболее точных в машинном обучении. Его ключевым недостатком являлась большая ресурсозатратность, но благодаря процессору «LeonHard», данный недостаток нивелируется.

6. Алгоритм работы программы

Исходя из всего вышесказанного можно составить схему работы нового метода защиты летательного аппарата от попадания в критические ситуации. Ее можно разделить на следующие этапы:

- отправление авиасимулятором данных для обучения системы;
- обучение системы;
- внедрение в бортовую систему самолёта;
- работа.

Функциональная схема итогового программного продукта представлена на рисунке 9.

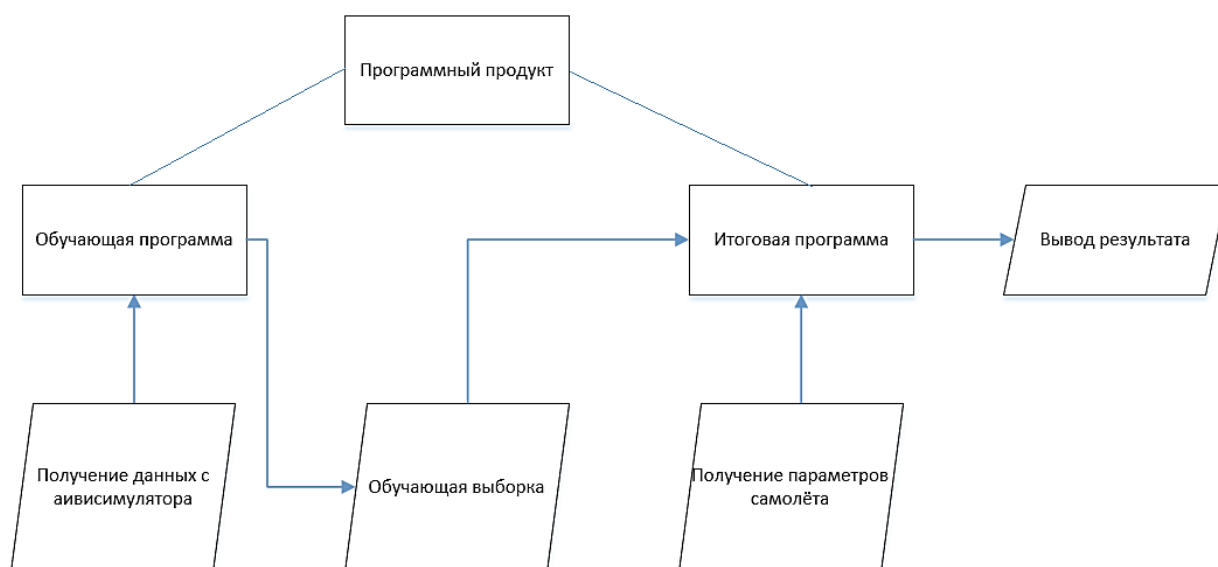


Рис. 9. Функциональная схема программного продукта

Заключение

В работе выполнен анализ различных критических ситуаций, описаны современные методы защиты летательных аппаратов от попаданий в них. Были выявлены ключевые недостатки, не позволяющие получить более точные результаты с помощью методов машинного обучения.

В качестве решения был предложен метод, основанный на алгоритмах машинного обучения и аппаратной платформы дискретной математики. Выполнен анализ данной платформы. В качестве метода машинного обучения был выбран алгоритм «К Ближайших Соседей», который также описан выше.

В результате получена функциональная схема итогового программного продукта, предназначенного предупреждать о возможном попадании в критическую ситуацию.

Список литературы

1. Статистика крупнейших авиакатастроф мира 1974-2018 // ФОРиншурер. Режим доступа: <https://forinsurer.com/public/17/01/10/3824> (дата обращения: 01.05.2018).
2. Бочкарёв А.Ф., Андреевский В.В., Белоконов В.М., Климов В.И., Турапин В.М. Аэромеханика самолёта. 2-е изд. / Под ред. Бочкарева А.Ф., Андреевского В.В. М.: Машиностроение. 1985. 356 с.
3. Федунев Б.Е. Бортовые оперативно-советующие экспертные системы – новый класс алгоритмов управления // International Book Series “Information Science and Computing”. 2008. № 6. С. 123-132.
4. Бюшгенс Г.С. Аэродинамика и динамика полета магистральных самолетов. М.: Пекин: Издательский отдел ЦАГИ - Авиа-Издательство КНР. 1995. 772 с.
5. Bruce Williams. Scenario-Based Training with X-Plane and Microsoft Flight Simulator: Using PC-Based Flight Simulations Based on FAA-Industry Training Standards. 1st Ed. John Wiley & Sons Limited. 2012. 624 p.
6. Попов А.Ю., Старчак С.Л., Шаменков Н.А. Вопросы применения вычислительной системы с многими потоками команд и одним потоком данных для повышения показателей эффективности программных и аппаратных средств ВКО // Проблемы создания и испытания вооружения РКО: Сборник научных трудов НИИЦ (г. Москва) ЦНИИ ВВКО МО РФ. М.: ЦНИИ ВВКО МО РФ. 2015. Ч. 2. С. 75 – 99, инв. 1975 с.
7. Harris D.M., Harris S.L. Digital Design and Computer Architecture. 2nd Ed. Boston: Morgan Kaufmann. 2012. 721 p.
8. Попов А.Ю. Применение вычислительных систем с многими потоками команд и одним потоком данных для решения задач оптимизации // Вестник МГТУ им. Н.Э. Баумана. Сер. Приборостроение. 2012. №1(1). Спец. вып. Компьютерные системы и технологии. С. 146 -154. DOI: 10.18698/2308-6033-2012-1-80
9. A Complete Guide to K-Nearest-Neighbors with Applications in Python and R // Github.io. Режим доступа: <https://kevinzakka.github.io/2016/07/13/k-nearest-neighbor/> (дата обращения: 3.05.2018).

Анализ моделей классификации текстовой информации

¹³Гуренко В.В., ¹⁴Гузий Е.А., ¹⁵Федоренков В.В.

МГТУ им. Н.Э. Баумана (НИУ), Москва, Россия

wgurenko@bmstu.ru

zhenya.guziy@mail.ru

ferrari2535@mail.ru

Аннотация: Работа посвящена сравнительному анализу основных моделей классификации текстов на естественном языке: Байесова классификатора, классификатора на основе метода k ближайших соседей и классификатора на основе метода опорных векторов. Выполнено экспериментальное исследование данных моделей на обучающей и тестовой выборках. Сравнение осуществлялось по точности классификации. Показано, что классификация по методу опорных векторов имеет преимущество перед остальными моделями. Результаты могут быть применены для решения прикладных задач в области автоматического анализа текстовой информации и разработки соответствующих программных систем.

Работа выполнена в рамках проекта 2.7782.2017/БЧ, осуществляемого при поддержке Министерства образования и науки Российской Федерации.

Ключевые слова: классификатор, ансамбль, наивный Байесов классификатор, метод k ближайших соседей, метод опорных векторов, библиотека Scikit-Learn.

Analysis of classification models for textual information

Gurenko V.V., Gusiy E.A., Fedorenkov V.V.

Bauman Moscow State Technical University, Moscow, Russia

wgurenko@bmstu.ru

zhenya.guziy@mail.ru

ferrari2535@mail.ru

Abstract: The work is devoted to a comparative analysis of the main models of text classification in natural language: Bayesian classifier, classifier based on the method to the nearest neighbors and a classifier based on the method of support vector machine. Experimental research of models was carried out on the training and test samples. The comparison was made according to the classification accuracy. It is shown that the

¹³Гуренко Владимир Викторович – доцент кафедры "Компьютерные системы и сети" МГТУ им. Н.Э. Баумана, SPIN-код 2675-4796.

¹⁴Гузий Евгений Андреевич – магистрант кафедры "Компьютерные системы и сети" МГТУ им. Н.Э. Баумана.

¹⁵Федоренков Владислав Владимирович – магистрант кафедры "Компьютерные системы и сети" МГТУ им. Н.Э. Баумана.

classification by the method of support vector machine has an advantage over other models. The results can be applied to solve applied problems in the field of automatic analysis of textual information and the development of appropriate software systems.

Keywords: classifier, ensemble, naive Bayes classifier, k-nearest neighbors method, support vector machine method, Scikit-Learn.

Введение

Задача классификации текстовой информации на естественном языке занимает значимое место среди задач машинного обучения и анализа данных. К областям применения классификации текстов как научно-практической задачи можно отнести следующие [1]:

- разделение веб-сайтов по тематическим каталогам;
- борьба со спамом;
- распознавание эмоциональной окраски текстов;
- персонализация рекламы.

Классификация текстовых документов заключается в отнесении документа к одному из заранее известных классов. В таком контексте классификацию нередко называют категоризацией или рубрикацией [2], что объясняется заимствованием терминологии из практики разделения документов по каталогам, категориям и рубрикам. Структура каталогов может быть, как одноуровневой, так и иерархической.

Формально задачу классификации текстовых документов можно описать с помощью аппарата теории множеств. Если обозначить множество документов как $D = \{d_i\}, i = 1, \dots, n$, а множество классов документов как $C = \{c_r\}, r = 1, \dots, m$, то задача классификации будет состоять в построении сюръективного отображения F множества D в множество C :

$$F : D \rightarrow C. \quad (1)$$

Иными словами, каждому документу из множества D необходимо сопоставить некоторый класс из множества C , причем каждый документ может входить только в один класс, а каждый класс может содержать произвольное количество документов.

Целью настоящей работы является выявление достоинств и недостатков основных применяемых на практике моделей, реализующих описанную задачу классификации, и оценка их точности. К исследуемым моделям отнесены следующие: наивный Байесов классификатор, классификатор на основе метода k ближайших соседей и классификатор на основе метода опорных векторов. Для оценки точности применено экспериментальное исследование на выборке из потока клиентских заявок одного из обслуживающих центров.

1. Характеристики, достоинства и недостатки различных моделей классификаторов

Наивный Байесов классификатор базируется на теореме Байеса о вычислении условной вероятности. Формула Байеса имеет следующий вид [3]:

$$p(A|B) = \frac{p(B|A) * p(A)}{p(B)}, \quad (2)$$

где $p(A)$ — априорная вероятность гипотезы A , $p(A|B)$ — апостериорная вероятность гипотезы A при наступлении события B , $p(B|A)$ — вероятность наступления события B при истинности гипотезы A , $p(B)$ — полная вероятность наступления события B . Применим формулу Байеса к задаче классификации. Для этого введём обозначения. Пусть c_j — j -ый класс из множества C , d_k — k -ый документ из множества D . С помощью данной теоремы можно вычислить вероятность присвоения класса c_j документу d_k при условии, что известна вероятность отношения документа d_k к классу c_j и независимые вероятности для c_j и d_k .

Таким образом, формула Байеса примет следующий вид:

$$p(c_j | d_k) = \frac{p(d_k | c_j) * p(c_j)}{p(d_k)}. \quad (3)$$

Для данной модели d_k представляет собой набор слов $d_k = \{w_1, w_2, \dots, w_n\}$. В таком случае вероятность $p(d_k | c_j)$ можно вычислить как сумму вероятностей для каждого слова $w_i \in d_k$. Учитывая свойство «наивности» данного классификатора, выражение для $p(d_k | c_j)$ примет следующий вид:

$$p(d_k | c_j) = p(w_1, w_2, \dots, w_n) = p(w_1 | c_j) p(w_2 | c_j) \dots p(w_n | c_j) = \prod_{i=1}^n p(w_i | c_j). \quad (4)$$

Знаменатель может быть опущен, так как для одного и того же документа d_k вероятность $p(d_k)$ будет одинаковой, а это значит, что ее можно не учитывать.

Наиболее вероятный класс c^* , которому принадлежит документ d_k , тот, при котором условная вероятность принадлежности документа d_k классу $c_j \in C$ максимальна:

$$c^* = \operatorname{argmax}_{c_j \in C} p(c_j | d_k). \quad (5)$$

Учитывая всё выше написанное, получим:

$$c^* = \operatorname{argmax}_{c_j \in C} [p(w_1, w_2, \dots, w_n | C) * p(c_j)]. \quad (6)$$

Достоинства наивного Байесова классификатора:

– высокая скорость классификации, в том числе для большого числа классов;

– высокая точность классификации. При допущении независимости признаков, наивный Байесов классификатор превосходит по точности отнесения документов к классам многие другие классификаторы, требуя при этом меньшего объема обучающих данных [3].

Из достоинств следует основной недостаток наивного Байесова классификатора — допущение о независимости классификационных признаков.

Вторым известным методом классификации текстовых данных является метод k ближайших соседей. В отличие от других методов, используемых для решения задачи классификации, данный метод не требует обучения. Для того чтобы определить, к какому классу относится документ d_j , классификатор сравнивает d_j со всеми документами из обучающей выборки и для каждого d_z , который принадлежит обучающей выборке, вычисляется расстояние между этими векторами. Далее среди d_z следует выбрать k ближайших соседей.

Пусть $d_j = \{w_1, \dots, w_n\}$, $d_z = \{w'_1, \dots, w'_n\}$. Обучающая выборка E представляется кортежем $\langle d_z, c_z \rangle$, где c_z — класс, который соответствует документу d_z . Чтобы оценить близость двух векторов, необходимо вычислить угол ρ между ними:

$$\rho(d_j, d_z) = \cos(d_j, d_z) = \frac{d_j * d_z}{|d_j| * |d_z|} = \frac{\sum_{i=1}^n (w_i * w'_i)}{\sqrt{\sum_{i=1}^n w_i^2 * \sum_{i=1}^n w_i'^2}}. \quad (7)$$

Далее выбираются k классов по принципу минимальности значений $\rho(d_j, d_z)$. Число k выбирается для более точной оценки принадлежности документа к тому или иному классу.

Метод k ближайших соседей является высокоэффективным, но требует больших вычислительных затрат. К достоинствам классификатора на основе метода k ближайших соседей (kNN) можно отнести:

- простоту реализации и возможность введения различных модификаций;
- возможность интерпретировать классификацию документа путем предъявления пользователю ближайшего документа из нескольких.

Недостатки классификатора kNN состоят в:

- полном хранении обучающей выборки, что влечет чрезмерные затраты памяти;
- сложности алгоритма по отысканию расстояний от классифицируемого документа до остальных для большой обучающей выборки;
- отсутствии четких критериев выбора параметра k .

Существует метод классификации, основанный на разделении классов гиперплоскостью. Он называется метод опорных векторов (SVM — сокр. от англ. Support vector machine). В наиболее простом случае метод опорных векторов заключается в нахождении разделяющей гиперплоскости на множестве признаков, разделяющей n -

мерное арифметическое пространство R^n на две части: в одной находятся все положительные примеры (документы, принадлежащие данному классу), а в другой – все отрицательные (документы, не принадлежащие этому классу). При этом ширина этой гиперплоскости должна быть как можно шире, чтобы происходило более точное выделение классов.

Пусть заданы множество векторов обучающей выборки $\{\vec{x}_1, \dots, \vec{x}_n\} \subset R^n$ и множество чисел $\{y_1, \dots, y_n\} \subset \{-1, 1\}$. Число $y_i = 1$, если вектор \vec{x}_i относится к классу c_j , и -1 — в противном случае [4]. Для данной задачи целесообразно применить линейный классификатор. Необходимо найти гиперплоскость, которая отделяет все точки одного класса от всех точек другого класса. Если такая гиперплоскость существует, то задача классификации сводится к определению взаимного расположения точки и данной гиперплоскости. Если точка расположена с одной стороны гиперплоскости, то она принадлежит классу c_j , если с другой стороны — то не принадлежит данному классу. В формальной постановке задачи требуется найти такой вектор \vec{w} , что для некоторых значений b и \vec{x}_i выполнялось условие:

$$y_i = \begin{cases} 1, & \vec{w}\vec{x}_i \geq b \\ -1, & \vec{w}\vec{x}_i < b \end{cases}, \quad (8)$$

где \vec{w} — вектор нормали к гиперплоскости (с его помощью можно узнать ширину коридора), b — параметр сдвига в уравнении гиперплоскости.

Таким образом, уравнение гиперплоскости примет вид $\vec{w}\vec{x}_i = b$. Если точка лежит левее гиперплоскости ($\vec{w}\vec{x}_i < b$), то она принадлежит классу c_j . Если правее, то не принадлежит ему. Вектор \vec{w} ортогонален гиперплоскости, а значение b зависит от кратчайшего расстояния между этой гиперплоскостью и началом координат.

Необходимо определить, какая гиперплоскость наилучшим образом разделяет классы. Очевидно, что этому свойству отвечает та, которая максимально удалена от ближайших до нее точек, принадлежащих обоим классам. Задачей метода SVM является отыскание таких \vec{w} и b , чтобы выполнялось следующее условие:

$$\begin{cases} \vec{w}\vec{x}_i - b \geq 1, & y_i = 1, \\ \vec{w}\vec{x}_i - b \leq -1, & y_i = -1. \end{cases} \quad (9)$$

Данная система задает коридор, разделяющий классы. Чем шире этот коридор, тем увереннее можно проводить классификацию. Необходимо сформулировать условие задачи оптимальной разделяющей полосы (коридора). Ни одна из точек не может лежать внутри коридора.

После выполнения преобразований над выражением (8) получается задача многокритериальной оптимизации с $n + 1$ условием [4]:

$$\left\{ \begin{array}{l} \lambda_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \lambda_i \lambda_j y_i y_j (\overrightarrow{x_i} \overleftarrow{x_j}) \rightarrow \max, \\ \sum_{i=1}^n \lambda_i y_i = 0, \\ \lambda_i \geq 0, i = 1, \dots, n, \end{array} \right. \quad (10)$$

где λ_i — i -я координата вектора двойственных переменных, полученных по теореме Куна-Таккера из системы (8).

Можно обобщить метод опорных векторов на случай нескольких классов (больше двух). Существует два подхода к обобщению: «один против всех» и «один против одного». При подходе «один против всех» для каждого класса строится отдельный классификатор, отделяющий этот класс от всех остальных. При втором подходе строится $n(n-1)/2$ классификаторов (n — число классов), которые отделяют каждые два класса друг от друга. Новый объект относят к тому классу, к которому его отнесло большинство классификаторов (функция моды).

Классификация по методу SVM обладает определенными преимуществами:

- на тестах с документальными массивами превосходит другие методы по точности классификации;

- при выборе разных функций в качестве ядра (линейное ядро, полиномиальное ядро, Гауссова радиальная базисная функция и т.д.) позволяет реализовать другие подходы. Например, большой класс нейронных сетей можно представить с помощью SVM со специальными ядрами;

- итоговое правило классификации определяется путем вычисления некоторой целевой функции, а не с помощью экспертных эвристик.

К недостаткам метода можно отнести:

- недостаточное число параметров настройки: после того, как фиксируется функция ядра, единственным изменяемым параметром остается коэффициент ошибки;

- отсутствуют четкие критерии выбора ядра;

- медленное обучение системы классификации.

2. Экспериментальная оценка и выбор классификатора

Для проведения экспериментального исследования моделей классификаторов, описанных в предыдущем разделе, была использована библиотека машинного обучения Scikit-Learn. С помощью этой библиотеки был получен расчёт метрики оценки качества классификаторов.

Все приведённые в этой статье классификаторы относятся к классификаторам, обучаемыми с учителем. Обучение с учителем подразумевает наличие пары «текст заявки-класс», на основе которой строится зависимость между заявками, которые необходимо отнести к какому-либо классу, и заранее известными классами.

В качестве обучающей выборки были использованы 2153 записи (одна запись представляет собой одну клиентскую заявку). В качестве тестовой выборки (это часть обучающей выборки, для которой выполняется процедура классификации с целью получения значений точности) были выбраны 1432 записи.

Поскольку заявки представляются в текстовом виде, непонятном для классификатора, необходимо выполнить процедуру предварительной обработки текста. Эта процедура включает в себя формирование первичного словаря с использованием процедуры токенизации (выделение токенов); формирование вторичного словаря из первичного с помощью стемминга (выделения основы слова); удаления стоп-слов (слов, не несущих никакой смысловой нагрузки для данного текста); приведения к одному регистру (эта операция необходима для того, чтобы одни и те же слова, начинающиеся со строчной или прописной буквы, воспринимались классификатором одинаково); векторизацию вторичного словаря (перевод слов текста в цифровой вектор для дальнейшей обработки классификатором). Существует несколько видов мер, используемых при векторизации вторичного словаря:

– TF. Определяется как отношение числа вхождения некоторого термина к общему количеству терминов документа (в данном случае документ взаимно-однозначно соответствует клиентской заявке из выборки);

– IDF. Определяется как отношение количества документов в коллекции к количеству документов, в которых встречается некоторый терм;

– TF-IDF. Определяется как произведение предыдущих двух метрик. Позволяет оценить важность термина в контексте документа, являющегося частью коллекции документов.

В ходе экспериментальной оценки алгоритмов исследовались следующие показатели работы классификаторов:

– точность (accuracy),

– время выполнения обучения и классификации.

Точность определяется по следующей формуле:

$$accuracy = \frac{P}{N}, \quad (11)$$

где P — количество документов, по которым классификатор принял правильное решение, N — размер обучающей выборки.

Исследование проводилось как с использованием методов предварительной обработки текстовой информации, так и без них. Результаты исследования показаны в таблице 2. Знак «+» в ячейках таблицы означает, что в ходе данного исследования применялся тот метод предобработки, который находится в заголовке данного столбца таблицы. Знак «-» означает, что данный метод не применялся. Параметр точности измеряется в долях. Эксперимент показал, что, используя только метрику TF в наивном классификаторе Байеса, точность классификации оказалась низкой (0.73). Применяв метрику взвешивания терминов TF-IDF, точность увеличилась на 0.05 единиц (почти на 6

% относительно точности, полученной в первом эксперименте). В каждом следующем эксперименте добавлялся новый метод предобработки текстовой информации на естественном языке. За счёт чего для Байесова классификатора удалось достичь предельной точности 0.87 единиц. Время обучения и классификации (в таблице 1 обозначено как t^*) уменьшалось по мере добавления нового метода предобработки. Это говорит о том, что словарь термов уменьшается, и применение этих методов позволяет сократить время обучения и классификации. Время обучения и классификации, которое показал наивный Байесов классификатор, при применении всех методов предобработки составляет 1.05 секунды. Классификатор на основе SVM с применением всех методов предобработки текстовой информации показал точность, равную 0.91 единиц. Этот результат на 4 процента лучше, чем тот, который показал наивный Байес. Время обучения и классификации для SVM на данной выборке составило 1.06 с, что ненамного хуже, чем время работы Байесова классификатора. Однако стоит учитывать, что при возрастании числа заявок разница между временем работы этих классификаторов начнёт заметно отличаться (время работы SVM станет значительно больше). Классификация по методу k ближайших соседей проводилась несколько раз, поскольку необходимо было определить предельное улучшение точности. При увеличении параметра k до 15 точность классификации возрастала. Далее точность перестала улучшаться, а при дальнейшем увеличении параметра k стала и вовсе ухудшаться. Предельное значение точности на данной выборке составляет 0.87 единиц. Время работы классификатора kNN уже при $k = 2$ значительно превышает время обучения и классификации как наивного Байесова классификатора, так и SVM.

Таблица 1.

Итоги сравнительного анализа классификаторов

Вид классификатора	Стемминг	Удаление стоп-слов	Приведение к одному регистру	Векторизация	Точность	t^* , с	
Наивный мультиномиальный Байесов	-	-	-	TF	0.7304	1.8	
	-	-	-	TF-IDF	0.7884	1.7	
	-	-	+	TF-IDF	0.8017	1.5	
	-	+	+	TF-IDF	0.8561	1.2	
	+	+	+	TF-IDF	0.8734	1.05	
SVM (по методу опорных векторов)	+	+	+	TF-IDF	0.9092	1.06	
kNN (по методу k ближайших соседей)	$k = 2$	+	+	+	TF-IDF	0.8589	1.21
	$k = 5$	+	+	+	TF-IDF	0.8547	1.20
	$k = 12$	+	+	+	TF-IDF	0.8666	1.27
	$k = 20$	+	+	+	TF-IDF	0.8645	1.21

Результаты оценки параметров классификаторов можно показать на лепестковой диаграмме (см. рис. 1).



Рис. 1. Лепестковая диаграмма сравнения классификаторов

Как видно из этой диаграммы, классификатор SVM показал наилучшее значение точности классификации, но по времени обучения и классификации он хуже наивного Байесова классификатора, но лучше kNN. kNN показал наихудшие результаты по времени обучения и классификации. Наивный Байес показал наихудшие результаты по точности классификации.

Заключение

В результате проведённого исследования моделей классификации текстовой информации были выделены основные преимущества и недостатки Байесова классификатора, классификатора на основе метода k ближайших соседей и классификатора на основе метода опорных векторов. Проведенный с использованием библиотеки Scikit-Learn практический анализ классификаторов по точности классификации показал, что классификация методом опорных векторов имеет значительное преимущество по сравнению с другими моделями. Время выполнения обучения и классификации у SVM меньше, чем у kNN, но больше, чем у наивного Байеса. В дальнейшем рекомендуется использовать данную модель для построения ансамблей классификаторов клиентских заявок службы технической поддержки с целью увеличения показателя точности классификации по сравнению с базовым SVM классификатором.

Список литературы

1. Bickel S. ECML/PKDD Discovery Challenge 2006 Overview. Режим доступа: http://www.ecmlpkdd2006.org/discovery_challenge2006_overview.pdf (дата обращения: 02.05.2018).
2. Барсигян А.А. Анализ данных и процессов. Учеб. Пособие. СПб.: БХВ-Петербург. 2009. 512 с.
3. Вентцель Е.С. Теория вероятностей. Учебное пособие. М.: Высшая школа. 1999. 576 с.
4. Вьюгин В.В. Математические основы теории машинного обучения и прогнозирования. М.: МЦМНО. 2013. 390 с.

Обработка цифровых сигналов в Уолше-подобном базисе

¹⁶Ким Т.А., ¹⁷Гуренко В.В., ¹⁸Сотников А.А.

МГТУ им. Н.Э. Баумана (НИУ), Москва, Россия

wgurenko@bmstu.ru

alex@sotnikov.com

kimtamaralex@gmail.com

Аннотация: Работа посвящена исследованию и разработке новых методов синтеза вещественных функций Уолша, анализу их свойств и реализации данных алгоритмов в виде программной системы для четверичной системы счисления. Согласно теории цифровой обработки сигналов один и тот же сигнал можно разложить по различным системам базисных функций (СБФ). В различных СБФ спектральному анализу соответствует различная физическая и практическая реализации. Полных и ортогональных СБФ существуют бесчисленное множество.

В работе рассмотрены классические системы базисных функций Уолша. Приведено математическое описание новых Уолше-подобных функций, выполнено исследование их основных свойств, способов построения различных ортогональных базисных систем на их основе. Описаны более быстрые способы получения систем, основанные на переупорядочении функций. Приведены схемы алгоритмов фрагментов программной системы, позволяющей автоматизировать процесс получения систем, представленных в виде квадратных матриц любой размерности. Результаты работы носят оригинальный характер и применимы для проведения дальнейших исследований в области спектральной обработки цифровых дискретных сигналов.

Работа выполнена в рамках проекта 2.7782.2017/БЧ, осуществляемого при поддержке Министерства образования и науки Российской Федерации.

Ключевые слова: цифровая обработка сигналов, функции Уолша, упорядочивание Адамара, упорядочивание Пэли, упорядочивание Хармута, дискретные сигналы, спектральный анализ.

Processing of digital signals in a Walsh-like basis

Kim T.A., Gurenko V.V., Sotnikov A.A.

Bauman Moscow State Technical University, Moscow, Russia

¹⁶Ким Тамара Александровна – бакалавр кафедры "Компьютерные системы и сети" МГТУ им. Н.Э. Баумана.

¹⁷Гуренко Владимир Викторович – доцент кафедры "Компьютерные системы и сети" МГТУ им. Н.Э. Баумана, SPIN-код 2675-4796.

¹⁸Сотников Алексей Александрович – кандидат технических наук, доцент кафедры «Компьютерные системы и сети» МГТУ им. Н.Э. Баумана, SPIN-код 1754-5370.

wgurenko@bmstu.ru

***Abstract:** The work is devoted to the study and development of new methods for the synthesis of real Walsh functions, analysis of their properties and implementation of these algorithms in the form of a software system for a quaternary number system. According to the theory of digital signal processing, the same signal can be decomposed into different basic function systems (SBF). Different physical and practical implementations correspond to different spectral analysis. Complete and orthogonal SBFs are innumerable.*

In this paper we consider classical systems of Walsh basis functions. A mathematical description of new Walsh-like functions is given, an investigation is made of their basic properties, methods of constructing various orthogonal basis systems on their basis. More rapid methods of obtaining systems based on reordering functions are described. Schemes of algorithms for fragments of the software system are given that allow to automate the process of obtaining systems represented in the form of square matrices of any dimension. The results of the work are of an original nature and are applicable for further research in the field of spectral processing of digital discrete signals.

***Keywords:** digital signal processing, Walsh functions, Hadamard ordering, Paley ordering, Harmut ordering, discrete signals, spectral analysis*

Введение

С развитием цифровых технологий все более актуальным становится вопрос обеспечения быстрых преобразований данных. Это относится и к такой востребованной области как цифровая обработка сигналов. При использовании спектральных методов вычислительная и функциональная эффективность решения многих задач цифровой обработки зависит от используемых систем базисных функций [1]. Выбор рационального базиса является сложной теоретической и прикладной задачей, так как количество существующих ортогональных систем базисных функций достаточно велико.

Особого внимания заслуживают двоично-ортогональные системы базисных функций, которые включают в себя системы функций меандрового типа Радемахера, Уолша и Хаара, интервал ортогональности которых представляется совокупностью двоично-рационального числа равных подинтервалов [1]. Эти системы очень важны для практики спектральной обработки, так как функции Радемахера и Уолша принимают значения ± 1 , либо ± 1 и 0 в случае функций Хаара. Данные функции связаны друг с другом, каждую из них можно получить из другой, создавая надлежащую линейную комбинацию. Благодаря данной особенности функции этих базисных систем легко реализуются с помощью средств вычислительной техники – как аппаратно, так и программно и их использование позволяет свести время машинной обработки к минимуму за счет исключения операции умножения и наличия эффективных алгоритмов быстрых преобразований [2].

Из функций этого класса особенно широко востребованы функции Уолша, допускающие различные способы упорядочивания, приводящие к трем базисным системам Пэли, Адамара и Хармута. Однако на практике встречаются задачи, решение которых требует других систем на базе функций Уолша. Разработка таких Уолше-подобных функций и систем не только увеличивает ассортимент эффективных базисов, что само по себе имеет важное значение, но еще больше расширяет область их применения в теории и практике цифровой спектральной обработки.

На возможность построения новых базисных систем за счет переупорядочивания функций в известных системах указывал еще А.М. Трахтман, не предлагая, однако, способа этого упорядочивания [2]. Поскольку основу существующих базисов Уолша составляет двоичная система счисления, то для синтеза новых систем Уолше-подобных функций целесообразно использовать другие системы счисления, в частности, четверичную систему счисления.

1. Классические функции Уолша

Функции Уолша представляют собой полную систему ортогональных, ортонормированных функций $\{wal(k, z)\}, k = 0, 1, \dots$, в которой возможно разложение произвольной математической функции с интегрируемым квадратом в ряд Фурье-Уолша. Число функций, включаемых в систему, обычно равно числу отсчетов каждой функции, так как при дискретном спектральном анализе с N отсчетами число спектральных составляющих также должно быть равно N [3].

$$z = \frac{i}{N}; N = 2^n; n = 1, 2, \dots; i = 0, 1, \dots, N - 1$$

Функции Уолша можно получить как произведение функций Радемахера, номера которых соответствуют коду Грея номера функции Уолша.

Первые 8 функций Уолша приведены на рисунке 1.

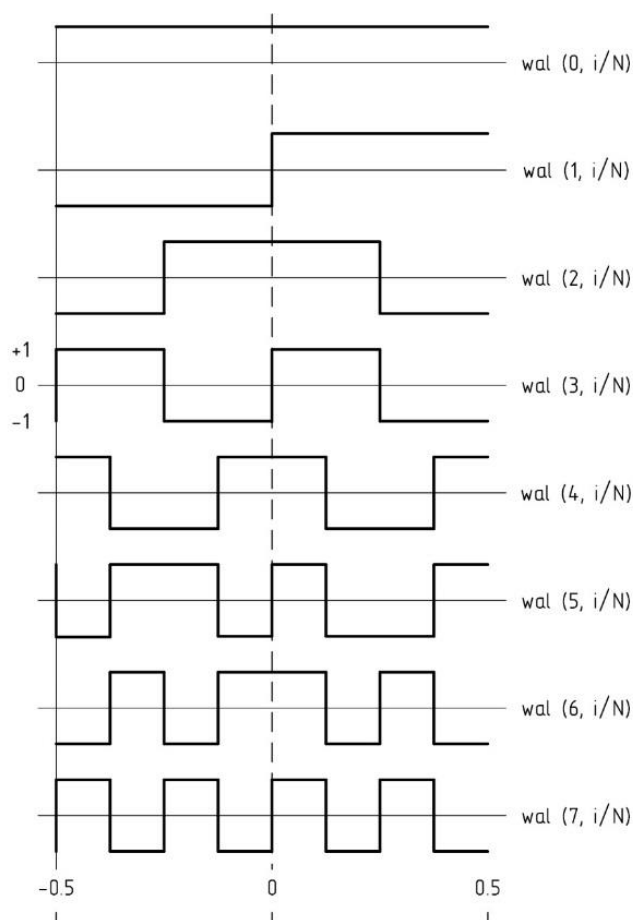


Рис. 1. Функции Уолша

Различные способы упорядочения функций Уолша приводят к различным базисным системам. Наиболее применяемыми на практике являются упорядочения Адамара, Пэли и Хармута [3]. Наиболее удобно представлять эти системы в виде квадратных матриц, каждая строка которых соответствует своей функции Уолша. Рассмотрим более подробно каждую из систем в аналитической и матричной формах.

Система Уолша-Адамара $\{had_2(k, i)\}$. Система Адамара для дискретных сигналов, определенных на конечном интервале, аналитически определяется следующим образом:

$$had_2(k, i) = \prod_{m=1}^n [r(m, i / N)]^{k_{n+1-m}} = (-1)^{\sum_{m=1}^n k_m i_m}, \quad (1)$$

где k – порядковый номер функции Уолша (для двоичной системы счисления k_m принимает значения 0 или 1), i – номер подинтервала, в двоичной системе счисления принимает значения 0 или 1, m – номер двоичного разряда, $r\left(m, \frac{i}{N}\right)$ – номер функции Радемахера.

В этой системе функции Уолша располагаются таким образом, что из них образуются матрица Адамара (рисунок 2).

Согласно упорядочению по Адамару квадратную матрицу H_{2^n} размера $2^n \times 2^n$ можно получить двумя способами:

1) Используя рекуррентное соотношение:

$$H_{2^n} = H_{2^{n-1}} \times H_2,$$

где \times – знак кронекеровского умножения.

2) Результат кронекеровского возведения в степень элементарной матрицы H_2 :

$$H_{2^n} = H_2^{[n]}.$$

Элементарной матрицей H_2 является:

$$H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

В случае $N = 4$ она будет иметь ту же структуру, но ее элементами являются элементарные матрицы $\pm H_2$:

$$H_4 = \begin{bmatrix} H_2 & H_2 \\ H_2 & -H_2 \end{bmatrix}.$$

Число элементов в строке при переходе от одной матрицы к другой удваивается.

Для $N = 2^n$ имеем:

$$H_{2^n} = \begin{bmatrix} H_{2^{n-1}} & H_{2^{n-1}} \\ H_{2^{n-1}} & -H_{2^{n-1}} \end{bmatrix}.$$

Для функций, принадлежащих системе Уолша-Адамара, будем использовать обозначения $had\left(k, \frac{i}{N}\right)$, где k – номер строки, i – номер столбца матрицы Адамара ($k, i = 0, 1, 2, \dots, 2^n - 1$).

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	-1	1	-1	1	-1	1	-1	1	-1	1	-1	1	-1	1	-1
2	1	1	-1	-1	1	1	-1	-1	1	1	-1	-1	1	1	-1	-1
3	1	-1	-1	1	1	-1	-1	1	1	-1	-1	1	1	-1	-1	1
4	1	1	1	1	-1	-1	-1	-1	1	1	1	1	-1	-1	-1	-1
5	1	-1	1	-1	-1	1	-1	1	1	-1	1	-1	-1	1	-1	1
6	1	1	-1	-1	-1	-1	1	1	1	1	-1	-1	-1	-1	1	1
7	1	-1	-1	1	-1	1	1	-1	1	-1	-1	1	-1	1	1	-1
8	1	1	1	1	1	1	1	1	-1	-1	-1	-1	-1	-1	-1	-1
9	1	-1	1	-1	1	-1	1	-1	-1	1	-1	1	-1	1	-1	1
10	1	1	-1	-1	1	1	-1	-1	-1	-1	1	1	-1	-1	1	1
11	1	-1	-1	1	1	-1	-1	1	-1	1	1	-1	-1	1	1	-1
12	1	1	1	1	-1	-1	-1	-1	-1	-1	-1	-1	1	1	1	1
13	1	-1	1	-1	-1	1	-1	1	-1	1	-1	1	1	-1	1	-1
14	1	1	-1	-1	-1	-1	1	1	-1	-1	1	1	1	1	-1	-1
15	1	-1	-1	1	-1	1	1	-1	-1	1	1	-1	1	-1	-1	1

Рис. 2. Матрица Адамара в классическом базисе Уолша для $N = 16$

Следует отметить, что в матрице Адамара любая строка, за исключением первой, содержит равное число знаков «+» и «-». Кроме того, матрица является симметричной, что является причиной равноправия переменных k и i .

Система Уолша-Пэли $\{pal_2(k, i)\}$. Функции Уолша образуют систему, упорядоченную по Пэли, в которой каждая k -ая функция есть произведение функции Радемахера с номерами, соответствующим номерам разрядом двоичного представления k , содержащих 1.

Система Пэли аналитически выражается следующей формулой:

$$pal_2(k, i) = \prod_{m=1}^n [r(m, i / N)]^{k_m} = (-1)^{\sum_{m=1}^n k_{n+1-m} i_m}. \quad (2)$$

В формуле (2) сумма произведений разрядов k и i не меняется при замене m на $n+1-m$, поэтому справедлива другая запись функции Уолша-Пэли:

$$pal_2(k, i) = (-1)^{\sum_{m=1}^n k_m i_{n+1-m}}.$$

Данная система может быть получена из системы Уолша-Адамара двоичной инверсией номеров функций, т.е. путем записи разрядов двоичного представления номера в обратном порядке, поэтому для данного и последующего способа упорядочений примеры матриц приводится не будут. Матрица Уолша-Пэли также является симметричной и отличается от матрицы для системы Адамара только расположением строк. Данную матрицу при произвольном $N = 2^n$ можно построить с помощью следующего мнемонического правила. Для $N = 2$ элементарная матрица имеет один и тот же вид для любого способа упорядочивания:

$$H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

Для $N = 4, 8, 16$ и т.д. необходимо каждую строку матрицы продублировать дважды, а затем к первой из них справа приписываются те же элементы, а ко второй — противоположные. Упорядочение функций в системе Уолша-Пэли называется диадическим, а в системе Уолша-Адамара естественным.

Система Уолша-Хармута $\{har_2(k, i)\}$. В данной системе функции располагаются в порядке увеличения числа знакоперемен на интервале определения. Данная система в основном повторяет систему, введенную самим Уолшем, и ее можно получить из системы Уолша-Пэли. Номер функции Уолша для системы Пэли равен коду Грея двоичного кода текущего номера функции Уолша в системе Хармута. Код Грея образуется последовательным суммированием по модулю два соседних разрядов двоичного разложения, начиная с младшего: $\langle k_m \rangle = k_m + k_{m+1} \pmod{2}$, $m = 1, 2, \dots, n$, $k_{n+1} = 0$, где сложению по модулю два соответствует следующим правилам: $0 + 0 = 0$, $1 + 0 = 1$, $0 + 1 = 1$, $1 + 1 = 0$.

Система Хармута аналитически записывается в виде:

$$har_2(k, i) = \prod_{m=1}^n [r(m, i / N)]^{<k_m>} = (-1)^{\sum_{m=1}^n <k_m> i_m} . \quad (3)$$

Очевидно, что все три системы связаны друг с другом, потому что состоят из одних и тех же функций Уолша. Данную взаимосвязь наглядно демонстрирует граф, изображенный на рисунке 3.

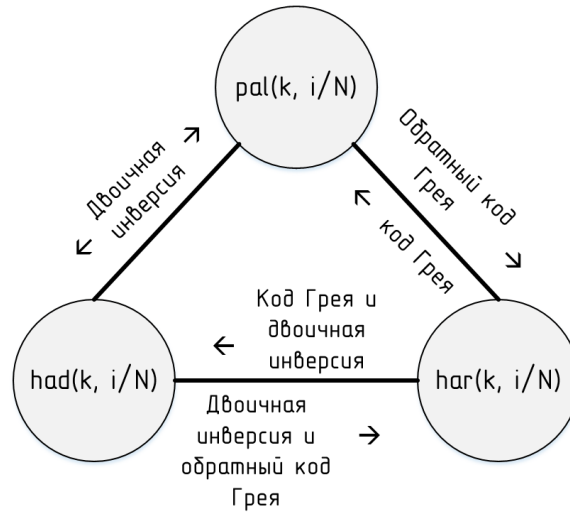


Рис. 3. Взаимосвязь номеров функций в различных системах функций Уолша

Взаимосвязь между номерами функций систем Адамара, Пэли и Хармута при конечном N отражает таблица 1.

Таблица 1.

Таблица переходов между системами

N = 4			N = 8			N = 16		
pal	had	har	pal	had	har	pal	had	har
0=00	00=0	00=0	0=000	000=0	000=0	0=0000	0000=0	0000=0
1=01	10=2	01=1	1=001	100=4	001=1	1=0001	1000=8	0001=1
2=10	01=1	11=3	2=010	010=2	011=3	2=0010	0100=4	0011=3
3=11	11=3	10=2	3=011	110=6	010=2	3=0011	1100=12	0010=2
			4=100	001=1	110=6	4=0100	0010=2	0110=6
			5=101	101=5	111=7	5=0101	1010=10	0111=7
			6=110	011=3	101=5	6=0110	0110=6	0101=5
			7=111	111=7	100=4	7=0111	1110=14	0100=4
						8=1000	0001=1	1100=12
						9=1001	1001=9	1101=13
						10=1010	0101=5	1111=15
						11=1011	1101=13	1110=14
						12=1100	0011=3	1010=10
						13=1101	1011=11	1011=11
						14=1110	0111=7	1001=9
						15=1111	1111=15	1000=8

2. Аналитическое описание функций Уолша и их свойства

Рассматриваемый в данной работе Уолше-подобный базис по свойствам близок к свойствам базисов комплексных экспоненциальных функций Виленкина-Крестенсона (ВКФ), являющимися в свою очередь обобщением функций Уолша. Основой математического аппарата нового базиса является процедура Хартли, использованная в качестве вещественной альтернативы комплексным экспоненциальным функциям Фурье и представляющая собой обычные тригонометрические функции. Данная процедура, распространённая на обобщенные тригонометрические функции, позволяет образовывать комплексные и вещественные части ВКФ, так как комплексный характер данных функций требует трудоемкой арифметики, что является существенным ограничением при программной или аппаратной реализации.

Пусть 4 есть целое положительное число, равное основанию системы счисления, а целое число k есть номер, а i – аргумент обобщенных тригонометрических функций $\cos\left(\frac{\pi}{2} \sum_{m=1}^n k_m i_m\right)$ и $\sin\left(\frac{\pi}{2} \sum_{m=1}^n k_m i_m\right)$, используемых в базисе ВКФ. На интервале из $N = 4^n$ точек аргументы имеют позиционные n -разрядные представления

$$k = \sum_{m=1}^n k_m 4^{m-1}, \quad i = \sum_{m=1}^n i_m 4^{m-1}, \quad (4)$$

где k_m и i_m являются m -ми разрядами этих представлений и лежат в диапазоне $[0, 3]$.

Из данных тригонометрических функций можно образовать следующие дискретные функции:

$$Cas(k, i) = \cos\left(\frac{\pi}{2} \sum_{m=1}^n k_m i_m\right) + \sin\left(\frac{\pi}{2} \sum_{m=1}^n k_m i_m\right). \quad (5)$$

Данные функции являются обобщением известных функций Хартли на систему счисления с произвольным основанием. Из развернутой записи (5) можно получить полезное более сжатое их представление, используя известные тригонометрические соотношения

$$Cas(k, i) = \sqrt{2} \sin\left(\frac{\pi}{2} \sum_{m=1}^n k_m i_m + \frac{\pi}{4}\right) = \sqrt{2} \cos\left(\frac{\pi}{2} \sum_{m=1}^n k_m i_m - \frac{\pi}{4}\right). \quad (6)$$

Записанные таким образом Уолше-подобные функции (УПФ) имеют ряд интересных свойств [4].

Данные свойства позволяют сделать вывод о возможности построения систем на базе данных функций.

Свойства функций Уолша. Перечислим основные свойства, которыми обладают Уолше-подобные функции. Данные свойства будут приведены без доказательств [4].

1. УПФ являются действительными функциями, принимающими только 2 различных значений. Выражения (5) и (6) являются подтверждением данного высказывания.

2. В УПФ переменные k и i являются равноправными, т.е. значение функции не изменится, если их поменять местами. Данное утверждение математически выражается следующим образом:

$$Cas(k,i) = Cas(i,k)$$

3. УПФ есть периодические функции, период которых равен $N = 4^n, n = 1, 2, \dots$
 4. Среднее значение любой УПФ, кроме нулевой, равно нулю, т.е.

$$\frac{1}{N} \sum_{i=0}^{N-1} Cas(k,i) = 0, \quad k \neq 0. \quad (7)$$

5. Мощность любой k -й УПФ равна единице:

$$P_k = \frac{1}{N} \sum_{i=0}^{N-1} [Cas(k,i)]^2 = 1. \quad (8)$$

6. УПФ являются ортогональными функциями, т.е.

$$\frac{1}{N} \sum_{i=0}^{N-1} Cas(k,i)Cas(\lambda,i) = 0, \quad k \neq \lambda. \quad (9)$$

3. Системы вещественных базисных функций Уолша

Системы дискретных УПФ удобно записывать в виде матриц C их значений. Эти матрицы будут симметрическими и ортогональными. Обратные матрицы будут совпадать с прямыми с точностью до постоянного множителя $1/N$: $C^{-1} = C/N$. Свойства матриц УПФ одинаковы для строк и столбцов (в силу их симметричности). Матрицы содержат ровно 2 различных действительных элементов. Элементы нулевых строк и столбцов равны единице.

Для полученной системы с основанием 4 возможны матрицы УПФ, отличающиеся порядком следования строк и столбцов, т.е., для данной системы возможны различные способы упорядочивания УПФ, которые образуют новые СБФ, что позволяет значительно расширить ассортимент существующих тригонометрических базисов.

Базисная система УПФ, описываемая выражениями (5) и (6), отличается тем, что структура ее матрицы имеет блочный характер. Подобным свойством обладает классическая матрица Уолша для упорядочения Адамара. Рассмотрим более подробно три вида упорядочения для системы УПФ.

3.1. Аналитическое представление систем

Система УПФ-Адамара $\{had_4(k,i)\}$. Система УПФ-Адамара строится по аналогии с системой Уолша-Адамара, главным отличием является задаваемое аналитическое выражение.

При конечной разрядности n и фиксированных значениях N справедлива следующая запись этой системы в функции нормированного аргумента i :

$$had_4(k, i) = \cos\left(\frac{\pi}{2} \sum_{m=1}^n k_m i_m\right) + \sin\left(\frac{\pi}{2} \sum_{m=1}^n k_m i_m\right). \quad (10)$$

Справедлива также запись следующего вида:

$$had_4(k, i) = \sqrt{2} \sin\left(\frac{\pi}{2} \sum_{m=1}^n k_m i_m + \frac{\pi}{4}\right) = \sqrt{2} \cos\left(\frac{\pi}{2} \sum_{m=1}^n k_m i_m - \frac{\pi}{4}\right). \quad (11)$$

Система УПФ-Пэли $\{pal_4(k, i)\}$. Система УПФ-Пэли получается из системы УПФ-Адамара путем замены прямого кода k на инверсный.

$$pal_4(k, i) = \cos\left(\frac{\pi}{2} \sum_{m=1}^n k_{n+1-m} i_m\right) + \sin\left(\frac{\pi}{2} \sum_{m=1}^n k_{n+1-m} i_m\right). \quad (12)$$

Более сокращенная запись (12) имеет вид:

$$pal_4(k, i) = \sqrt{2} \sin\left(\frac{\pi}{2} \sum_{m=1}^n k_{n+1-m} i_m + \frac{\pi}{4}\right) = \sqrt{2} \cos\left(\frac{\pi}{2} \sum_{m=1}^n k_{n+1-m} i_m - \frac{\pi}{4}\right). \quad (13)$$

Система УПФ-Хармута $\{har_4(k, i)\}$. Система УПФ-Пэли получается из системы УПФ-Адамара путем замены прямого кода k на код Грея.

$$har_4(k, i) = \cos\left(\frac{\pi}{2} \sum_{m=1}^n \langle k_m \rangle i_m\right) + \sin\left(\frac{\pi}{2} \sum_{m=1}^n \langle k_m \rangle i_m\right). \quad (14)$$

Или сжатая форма представления (14):

$$har_4(k, i) = \sqrt{2} \sin\left(\frac{\pi}{2} \sum_{m=1}^n \langle k_m \rangle i_m + \frac{\pi}{4}\right) = \sqrt{2} \cos\left(\frac{\pi}{2} \sum_{m=1}^n \langle k_m \rangle i_m - \frac{\pi}{4}\right), \quad (15)$$

где разряды кода Грея $\langle k_m \rangle$ вычисляются по формуле $\langle k_m \rangle = k_m + k_{m+1} \pmod{4}$

Следует также отметить, что все ранее приведенные свойства УПФ будут справедливы для всех перечисленных базисных систем.

3.2. Матричное представление систем

Прежде чем приступить к матричному описанию систем, полученных в пункте 3.1, следует рассмотреть таблицу переходов из системы счисления с основанием 10 в систему счисления с основанием 4. Взаимосвязь между системами счисления отражает таблица 2.

Таблица 2.

Таблица перехода из десятичной системы счисления в четверичную систему счисления

i_{10}	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
i_4	00	01	02	03	10	11	12	13	20	21	22	23	30	31	32	33

Стоит также отметить, что для нового базиса также возможно совершать переход от одной системы к другой, используя граф перехода, проиллюстрированный на рисунке 1(2).

Матричное представление системы УПФ-Адамара. В пункте 3.1 было получено аналитическое описание данной системы (10,11). Матричное представление для системы УПФ-Адамара при N=16 отражает рисунок 4.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	-1	-1	1	1	-1	-1	1	1	-1	-1	1	1	-1	-1
2	1	-1	1	-1	1	-1	1	-1	1	-1	1	-1	1	-1	1	-1
3	1	-1	-1	1	1	-1	-1	1	1	-1	-1	1	1	-1	-1	1
4	1	1	1	1	1	1	1	1	-1	-1	-1	-1	-1	-1	-1	-1
5	1	1	-1	-1	1	-1	-1	1	-1	-1	1	1	-1	1	1	-1
6	1	-1	1	-1	1	-1	1	-1	-1	1	-1	1	-1	1	-1	1
7	1	-1	-1	1	1	1	-1	-1	-1	1	1	-1	-1	-1	1	1
8	1	1	1	1	-1	-1	-1	-1	1	1	1	1	-1	-1	-1	-1
9	1	1	-1	-1	-1	-1	1	1	1	1	-1	-1	-1	-1	1	1
10	1	-1	1	-1	-1	1	-1	1	1	-1	1	-1	-1	1	-1	1
11	1	-1	-1	1	-1	1	1	-1	1	-1	-1	1	-1	1	1	-1
12	1	1	1	1	-1	-1	-1	-1	-1	-1	-1	-1	1	1	1	1
13	1	1	-1	-1	-1	1	1	-1	-1	-1	1	1	1	-1	-1	1
14	1	-1	1	-1	-1	1	-1	1	-1	1	-1	1	1	-1	1	-1
15	1	-1	-1	1	-1	-1	1	1	-1	1	1	-1	1	1	-1	-1

Рис. 4. Матрица Адамара в базисе УПФ для N = 16

Матричное представление системы УПФ-Пэли. Матричное представление системы УПФ-Пэли, имеющей аналитическую запись (12,13) имеет форму, представленную на рисунке 5.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	-1	-1	-1	-1	-1	-1	-1	-1
2	1	1	1	1	-1	-1	-1	-1	1	1	1	1	-1	-1	-1	-1
3	1	1	1	1	-1	-1	-1	-1	-1	-1	-1	-1	1	1	1	1
4	1	1	-1	-1	1	1	-1	-1	1	1	-1	-1	1	1	-1	-1
5	1	1	-1	-1	1	-1	-1	1	-1	-1	1	1	-1	1	1	-1
6	1	1	-1	-1	-1	-1	1	1	1	1	-1	-1	-1	-1	1	1
7	1	1	-1	-1	-1	1	1	-1	-1	-1	1	1	1	-1	-1	1
8	1	-1	1	-1	1	-1	1	-1	1	-1	1	-1	1	-1	1	-1
9	1	-1	1	-1	1	-1	1	-1	-1	1	-1	1	-1	1	-1	1
10	1	-1	1	-1	-1	1	-1	1	1	-1	1	-1	-1	1	-1	1
11	1	-1	1	-1	-1	1	-1	1	-1	1	-1	1	1	-1	1	-1
12	1	-1	-1	1	1	-1	-1	1	1	-1	-1	1	1	-1	-1	1
13	1	-1	-1	1	1	1	-1	-1	-1	1	1	-1	-1	-1	1	1
14	1	-1	-1	1	-1	1	1	-1	1	-1	-1	1	-1	1	1	-1
15	1	-1	-1	1	-1	-1	1	1	-1	1	1	-1	1	1	-1	-1

Рис. 5. Матрица Пэли в базисе УПФ для N = 16

Данную матрицу можно получить более быстрым способом, зная, что номер функции системы Пэли являются обратным кодом от номера функции системы Адамара. Таблица 3 демонстрирует эту связь.

Таблица 3.

Связь номеров функций системы Пэли и Адамара в новом базисе

Номер функции Пэли	00	01	02	03	10	11	12	13	20	21	22	23	30	31	32	33
Номер функции Адамара	00	10	20	30	01	11	21	31	02	12	22	32	03	13	23	33

Матричное представление системы УПФ-Хармута. Матричное представление системы, описываемой формулами (14,15) опустим, поскольку алгоритм получения данной системы ничем не отличается от того, что был использован в классическом базисе Уолша. Как и в классической системе Уолша, в данной системе все три системы взаимосвязаны друг с другом. Поэтому при конечном N их можно построить, используя одну систему (например, Пэли) в качестве базовой. Так, например, номер функции Пэли в новой системе есть код Грея от номера функции Хармута (таблица 4).

Таблица 4.

Связь номеров функций системы Пэли и Хармута в новом базисе

Номер функции в системе Хармута		Номер функции в системе Пэли	
Номер функции Хармута (10 СС)	Представление в четверичной системе счисления	Код Грея в четверичной системе счисления	Номер функции Пэли (10 СС)
0	000	00	0
1	001	01	1
2	002	02	2
3	003	03	3
4	010	11	5
5	011	12	6
6	012	13	7
7	013	10	4
8	020	22	10
9	021	23	11
10	022	20	8
11	023	21	9
12	030	33	15
13	031	30	12
14	032	31	13
15	033	32	14

4. Программная реализация

Так как получение матриц, рассмотренных ранее, это достаточно долгий и монотонный процесс, то была написана система, в рамках которой возможно произвести генерацию любой матрицы, являющейся либо степенью 2 (классические функции Уолша), либо степенью 4 (Уолше-подобные функции).

За логику генерации таких матриц отвечают:

- функция вычисления суммы числового ряда,
- процедура генерации матриц.

Рассмотрим схемы алгоритма и дадим описание данным подпрограммам.

Функция вычисления суммы числового ряда. Функция предназначена для вычисления суммы числового ряда, являющегося показателем степени для основания 1 (в случае двоичной системы счисления), или аргументом тригонометрической функции (в случае четверичной системы счисления). Результатом выполнения функции является значение переменной SUM формата EXTENDED.

Данная функция в качестве входных параметров принимает следующие переменные:

- k - порядковый номер функции Уолша,
- i – номер участка знакопостоянства интервала определения функций Уолша (номер подинтервала),
- ss – система счисления, передается с формы,
- n – количество подинтервалов.

Схема алгоритма данной функции представлена на рисунке 6.

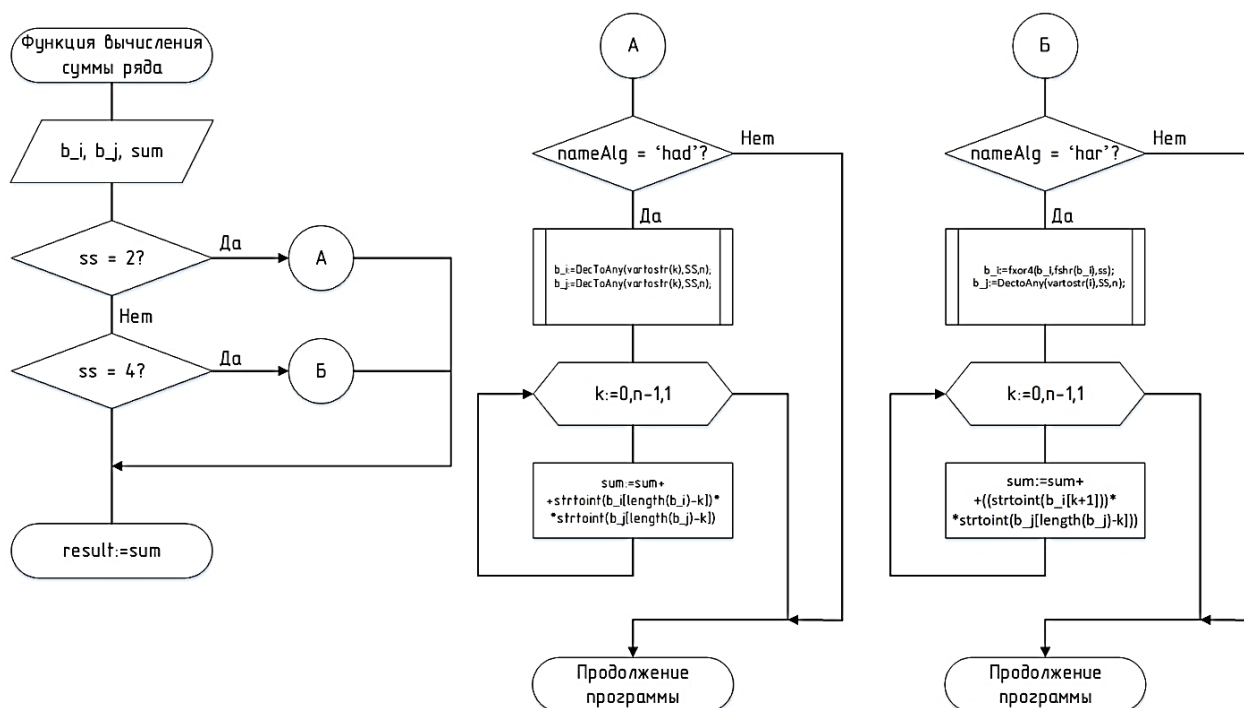


Рис. 6. Схема алгоритма функции вычисления суммы числового ряда

На вход функции поступают текущие значения переменных k , i , основание системы счисления и количество интервалов. Так как рассматриваемых систем счисления у нас две, был реализован оператор CASE, который производит операции в соответствии с системой счисления. Внутри каждого блока идет проверка на выбранный алгоритм (Пэли, Адамар, Хармут) и перевод данных переменных из 10 системы счисления (СС) в соответствующую СС (вызов функции DecToAny), далее начинается цикл, в ходе которого происходит вычисление суммы числового ряда. Для системы УПФ, упорядоченной по Хармуту, написана универсальная функция, позволяющая реализовать XOR в 4 СС. В результате выполнения функции мы имеем сумму числового ряда.

Процедура генерации матрицы. Процедура предназначена для генерации матриц в соответствии с выбранной системой счисления, размерностью и способом упорядочивания (Адамар, Пэли, Хармут)

Данная процедура работает со следующими переменными:

- n – SizeArray, отвечающий за размерность матрицы,
- i, j – строки и столбцы матрицы соответственно,
- $Ss1$ – SystemCC, отвечающий за систему счисления,
- nn - $\log_{ss1} n$,
- sum – результат выполнения суммы числового ряда,
- $k2$ – переменная, отвечающая за корень из двух,
- $DataArray$ – массив массивов.

Схема алгоритма представленной процедуры изображена на рисунке 7.

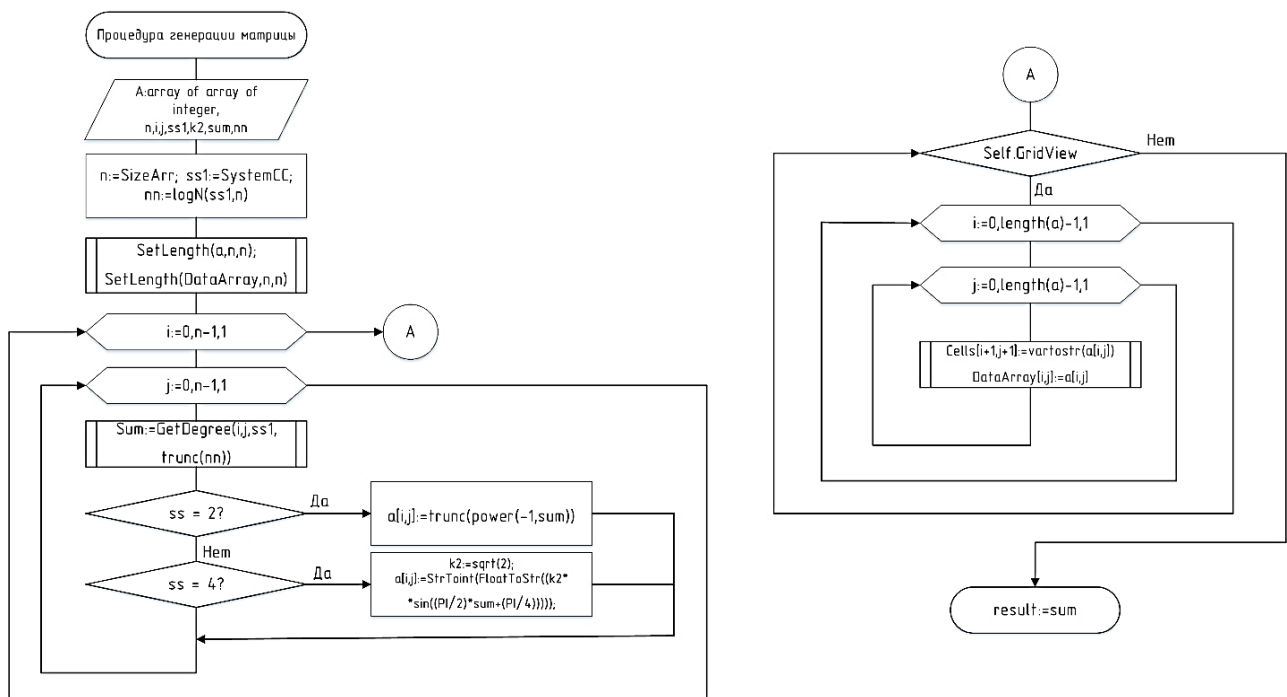


Рис. 7. Схема алгоритма процедуры генерации матрицы

Пример сгенерированной матрицы приведен на рисунке 8.

k,j	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
1	1	-1	-1	1	1	-1	-1	1	1	-1	-1	1	1	-1	-1	1	1	-1	-1	1	1	-1	-1	1	1	-1	-1
2	-1	1	-1	1	-1	1	-1	1	-1	1	-1	1	-1	1	-1	1	-1	1	-1	1	-1	1	-1	1	-1	1	-1
3	-1	-1	1	1	-1	-1	1	1	-1	-1	1	1	-1	-1	1	1	-1	-1	1	1	-1	-1	1	1	-1	-1	1
4	1	1	1	-1	-1	-1	-1	-1	-1	-1	1	1	1	1	1	1	1	1	1	-1	-1	-1	-1	-1	-1	-1	-1
5	-1	-1	1	1	-1	1	1	-1	1	1	-1	1	1	-1	-1	1	-1	-1	1	-1	-1	1	1	-1	1	1	-1
6	-1	1	-1	-1	1	-1	1	-1	1	-1	1	1	-1	1	-1	1	-1	1	-1	-1	1	-1	1	-1	1	-1	1
7	1	-1	-1	-1	1	1	-1	-1	1	1	1	-1	-1	1	1	1	-1	-1	-1	1	1	-1	-1	-1	-1	1	1
8	-1	-1	-1	1	1	1	1	-1	-1	-1	1	1	1	1	-1	-1	-1	-1	1	1	1	1	-1	-1	-1	-1	-1
9	-1	1	1	1	1	-1	-1	-1	1	1	1	1	-1	-1	1	1	1	1	1	1	1	-1	-1	-1	-1	1	1
10	1	-1	1	1	-1	1	-1	-1	1	-1	1	1	-1	1	-1	-1	1	-1	1	1	-1	1	-1	-1	1	-1	1
11	1	1	-1	1	-1	-1	1	-1	1	1	-1	1	-1	1	-1	1	1	-1	1	-1	1	-1	1	-1	1	1	-1
12	-1	-1	-1	-1	-1	-1	-1	1	1	1	1	1	1	1	1	-1	-1	-1	-1	-1	-1	-1	-1	-1	1	1	1
13	1	1	-1	-1	-1	1	1	1	-1	-1	1	1	1	-1	-1	-1	1	1	-1	-1	-1	1	1	1	-1	-1	1
14	1	-1	1	-1	1	-1	1	1	-1	1	-1	1	-1	1	-1	1	-1	1	-1	1	-1	1	-1	1	1	-1	1
15	-1	1	1	-1	1	1	-1	1	1	-1	-1	1	-1	1	-1	-1	1	1	-1	1	1	-1	1	1	-1	1	-1
16	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
17	-1	1	1	-1	-1	1	1	-1	-1	1	1	-1	1	1	-1	-1	1	1	-1	-1	1	1	-1	-1	1	1	-1
18	1	-1	1	-1	1	-1	1	-1	1	-1	1	-1	1	-1	1	-1	1	-1	1	-1	1	-1	1	-1	1	-1	1
19	1	1	-1	-1	1	1	-1	1	1	-1	-1	1	1	-1	-1	1	1	-1	1	1	-1	-1	1	1	-1	-1	1
20	-1	-1	-1	1	1	1	1	1	1	1	1	-1	-1	1	1	1	1	1	1	1	1	1	1	-1	-1	-1	-1

Рис. 8. Матрица системы УПФ-Адамара для N = 64

Размерность матрицы, для которой система выходила бы из строя, обнаружено не было. Примечательно, что матрица, размером 4096 на 4096, для четверичной системы счисления строится быстрее, чем для двоичной. Для четверичной системы счисления время обработки составило 71 сек., а для двоичной системы счисления данное время составило 84 сек.

Заключение

В данной статье была решена задача разработки новой базисной системы для четверичной системы счисления, получены необходимые алгоритмы для генерации матриц в новом базисе, упорядоченных по трем системам: Адамара, Пэли и Хармута. Получены и доказаны свойства для функций, на базе которых строятся новые базисные системы. А также приведены схемы алгоритмов, позволяющих автоматизировать процесс получения матриц.

Список литературы

1. Залманзон Л.А. Преобразования Фурье, Уолша, Хаара и их применение в управлении, связи и других областях. М.: Наука, 1989. 496 с.
2. Трахтман А.М. Введение в обобщенную спектральную теорию сигналов. М.: Сов.радио, 1972. 352 с.
3. Сюезв В.В. Основы теории цифровой обработки сигналов. Учебное пособие: - М.: Издательство «РТСофт», 2014. 752 с.
4. Сюезв В.В. Обобщенные функции и преобразования Хартли в системах счисления с постоянным основанием // Вестник Московского государственного технического университета им. Н.Э. Баумана. Серия «Приборостроение», 2014. №2. С. 63-79. Режим доступа: <http://vestnikprib.bmstu.ru/catalog/it/hidden/351.html> (Дата обращения: 01.05.18).

Анализ методов предобработки рентгеновских снимков

¹⁹Иванова Г.С., ²⁰Головков А.А., ²¹Лоншакова К.А.

МГТУ им. Н.Э. Баумана (НИУ), Москва, Россия

gsivanova@gmail.com

alexander.golovkov1@gmail.com

beautykika55@gmail.com

Аннотация: В статье рассмотрены методы предобработки маммологических рентгеновских снимков перед их использованием в качестве обучающей выборки нейронной сети и для последующего распознавания онкологии с помощью обученной нейронной сети. Реальные рентгеновские снимки имеют размер примерно 4000×5000 точек. Набор данных обучающей выборки, состоящий из изображений такого размера, даже при использовании современных средств вычислительной техники обработать в процессе обучения сети крайне сложно. Поэтому целесообразно выполнить сжатие изображений с минимально возможной потерей информации. Также следует повысить яркость/контрастность маммографических снимков, чтобы проявить признаки онкологии на фоне нормальных тканей груди, а также доброкачественных новообразований. В работе выполнено исследование существующих методов сжатия и увеличения яркости/контрастности изображений. Приведены результаты сравнительного анализа различных методов с точки зрения их использования при предобработке снимков обучающей выборки для распознавания онкологии с помощью нейросети и при работе уже с обученной сетью. В качестве критериев сравнения применены следующие характеристики: максимальная яркость пикселей, минимальная яркость пикселей, средняя яркость пикселей, среднеквадратическое отклонение и др.

Ключевые слова: маммография, рентгенография, предобработка данных, искусственные нейронные сети, яркость, контрастность, сжатие изображений.

Введение

Последнее время сильно возрос интерес к применению нейронных сетей для анализа информации в медицине [1, 2]. В том числе машинное обучение применяют для распознавания онкологических заболеваний по рентгеновским снимкам.

Одной из основных задач в процессе работы с искусственными нейронными сетями является предварительная обработка данных. В случае использования нейросетей

¹⁹Иванова Галина Сергеевна, д.т.н., профессор кафедры «Компьютерные системы и сети» МГТУ им. Н.Э. Баумана

²⁰Головков Александр Александрович, ассистент, аспирант кафедры «Компьютерные системы и сети» МГТУ им. Н.Э. Баумана

²¹Лоншакова Кристина Александровна, бакалавр кафедры «Компьютерные системы и сети» МГТУ им. Н.Э. Баумана

для классификации изображений предобработка заключается в качественных и количественных преобразованиях изображений, позволяющих «проявить» искомые объекты. С этой целью применяют: масштабирование, повышение яркости/контрастности, повышение резкости, поворот, отражение.

Исследуемый в работе набор данных представлен в виде базы медицинских файлов, содержащих снимки и документы обследованных пациентов, в формате DICOM (англ. Digital Imaging and Communications in Medicine) [3].

В ходе анализа предоставленных данных была выявлена необходимость:

- уменьшения размеров изображений, первоначальное разрешение которых составляло 5000x6000 пикселей, что существенно превышало возможности используемой техники в процессе обучения нейронной сети;
- преобразования качества изображений, так как исследуемая область на снимке имеет сложную, неоднородную структуру и включает большое количество плотных структур (железистая ткань, жировая ткань), которые затеяют опухоли и делают картину неразборчивой.

Следовательно, появилась необходимость выбора подходящих методов сжатия и повышения яркости/контрастности изображений.

1. Исследование алгоритмов сжатия изображений

Для анализа было отобрано несколько алгоритмов сжатия: избыточная выборка сглаживания или суперсемплинг (англ. Supersample anti-aliasing, SSAA) [4], бикубическая интерполяция сплайнами (англ. Bicubic spline interpolation) [5], свертка с фильтром Ланцоша (англ. Lanczos) [4], билинейная интерполяция (англ. Bilinear interpolation) [3] и метод ближайшего соседа (англ. Nearest neighbor) [4], которые, предположительно, могут обеспечить сжатие без существенной потери качества изображения.

Оценка результатов сжатия осуществлялась по следующим критериям:

- среднее максимальное значение яркости пикселей всего изображения — max (для области груди — max_area):

$$max = \frac{\sum_{j=1}^m \max_{i=1,n} (p_i)}{m},$$

где max_j — значение максимальной яркости j -ого снимка, m — количество снимков, n — количество пикселей, p_i — яркость i -ого пикселя (для max_area яркость пикселей фона не учитывается);

- среднее минимальное значение яркости пикселей всего изображения — min (для области груди min_area):

$$min = \frac{\sum_{j=1}^m \min_{i=1,n} (p_i)}{m},$$

где min_j — значение минимальной яркости j -ого снимка, m — количество снимков, n

— количество пикселей, p_i — яркость i -ого пикселя (для min_area яркость пикселей фона не учитывается);

- среднее значение средних значений яркости пикселей всего изображения — A (для области груди — A_area):

$$A = \frac{\sum_{j=1}^m I_j}{m}, \quad I_j = \frac{\sum_{i=1}^n p_i}{n},$$

где m — количество снимков, n — количество пикселей, p_i — яркость i -ого пикселя (для A_area яркость пикселей фона не учитывается);

- среднеквадратическое отклонение (СКО) средних значений яркости пикселей всего изображения — B (для области груди — B_area):

$$B = \sqrt{\frac{1}{m} \sum_{i=1}^m (A_i - A)^2},$$

где A — среднее значение средних значений яркости пикселей всего изображения, A_i — среднее значение яркости пикселей i -ого изображения, m — количество снимков (для B_area яркость пикселей фона не учитывается);

- среднее значение СКО всего изображения — C (для области груди — C_area):

$$C = \frac{\sum_{j=1}^m S_j}{m}, \quad S_j = \sqrt{\frac{1}{n} \sum_{i=1}^n (p_i - \bar{p})^2},$$

где m — количество снимков, n — количество пикселей, p_i — яркость i -ого пикселя, \bar{p} — средняя яркость снимка (для C_area яркость пикселей фона не учитывается);

- СКО среднеквадратических отклонений всего изображения — D (для области груди — D_area):

$$D = \sqrt{\frac{1}{m} \sum_{i=1}^m (S_i - \bar{S})^2},$$

где m — количество снимков, S_i — СКО i -ого снимка, \bar{S} — средняя СКО всех снимков (для D_area яркость пикселей фона не учитывается);

- среднее значение локальных яркостей (средняя яркость пикселей всех наложения ядра на снимок) — A_kernel :

$$A_kernel = \frac{\sum_{j=1}^m Q_j}{m}, \quad Q_j = \frac{\sum_{k=1}^l I_k}{l},$$

где m — количество снимков, l — количество наложений ядра на снимок, I_k — среднее значение в k -ом наложении ядра на снимок;

- СКО локальных яркостей — B_kernel :

$$B_kernel = \sqrt{\frac{1}{m} \sum_{j=1}^m (Q_j - A_kernel)^2},$$

где m — количество снимков, Q_j — среднее значение яркости для всех наложений ядра в j -ом снимке (см. выше), A_kernel — среднее значение локальных яркостей (см. выше);

- среднее значение СКО локальных яркостей — C_kernel :

$$C_kernel = \frac{\sum_{j=1}^m S_j}{m}, \quad S_j = \sqrt{\frac{1}{l} \sum_{k=1}^l (I_k - \bar{I})^2},$$

где m — количество снимков, l — количество наложений ядра на снимок, I_k — среднее значение в k -ом наложении ядра на снимок, \bar{I} — средняя яркость среди всех наложений ядра на снимок;

- СКО среднеквадратических отклонений локальных яркостей — D_kernel :

$$D_kernel = \sqrt{\frac{1}{m} \sum_{j=1}^m (S_j - C_kernel)^2},$$

где m — количество снимков, S_j — СКО локальной яркости j -ого снимка (см. выше), \bar{S} — средняя СКО всех локальных яркостей.

Результаты исследований представлены в таблице 1.

Таблица 1.

Оценка алгоритмов сжатия

Критерии	Исходные данные	Супер-сэмплинг		Бикубическая интерполяция сплайнами		Свертка с фильтром Ланцоша		Билинейная интерполяция		Метод ближайшего соседа	
		Результаты (S)	Разница (I-S)	Результаты (C)	Разница (I-C)	Результаты (L)	Разница (I-L)	Результаты (R)	Разница (I-R)	Результаты (N)	Разница (I-N)
<i>max</i>	4060	4058	2	4915	-855	5073	-1013	4059	1	4059	1
<i>min</i>	0	0	0	0	0	0	0	0	0	0	0
<i>A</i>	3350	3350	0	3350	-0,12	3350	-0,19	3350	-0,03	3349	0,83
<i>B</i>	135,5	135,5	0	135,5	-0,03	135,6	-0,04	135,5	-0,01	135,3	0,24
<i>C</i>	814,7	811,1	3,57	813,9	0,73	813,8	0,85	813,6	1,09	814,9	-0,23
<i>D</i>	47,18	47,48	-0,3	47,19	-0,01	47,2	-0,02	47,24	-0,06	47,48	-0,3
<i>max_area</i>	3849	3849	0	4915	-1066	5073	-1224	3849	0	3849	0
<i>min_area</i>	416,0	4	412	5	411	1	415	32	384	476	60
<i>A_area</i>	2636	2633	3,5	2637	-0,65	2635	0,67	2635	0,83	2635	0,64
<i>B_area</i>	231,0	232,6	-1,62	231,3	-0,32	231,9	-0,9	231,9	-0,92	230,8	0,22
<i>C_area</i>	583,6	583,5	0,15	585,3	-1,67	590,5	-6,83	581,8	1,84	583,8	-0,17
<i>D_area</i>	156,8	157,9	-1,1	156	0,78	154,2	2,5	157,7	-0,99	156,8	-0,04
<i>A_kernel</i>	3348	3347	0,95	3348	0,82	3347	0,74	3347	0,91	3346	1,78
<i>B_kernel</i>	137,0	136,7	0,33	136,7	0,31	136,7	0,29	136,7	0,32	136,4	0,63
<i>C_kernel</i>	789,9	790,4	-0,49	790,5	-0,54	790,5	-0,53	790,5	-0,57	791	-1,03
<i>D_kernel</i>	48,71	48,71	0	48,74	-0,03	48,74	-0,03	48,72	-0,01	48,82	-0,11

Цель сжатия в рассматриваемом случае — уменьшить размер, сохранив количество информации на снимке. Откуда следует, что хорошими следует считать методы, после применения которых:

- max , min , max_area , min_area , A , A_area , A_kernel , B , B_area , B_kernel — остались неизменными;
- C , C_area , C_kernel — возросли по сравнению с исходными;
- D , D_area , D_kernel — уменьшились по сравнению с исходными.

По полученным данным было выделено три наилучших метода: избыточная выборка сглаживания, свертка с фильтром Ланцоша и метод ближайшего соседа. Однозначно определить наилучший метод сжатия не получилось, так как для разных критериев оценки наилучшими становились различные алгоритмы. Так при оценке яркости всего снимка наивысшую оценку получил алгоритм избыточной выборки сглаживания, при оценке яркости области груди лучше всего себя показал метод свертки с фильтром Ланцоша, а при оценке локальной яркости — все три алгоритма. В ходе дальнейшего анализа результатов было выбрано два алгоритма сжатия: суперсемплинг и свертка с фильтром Ланцоша.

В дальнейшем перед запуском обучения нейросети планируется использование двух указанных методов, показавших лучшие результаты. Окончательный выбор метода будет выполнен по результатам обучения нейронной сети.

2. Исследование методов повышения контрастности изображений

Так как большую часть снимка занимает фон, гистограмма распределения яркостей маммограммы имеет ярко выраженный подъем в сторону темных уровней. Распределение яркости снимков в области груди также имеет ярко выраженный пик. Поэтому мелкие новообразования на темных участках плохо видимы, а сами изображения характеризуются низким контрастом.

Поэтому был выбран метод преобразования гистограмм [6], позволяющий увеличить диапазон распределения яркостей, что должно способствовать увеличению контрастности изображения. Выбранный метод позволяет получить желаемую форму гистограммы распределения яркостей, путем преобразования яркостей исходного изображения. Это позволит сократить энтропию изображения и увеличить среднее квадратичное отклонение.

Для повышения яркости/контрастности изображений рассмотрены такие методы гистограммных преобразований, как нормализация [6] и эквализация [6].

Для оценки выбранных методов использованы следующие критерии:

- максимальное значение яркости пикселей всего изображения — max ;
- минимальное значение яркости пикселей всего изображения — min ;
- энтропия — H ;

$$H = \sum_{i=0}^{n-1} p_i \log_b p_i,$$

где n — количество уровней серого (4096 для 12-битных медицинских снимков), p_i — вероятность того, что пиксель будет иметь i -ый уровень серого, b — база функции логарифма, если $b=2$, то возвращаемое значение измеряется в битах;

- среднее значение яркости пикселей всего изображения — *mean*:

$$mean = \frac{\sum_{i=1}^n p_i}{n},$$

где n — количество пикселей, p_i — яркость i -ого пикселя;

- среднеквадратическое отклонение яркости всего изображения — δ :

$$\delta = \sqrt{\frac{1}{n} \sum_{i=1}^n (p_i - \bar{p})^2},$$

где n — количество пикселей, p_i — яркость i -ого пикселя, \bar{p} — средняя яркость снимка.

Результаты исследований представлены в таблице 2.

Таблица 2.

Оценка алгоритмов повышения яркости/контрастности

Критерии	Исходное	Нормализация гистограммы	CLAHE (10, 8, 8)	CLAHE (50, 5, 5)	CLAHE (2, 8, 8)	Эквализация гистограммы
<i>max</i>	4064	4096	4090	4095	4075	4096
<i>min</i>	0	0	8	0	2	6
<i>mean</i>	3123,59	2105	2883,67	2636,4	3046,1	2109,15
<i>H</i>	23,95	23,71	23,91	23,83	23,94	23,71
δ	798,21	1238,79	962,1	1147,4	840,8	1237,02

Основными критериями оценки, как уже упоминалось ранее, является энтропия и среднеквадратическое отклонение, поэтому наилучшим является алгоритм, построенный по методу эквализации гистограммы — адаптивная эквализация гистограммы с ограничением (англ. *Contrast Limited Adaptive Histogram Equalization, CLAHE*). Работа алгоритма напрямую зависит от параметров: *clipLimit* — порог для ограничения контрастности, большее число соответствует большему контрасту, *tileGridSize* — размер сетки для выравнивания, определяющий количество разбиений исходного изображения.

При запуске программы обработки были исследованы три варианта значений параметров (см. таблицу 2):

- *clipLimit* = 10, *tileGridSize* = (8,8);
- *clipLimit* = 50, *tileGridSize* = (5,5);
- *clipLimit* = 2, *tileGridSize* = (8,8).

Лучшим из рассмотренных вариантов согласно значениям показателей СКО, энтропии и визуальной оценки, оказался последний.

Заключение

Таким образом, по результатам исследований было установлено, что для предобработки маммографических снимков перед обучением и работой нейросети наилучшими алгоритмами сжатия являются алгоритмы, построенные по методам избыточной выборки сглаживания и свертки с фильтром Ланцоша. Среди алгоритмов контрастирования был выделен алгоритм, реализующий метод адаптивной эквализации гистограммы с ограничением. Именно эти алгоритмы будут использоваться для предобработки данных перед их подачей в нейросеть в процессе обучения и перед диагностикой с применением обученной сети.

Список литературы

1. Молодченков А.И., Фраленко В.П., Хачумов В.М. Классификация степени тяжести заболевания на основе искусственных нейронных сетей // Вестник Российского университета дружбы народов. Серия: Математика, информатика, физика. 2014. № 2. С. 150-156.
2. Дунин В.О., Егоров В.А. Перспективы создания и развития информационной системы с элементами интеллектуального поиска, анализа и обработки биомедицинской информации // Прикаспийский журнал: управление и высокие технологии. 2013. № 3. С. 162-169.
3. Плотников А.В., Прилуцкий Д.А., Селищев С.В. Стандарт DICOM в компьютерных медицинских технологиях [Электронный ресурс] // Московский Институт Электронной Техники URL: <https://mks.ru/library/article/1997/dicom.html> (дата обращения: 11.04.2018).
4. Карпинский А. Ликбез: методы ресайза изображений [Электронный ресурс] // Хабрахабр. Режим доступа: <https://habrahabr.ru/post/243285/> (дата обращения: 11.04.2018).
5. Multivariate interpolation [Электронный ресурс] // Wikipedia. Режим доступа: https://en.wikipedia.org/wiki/Multivariate_interpolation (дата обращения: 11.04.2018).
6. Грузман И.С., Киричук В.С., Косых В.П., Перетягин Г.И., Спектор А.А. Цифровая обработка изображений в информационных системах // Учебное пособие. Новосибирск: М.: НГТУ. 2002. 352 с.

Система управления виртуальной моделью инвалидной коляски в 3D-тренаже при помощи интерфейса «мозг-компьютер»

²²Попов А.Ю., ²³Колотовкин М.И., ²⁴Куклина Н.И.

МГТУ им. Н.Э. Баумана (НИУ), Москва, Россия

alexpopov@bmstu.ru
maxim.kolotovkin@yandex.ru
akenoq@yandex.ru

Аннотация: В статье рассмотрена технология получения и передачи управляющих сигналов от интерфейса «мозг-компьютер» в 3D-тренажер управления инвалидной коляской для людей с параличом посредством web-сокетов. Пользователями системы могут быть люди с параличом, испытывающие трудности в управлении как обычными инвалидными колясками, так и инвалидными колясками с электрическим приводом и ручным манипулятором. Благодаря тренировкам в 3D-тренажере люди не будут выезжать на городские улицы без определенного уровня подготовки. Задачи, решаемые в процессе реализации системы: осуществление взаимодействия с предоставленным ЭЭГ ИНЭУМ им. И.С. Брука; осуществление взаимодействия с web-сокет сервером; разработка алгоритма функционирования и соответствующего программного кода. Система реализована с использованием одноплатного компьютера Raspberry Pi Model B и электроэнцефалографа. Программный код написан на языке Python 2.7 и JavaScript.

Ключевые слова: 3D, интерфейс «мозг-компьютер», электроэнцефалограф, web-сокет, управляющие сигналы, тренажер, протокол, сервер, клиент.

The system for managing the virtual model of a wheelchair in 3D-training using the interface "brain-computer"

Popov A.Yu., Kolotovkin M.I., Kuklina N.I.

Bauman Moscow State Technical University, Moscow, Russia

alexpopov@bmstu.ru
maxim.kolotovkin@yandex.ru
akenoq@yandex.ru

Abstract: In this article there is a technology of getting and sending controlling signals

²²Попов Алексей Юрьевич – канд. техн. наук, доцент кафедры “Компьютерные системы и сети” МГТУ им. Н.Э. Баумана, SPIN-код 7380-5537.

²³Колотовкин Максим Ильич – студент кафедры "Компьютерные системы и сети" МГТУ им. Н.Э. Баумана.

²⁴Куклина Нина Ильинична – студентка кафедры "Компьютерные системы и сети" МГТУ им. Н.Э. Баумана.

from interface "brain-computer" to 3D training device of controlling wheelchair for disabled people using technology of web sockets. Users of the system could be people having the paralysis, people with difficulties of controlling simple wheelchairs and wheelchairs with electric drive or manual manipulator. With the help of training using 3D training device people, need not to move to the dangerous streets of the cities with bad level of training. The goals of the process of creating the system are: create the connection with 3D training device, create the connection with the server of web-socket, create the algorithm of the program and programming code. The system was created by using single-board computer Raspberry Pi Model B and training device. The code of the program written using Python 2.7 and JavaScript programming languages.

Keywords: 3D, interface "brain-computer", electroencephalograph, web-socket, control signals, training, protocol, server, client.

Введение

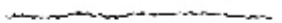




Интерфейс мозг-компьютер — это программно-аппаратный комплекс, позволяющий управлять непосредственно сигналами мозга внешними техническими системами, минуя традиционную для этого мышечную активность [1]. В последние годы ИМК активно разрабатывают для применения в нейрореабилитации. Одно из направлений разработки интерфейсов в этой сфере — так называемые вспомогательные интерфейсы, которые могут оказаться единственным каналом общения с окружающим миром для больных с выраженными двигательными нарушениями конечностей или мимической мускулатуры (при постинсультном или посттравматическом парезе, боковом амиотрофическом склерозе и др.). Пациенты с помощью интерфейса смогли бы управлять роботизированным протезом руки, инвалидной коляской, функциональной электростимуляцией, специфически активирующей парализованные мышцы, и прочими внешними техническими устройствами [2]. Работа ИМК основана на анализе электроэнцефалограммы головного мозга. Несмотря на то, что такой анализ позволяет получить лишь весьма общие данные о работе мозга, тем не менее существуют методы, позволяющие принимать решения на основе тех изменений, которые регистрирует электроэнцефалограф (ЭЭГ).

Целью разработки системы управления виртуальной моделью инвалидной коляски в 3D-тренаже при помощи интерфейса «мозг-компьютер» является реализация обучающего режима работы со вспомогательным интерфейсом для людей с ограниченными возможностями. Полученные от ЭЭГ им. И.С. Брука сигналы анализируются вычислительным узлом на базе одноплатного компьютера Raspberry Pi и позволяют управлять виртуальной моделью инвалидной коляски в режиме обучения, а в дальнейшем эти управляющие сигналы будут использоваться для реализации рабочего режима — управления электромоторами, установленными на инвалидной коляске. В данной работе особое внимание уделено проблеме обучения людей с ограниченными возможностями работе с ИМК, так как без должного уровня подготовки появляется

опасность не справиться с управлением в реальных условиях города. В работе рассмотрен вариант обучения в 3D-тренажере, так как такой подход позволит избежать затрат на оборудование тренировочных площадок в специализированных реабилитационных центрах, и даст возможность организовывать тренировки как в медицинских учреждениях, так и дома.

1. Анализ технологии получения и обработки сигналов с головного мозга

Основной целью применения ИМК является получение компьютерной системой однозначно интерпретируемых команд непосредственно от головного мозга без использования мышечной активности. Существуют различные типы ИМК: инвазивные или неинвазивные технологии, позволяющие преобразовывать некоторые нейрофизиологические сигналы в команды, адресованные внешнему техническому устройству или компьютеру. При реализации системы использовался неинвазивный интерфейс «мозг-компьютер» (ИМК), построенный на основе многоканальной электроэнцефалограммы головного мозга. В основе работы системы с неинвазивным ИМК лежит регистрация электрического потенциала на поверхности головы и его компьютерная обработка. В электроэнцефалограмме человека прослеживается определенная ритмическая активность, которая делится на несколько групп в зависимости от частоты волн (альфа-ритм, бета-ритм, гамма-ритм, дельта-ритм, мю-ритм). Для состояния бодрствования характерны бета- и мю-ритмы. Гамма- и дельта-ритмы появляются во время сна и при засыпании, альфа-ритм появляется при закрывании глаз. В связи с этими особенностями ритмов в ИМК чаще всего используются мю-, бета- и альфа-ритмы. Электроэнцефалограммы основных ритмов представлены на рисунке 1.

Форма ЭЭГ	Название	Частотные диапазоны волн	Состояние
	гамма	40-150 Гц	Максимум внимания, осознание, инсайт
	бета	14-40 Гц	Активное мышление, переживания, опасения
	альфа	8-14 Гц	Спокойный тихий ум, безмыслие в состоянии бодрствования
	тета	4-8 Гц	Сон со сновидениями
	дельта	1-4 Гц	Глубокий сон без сновидений

50 мкВ
1 секунда

Рис. 1. Основные ритмы ЭЭГ человека

С помощью компьютерной обработки из полученных сигналов выделяются компоненты, значимые для управления внешним устройством. Затем происходит преобразование этих компонентов и формирование команды, передаваемой на внешнее

устройство [2]. Так, например, в электроэнцефалограмме человека присутствует компонент P300, возникающий примерно на 300-400 мс с амплитудой 5-15 мкВ, в ответ на неожиданно и достаточно редко предъявляемый зрительный или иной стимул. Классический интерфейс на основе волны P300 – система для печати, которая представляет собой матрицу из букв, в которой последовательно подсвечиваются строки и столбцы. Пример того, как данный компонент проявляется на электроэнцефалограмме представлен на рис. 2. Именно этот компонент был взят за основу при проектировании интерфейса системы – перед пользователем на экране возникают возможные направления движения и регистрируются сигналы-реакции головного мозга на предъявляемые стимулы.

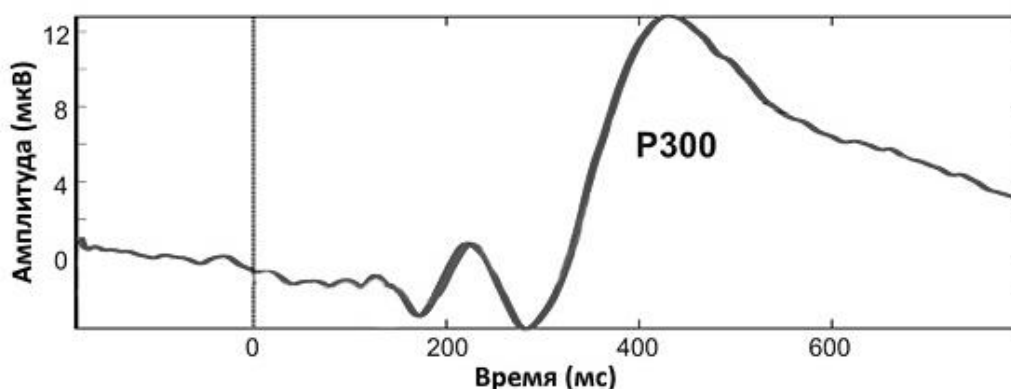


Рис. 2. Позитивный компонент P300

Для систематизации полученных от ЭЭГ данных используются различные методы ЭЭГ-анализа, наиболее универсальным на данный момент является метод нейросетевой классификации данных электроэнцефалограммы. Использование искусственных нейронных сетей позволяет решать сложно формализуемые задачи, в которых не представляется возможным принять во внимание все реально доступные условия и точно выписать алгоритм нахождения ответа. Учитывая сложность алгоритмизации работы при составлении клинического заключения электроэнцефалограммы, оптимальным является использование аппарата искусственных нейронных сетей, так как нейронные сети обучаются на примерах, не требуют описания строго формализованной задачи, способны к обобщению и извлечению существенных данных из избыточной информации [3, 4]. Также использование нейронных сетей обеспечивает гибкость при работе с различными компонентами электроэнцефалограммы. То есть, при изменении ключевого анализируемого компонента, на основе которого и будет происходить систематизация результатов, менять алгоритмы анализа данных не нужно.

При реализации данной системы использовался ЭЭГ ИНЭУМ им. И.С. Брука цифрового типа с сухими электродами, которые преобразуют электроэнцефалограмму в цифровую форму и вводят ее в микроконтроллер STM32, который управляет непрерывным процессом регистрации электроэнцефалограммы, одновременно

записываемой в память вычислительного узла. В качестве вычислительного узла для сбора данных от ЭЭГ, для предъявления возможного направления движения, для формирования и передачи управляющих сигналов в 3D-тренажер использовался одноплатный компьютер Raspberry Pi Model B. Обработка данных полученных от ЭЭГ построена на использовании языка программирования Python 2.7 и библиотеки PyBrain, позволяющей на основе принципов нейронных сетей, систематизировать результаты. Таким образом, в результате систематизации возможно однозначно определить выбранное пользователем направление движения и сформировать соответствующую управляющую команду, которую далее необходимо передать на управляемое техническое устройство или его модель.

2. Организация непрерывного сокет-соединения

В разрабатываемой системе управления виртуальной моделью инвалидной коляски в 3D-тренажере возникла необходимость реализовать непрерывное сокет-соединение между подсистемами 3D-тренажера и вычислительного узла, взаимодействующего с ЭЭГ. Схема взаимодействия представлена на рисунке 3.

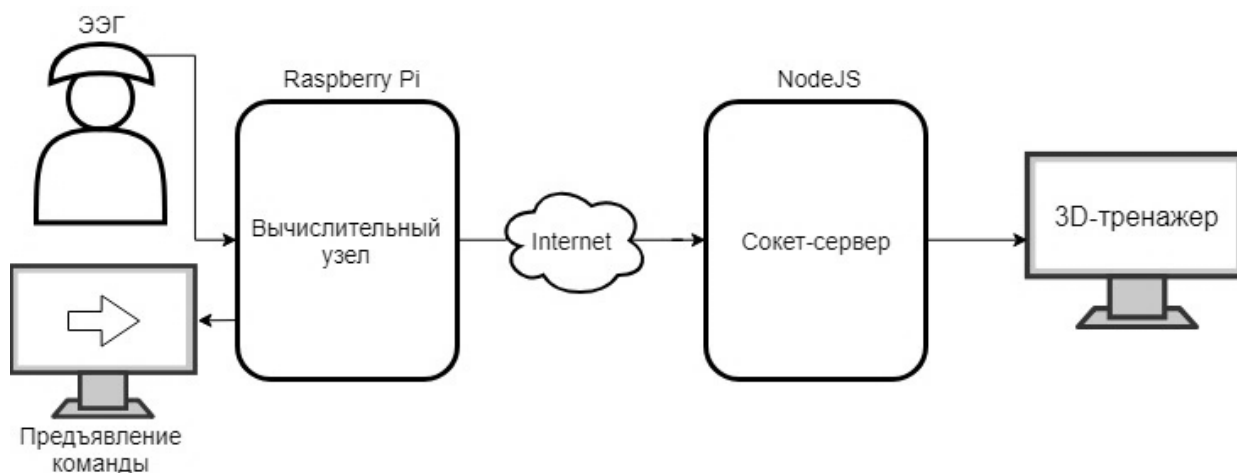


Рис. 3. Схема взаимодействия 3D-тренажера с ИМК

Сокетом называют программный интерфейс для обеспечения обмена данными между процессами, которые могут находиться как на одной ЭВМ, так и на ЭВМ, находящихся в одной сети. Для взаимодействия между машинами с помощью стека протоколов TCP/IP используются адреса и порты. Сетевой сокет в соединении является конечной точкой. Сокет может быть клиентским или в серверным – в зависимости от того, какую функцию он выполняет.

Для организации непрерывного соединения между клиентом и сервером не подходит HTTP-соединение в формате «запрос-ответ», так как в решаемой задаче особенно важна скорость обмена данными, чего не может обеспечить HTTP-протокол. При работе с протоколом HTTP отправку данных не может инициировать сервер, значит, клиентом затрачивается лишнее время на систематические опросы сервера с определенной частотой, а сервером - на ответы. Поэтому нужно использовать протокол

обмена данными по непрерывному соединению. Непрерывное соединение в сети можно организовать с использованием протокола полнодуплексной связи WebSocket с открытым «рукопожатием» на уровне TCP. WebSocket протокол работает поверх TCP-соединения, также, как и протокол HTTP – оба протокола расположены на седьмом уровне модели OSI и базируются на TCP-протоколе пятого уровня модели OSI [5]. Но в отличие от HTTP WebSocket-протокол предназначен для обмена сообщениями между клиентом и сервером в режиме реального времени – способствует распространению интерактивного содержимого. Хотя они отличаются друг от друга, WebSocket предназначен для работы через HTTP-порты 80 и 443, а также для поддержки HTTP-прокси и посредников, что делает его совместимым с протоколом HTTP. Чтобы установить соединение WebSocket клиент и сервер используют протокол, похожий на HTTP.

Пример запроса:

```
GET /demo HTTP/1.1
Upgrade: WebSocket
Connection: Upgrade
Host: example.com
Origin: http://example.com
WebSocket-Protocol: sample
```

На запрос сервер отвечает определенным образом, подтверждая переход на WebSocket-протокол:

```
HTTP/1.1 101 Web Socket Protocol Handshake
Upgrade: WebSocket
Connection: Upgrade
WebSocket-Origin: http://example.com
WebSocket-Location: ws://example.com/demo
WebSocket-Protocol: sample
```

Сразу после отправки ответа WebSocket-соединение считается установленным, клиент и сервер могут начинать двунаправленный обмен сообщениями поэтому же TCP-соединению. Для передачи текстового сообщения (в кодировке UTF-8) необходимо перед ним передать нулевой байт, а после — байт со значением 255 [6].

Таким образом, при реализации системы используется протокол WebSocket, посредством которого происходит взаимодействие клиентов сокет-сервера: вычислительного узла ИМК и 3D-тренажера.

3. Сокет-сервер

Для обмена информацией между клиентским приложением, отвечающим за приём управляющих сигналов от ЭЭГ, и клиентским приложением, отвечающим за

реализацию действий в 3D-тренажере и отображением трёхмерной графики на монитор, был создан сокет-сервер.

На сокет-сервере находится приложение, которое хранит в себе массив всех подключённых к серверу клиентских сокетов (абстрактных объектов, являющихся конечной точкой соединения).

Приложение, реализующее сокет-сервер, выполняет следующие действия:

- 1) при подключении нового клиента сохраняет соответствующий ему сокет в ассоциативный массив;
- 2) при получении сообщения от клиентского сокета обрабатывает полученное сообщение и пересылает его всем остальным клиентским сокетам в ассоциативном массиве;
- 3) при отключении клиентского сокета приложение удаляет его из ассоциативного массива.

В определённый момент времени сокет-сервер может управлять действиями пользователя только на одном тренировочном уровне. Во время тренировки пользователь может выполнять следующие основные действия:

- 1) двигаться вперёд;
- 2) двигаться назад;
- 3) двигаться вокруг оси, перпендикулярной плоскости земли в виртуальном мире, в левую сторону;
- 4) двигаться вокруг оси, перпендикулярной плоскости земли в виртуальном мире, в правую сторону.

Пользователь может выполнять в 3D-тренажере вышеописанные действия одновременно, например, двигаться вперёд и вращаться в левую сторону. Чтобы хранить на сервере информацию о том, какие действия выполняет пользователь в данный момент времени, были введены четыре глобальные переменные логического типа «w», «a», «s», «d». Переменная «w» отвечает за движение вперёд, переменная «s» отвечает за движение назад, переменная «a» отвечает за поворот налево, переменная «d» отвечает за движение направо. Если переменная «w» в текущий момент времени имеет значение, равное истине, то виртуальная модель инвалидной коляски в 3D-тренажере движется вперёд, в противном случае — не движется вперёд. Принцип работы с переменными «a», «s», «d» аналогичен.

4. 3D-тренажер

Для того, чтобы обеспечить пользователю возможность прохождения обучения управлению инвалидной коляской при помощи ИМК до выезда в городские условия, разработан 3D-тренажер. 3D-тренажер является кросс-браузерным и реализован с использованием языка программирования JavaScript и легковесной библиотеки ThreeJS, используемой для создания и отображения анимированной компьютерной 3D графики. Важно было реализовать 3D-тренажер, в котором человек с ограниченными

возможностями сможет тренироваться в управлении виртуальной моделью инвалидной коляски в условиях, приближенным к реальным. В виртуальной модели городского пространства, намного эффективнее и реалистичнее воспроизводить объекты и явления реальных городских условий, чем делать это в специализированных тренировочных помещениях. Данный тренажер позволяет проходить обучение без наличия каких-либо специально оборудованных помещений – достаточно наличия компьютера с выходом в Интернет и браузером.

Для уточнения требований к функционированию 3D-тренажера необходимо определить варианты его использования. Система взаимодействует с двумя видами действующих лиц: неавторизованными пользователями и авторизованными пользователями.

Авторизованному пользователю доступны следующие функции:

- создание нового уровня для тренировки;
- редактирование существующего уровня для тренировки;
- прохождение тренировки на выбранном уровне;
- сохранение результата тренировки на сервер;
- выход из аккаунта.

Неавторизованному пользователю доступны следующие функции:

- ввод логина;
- ввод адреса сокет-сервера;
- вход в аккаунт под своим логином.

Как только пользователь заходит в веб-приложение 3D-тренажера, перед ним появляется форма ввода, заполнение которой необходимо для входа в систему. Пользователю необходимо ввести адрес сокет-сервера, к которому в процессе игры будут подключаться сокет-клиенты: вычислительный узел ИМК и 3D-тренажер. Подключение необходимо для получения данных с сокет-сервера с целью реализации движения виртуальной модели инвалидной коляски в 3D-тренажере. Также пользователю необходимо ввести свой логин, чтобы результат прохождения тренировки был закреплен за соответствующим пользователем.

После ввода данных пользователем в текстовые поля ввода веб-приложение 3D-тренажера сохраняет введенные данные на стороне клиента в постоянное локальное хранилище браузера (localstorage). Переменные, находящиеся в локальном хранилище, привязаны к определённому домену. Это означает, что приложения, запущенные на других доменах, не будут иметь доступа к переменным, привязанным к 3D-тренажеру. После успешного сохранения пользовательских данных в хранилище веб-приложения, пользователь попадает в главное меню 3D-тренажера, где ему предлагается создание/редактирование карты уровня для тренировки, выбор уровня для тренировки и его прохождение, выход из системы.

4.1. Создание и редактирование уровней

При разработке 3D-тренажера вставала задача реализовать достаточно близкую к реальному миру виртуальную модель, поэтому было принято решение реализовать два вида препятствий на уровнях для тренировки. В редакторе уровня, представленном на рисунке 4, предусмотрены как статические препятствия - «стены», так и динамические - «машины». Для успешного прохождения тренировки пользователю необходимо, преодолевая препятствия, добраться от стартовой позиции до финишной.

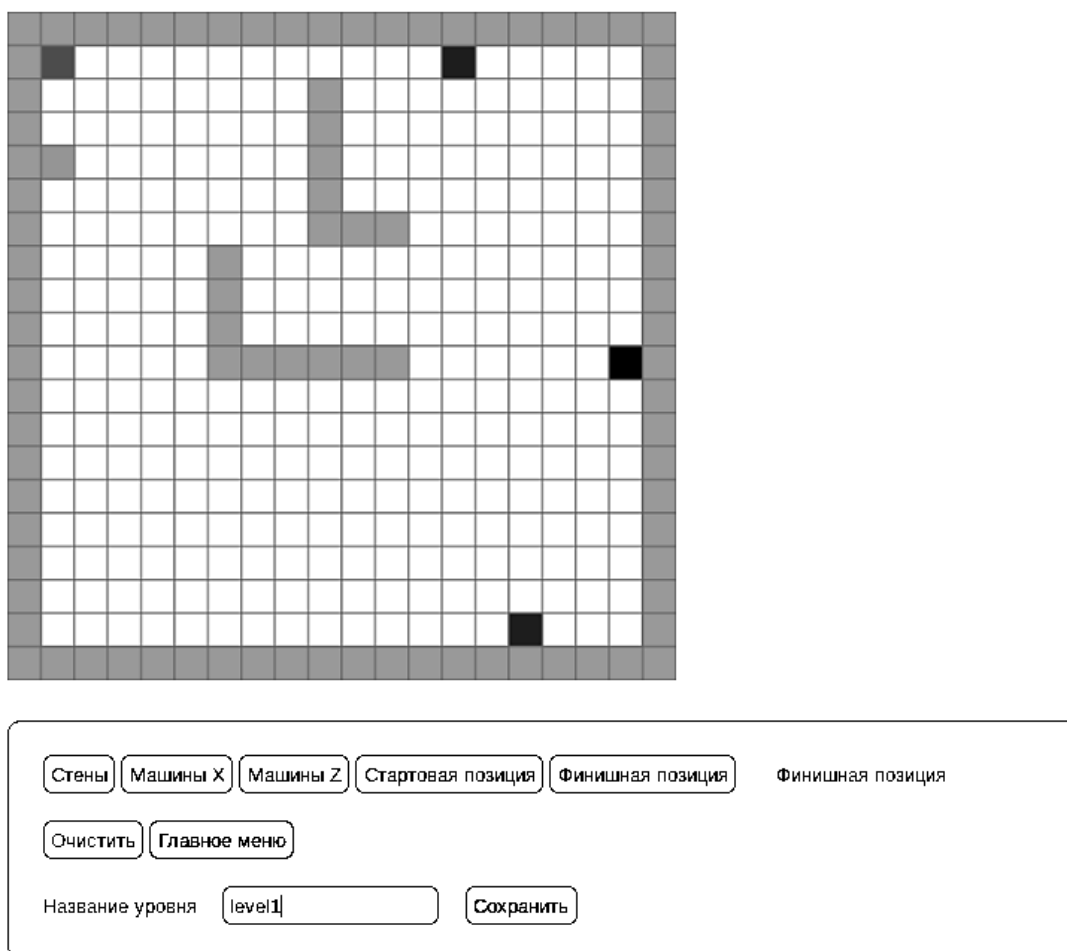


Рис. 4. Редактор уровня для тренировки

Уровень представляет собой матрицу, ширина и высота которой равняются двадцати ячейкам. У ячейки может быть один из шести типов, который определяет роль ячейки на карте тренировочного уровня. Типы задаются соответствующими целым числам в диапазоне от 1 до 6. Возможные типы ячеек:

- 1) Стена. Стеной является непроходимый участок, через который виртуальная модель инвалидной коляски не может пройти.
- 2) Машина, движущаяся по оси «X». При столкновении виртуальной модели инвалидной коляски с машиной тренировка завершается неудачей.
- 3) Машина, движущаяся по оси «Z». При столкновении виртуальной модели инвалидной коляски с машиной тренировка завершается неудачей.

- 4) Стартовая позиция. Со стартовой позиции начинается тренировка пользователя на данном уровне.
- 5) Финишная позиция. Для успешного прохождения тренировки пользователю необходимо добраться до финишной позиции, обойдя все препятствия.
- 6) Пустая клетка. Пустая клетка описывает проходимый участок, через который виртуальная модель инвалидной коляски может пройти.

Для редактирования типа ячейки необходимо в кнопочном меню щелкнуть на название типа нужного типа и после этого кликнуть на саму ячейку в матрице. После этого тип ячейки будет задан. Матрица, демонстрирующая структуру уровня, отображается в элементе веб-страницы, предназначенного для создания растрового двухмерного изображения. Для определения строки и столбца ячейки матрицы, по которой был сделан щелчок, используется следующий алгоритм: вначале определяется позиция курсора мыши относительно холста, после этого вычисляется номер строки и столбца выбранной ячейки путём деления положения курсора по определённой оси на размер клетки и последующего округления результата в меньшую сторону. После успешного изменения содержимого уровня результат сохраняется в постоянное локальное хранилище. Для сохранения уровня необходимо указать его имя в текстовом поле ввода и нажать на кнопку «Сохранить». Имя уровня может состоять только из строчных латинских букв и цифр.

4.2. Тренировка на уровне

Для реализации тренировки на определённом уровне 3D-тренажера использовался объектно-ориентированный подход к программированию на языке JavaScript, так как при использовании данного подхода имеется возможность реализовать определённый набор сущностей, отвечающих за действия внутри приложения. В объектно-ориентированном подходе основной категорией объектной модели является класс. Класс объединяет в себе на элементарном уровне как данные, так и операции, которые над ними выполняются (методы). Во время разработки структуры и отношений классов был выбран следующий принцип: будет иметься базовый класс, у которого будет создан только один экземпляр, и этот экземпляр класса будет управлять через интерфейсы работой других объектов.

Для работы с трёхмерной графикой было принято решение использовать легковесную кросс-браузерную библиотеку ThreeJS. Данная библиотека позволяет создавать ускоренную на GPU 3D-графику, используя язык JavaScript, без подключения плагинов для браузера. Это возможно благодаря использованию технологии WebGL. Режим тренировки 3D-тренажера представлен на рисунке 5.

Так как виртуальная модель инвалидной коляски в 3D-тренажере меняет свою позицию только по двум осям (в плоскости «поверхности земли»), были выведены следующие формулы для расчета позиции модели в двумерном пространстве:

$$P_x = s * \cos \alpha, \quad (1)$$

$$P_z = s * \sin \alpha, \quad (2)$$

$$x = \frac{W}{2} * \cos \alpha + P_x, \quad (3)$$

$$z = \frac{W}{2} * \sin \alpha + P_z, \quad (4)$$

где W – ширина модели, s – длина шага, P_x – длина проекции шага на ось X , P_z – длина проекции шага на ось Z , x – позиция по оси X после шага, z – позиция по оси Z после шага, α – угол поворота.

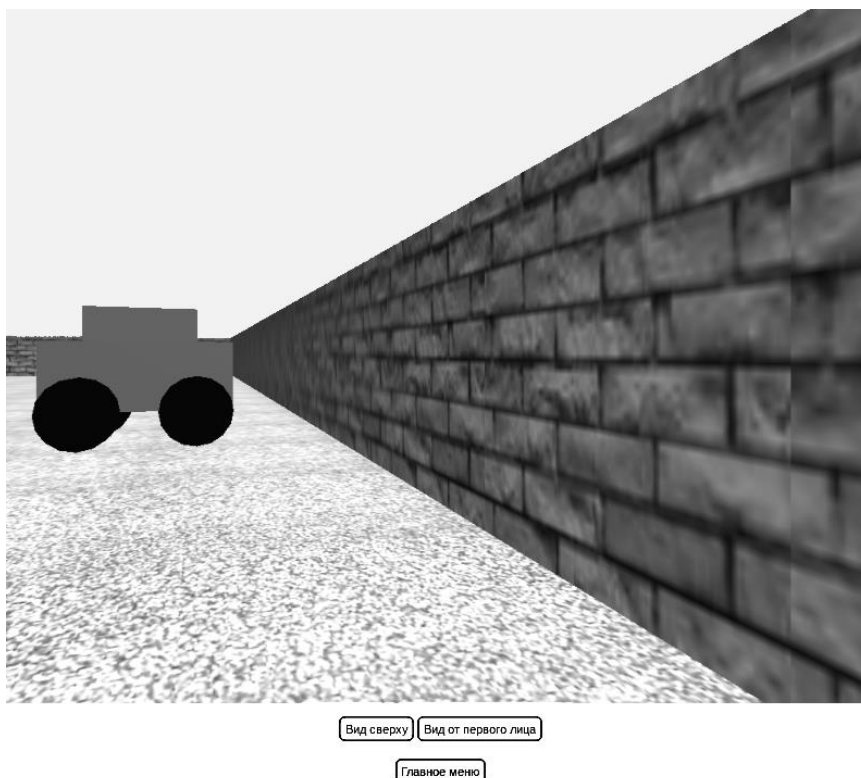


Рис. 5. Режим тренировки 3D-тренажера

4.3. Сохранение результатов тренировок

В целях предоставления возможности анализировать результаты тренировок пользователей, необходимо хранить результаты не на компьютере клиента, а в базе данных на отдельном сервере. Сохраняя данные об успешных и неудачных тренировках определённых пользователей, можно проанализировать их навыки и умения управления виртуальной моделью инвалидной коляски с помощью ИМК. В данном проекте принято решение использовать СУБД PostgreSQL, потому что взаимодействие с базой данных осуществляется путём использования языка структурированных запросов SQL, которой прост в использовании. К тому же на сегодняшний день СУБД PostgreSQL является одной из самых популярных и широко используемых СУБД из существующих, а также поддерживается многими облачными платформами, что гарантирует наличие понятной

документации, множества литературы и других ресурсов. С целью хранения данных о результатах прохождения тренировок, создана одна реляционная таблица, имеющая следующие поля:

- 1) Ключевое поле – счётчик (суррогатный ключ);
- 2) Матрица для хранения карты уровня (хранится в виде простой текстовой строки);
- 3) Название уровня (текстовая строка);
- 4) Имя пользователя (логин пользователя, который проходил тренировку);
- 5) Дата и время тренировки;
- 6) Результат тренировки (строка, которая имеет значение «YES» в случае успешного прохождения тренировки; в случае неудачного прохождения тренировки строка имеет значение «NO»).

5. Анализ результатов

В результате анализа функциональности разработанной системы управления виртуальной моделью инвалидной коляски в 3D-тренажере при помощи ИМК были выявлены качественные результаты, представленные в таблице 1.

Таблица 1.

Анализ результатов разработки 3D-тренажера

Результат		С использованием 3D-тренажера	Без использования 3D-тренажера
Трудоёмкость создания карты		2-3 мин	8 часов (ThreeJS)
Режим тренировки	Возможность удаленного тренинга	Присутствует (дом, поликлиника, больница)	Отсутствует (тренировки проводятся только в специально оборудованных помещениях для тренировок в реабилитационных центрах)
	Количество попыток в день	Не ограничено	1 (ограничено очередью)
	Риск травматизма	Отсутствует	Присутствует
	Присутствие специалиста-оператора	Не требуется	Необходимо
Автоматизированный анализ результатов и сбор статистики		Возможен	Невозможен

Стоит отметить, что в задачи разработки и проектирования системы управления виртуальной моделью инвалидной коляски в 3D-тренажере при помощи ИМК не входило повышение точности определение выбранного пользователем направления движения. Так как разработка алгоритмов предобработки сигналов от ЭЭГ для формирования максимально полезных обучающих выборок, поступающих на вход

нейронной сети, и проектирование структуры нейронной сети требуют отдельного рассмотрения.

Заключение

В работе проведен анализ технологии получения и обработки сигналов с головного мозга и разработана система управления виртуальной моделью инвалидной коляски в 3D-тренажере при помощи ИМК, благодаря которой люди с ограниченными возможностями смогут эффективно осваивать вспомогательный интерфейс управления инвалидной коляской. В процессе реализации системы: осуществлено взаимодействие с предоставленным ЭЭГ ИНЭУМ им. И.С. Брука; осуществлено взаимодействие веб-сокета сервера с 3D-тренажером и вычислительным узлом, взаимодействующим с ИМК; разработаны алгоритмы и написан соответствующий программный код.

Список литературы

1. Фролов А.А., Бобров П.Д. Интерфейс мозг-компьютер: нейрофизиологические предпосылки и клиническое применение. // Журнал высшей нервной деятельности имени И.П. Павлова. Специальный выпуск: интерфейс мозг – компьютер. 2017. Т. 67. №4. С. 365-376. DOI: [10.7868/S0044467717040013](https://doi.org/10.7868/S0044467717040013)
2. Мокиенко О.А., Черникова Л.А., Фролов А.А. Интерфейс мозг-компьютер как новая технология нейрореабилитации. // Анналы клинической и экспериментальной неврологии. 2011. № 3. С. 46.
3. Мусакулова Ж. А. Разработка нейросетевой автоматизированной системы классификации данных ЭЭГ. // Альманах современной науки и образования. Тамбов: Грамота. 2014. № 2. С. 118-120.
4. Мусакулова Ж.А. Нейросетевой анализ данных электроэнцефалограммы. // Альманах современной науки и образования. Тамбов: Грамота. 2017. № 4-5. С. 68-72.
5. Таненбаум Э., Уэзеролл Д. Компьютерные сети. 5-е изд. СПб.: Питер. 2012. 960 с.
6. The WebSocket Protocol // Internet Engineering Task Force Tools: сайт. Режим доступа: <https://tools.ietf.org/html/rfc6455> (дата обращения: 12.05.2018г.)

Акустический процессор формантного синтезатора речи с фонемным возбуждением

²⁵Хартов В.Я., ²⁶Белкина Т.А.

МГТУ им. Н.Э. Баумана (НИУ), Москва, Россия

khartovbmstu@gmail.com

squirrela@list.ru

Аннотация: Цель этой статьи – предложить вариант реализации акустического процессора синтезатора речи, не использующего записанных элементов человеческой речи и способного генерировать сообщения по произвольному тексту. В этой работе используется формантный синтез по правилам, позволяющий иметь неограниченный словарь, так как речь создается из отдельно генерируемых звуков. В рамках формантной модели описываются частотные характеристики речи, которые содержат основную информацию о произносимых фонемах. Предложен вариант реализации формантного синтезатора речи как микропроцессорного устройства, вырабатывающего все необходимые частоты речевого сигнала с использованием широтно-импульсной модуляции. Получена функциональная схема генераторного варианта синтезатора. Разработан формат специальной управляющей таблицы – “кодовой книги” – для параметров формантных частот звуков русского языка. Главное преимущество формантного синтезатора – низкая скорость передачи данных, управляющих речевым выходом, и простота схемотехнических решений при создании подобного синтезатора. Разрабатываемое устройство может быть использовано в системах оповещения на транспорте.

Ключевые слова: акустический процессор, кодовая книга, фонема, форманта, формантный синтез.

Acoustic processor of a formant speech synthesizer with phonemic excitation

Khartov V.Ya., Belkina T.A.

Bauman Moscow State Technical University, Moscow, Russia

khartovbmstu@gmail.com

squirrela@list.ru

Abstract: The purpose of this article is to suggest a solution for an acoustic processor of a speech synthesizer that does not require pre-recorded speech segments and can generate

²⁵Хартов Вячеслав Яковлевич — кандидат технических наук, доцент кафедры «Компьютерные системы и сети» МГТУ им. Н.Э. Баумана.

²⁶Белкина Татьяна Алексеевна — студент магистратуры кафедры «Компьютерные системы и сети» МГТУ им. Н.Э. Баумана.

speech from any input text. This solution uses formant synthesis by rules, which allows having an unlimited dictionary, since speech is created from separately generated sounds. The formant model describes frequency characteristics that contain the basic information about the phonemes being spoken. The proposed solution allows implementing formant synthesizer as a microprocessor device that generates all necessary speech frequencies using pulse width modulation. This paper presents a functional block diagram of the generators-based device and describes a format for a “code dictionary” – a control table that could store the parameters of formant frequencies for all sounds of the Russian language. The main advantage of a formant synthesizer is the low transfer rate of data controlling speech output, and thus the simplicity of circuitry solutions when creating such a synthesizer. The proposed device is meant to be used in announcement systems on public transport.

Key words: acoustic processor, code dictionary, phoneme, formant, formant synthesis.

Введение

Основная задача, которую призван решить синтезатор речи в системе информационного оповещения, – снять зависимость разработчиков таких систем от внешнего диктора и процедур работы с ним. То есть, синтезатор не должен использовать готовых записанных элементов человеческой речи, а синтезировать их в процессе высказывания. Поэтому было принято решение о построении акустического процессора синтезатора с применением формантного синтеза по правилам. Преимущество формантного метода синтеза – в его универсальности, позволяющей иметь неограниченный словарь, так как здесь речь создается из отдельно генерируемых звуков. Правильно сгенерировав и расставив звуки, можно произнести любое слово. Разборчивость и натуральность таких синтезаторов может быть доведена до уровней, сравнимых с характеристиками естественной речи.

Под формантами понимаются частотные резонансы речевого тракта человека. Параметры формант (частота, ширина, уровень) определяются акустическими свойствами речевого тракта. Наиболее важный параметр – частота форманты, – тесно связан с геометрической конфигурацией речевого тракта. Поскольку в процессе речи конфигурация речевого тракта меняется – меняется положение и состояние артикуляторов, – то, соответственно, меняются формантные частоты.

В рамках формантной модели описываются частотные характеристики речи, которые содержат основную информацию о произносимых фонемах. Акустические характеристики фонем тесно связаны с артикуляционными особенностями их образования – местом и способом. Речевой аппарат человека представляется в виде двух параллельных каналов – ротового и носового, образующих единую акустическую систему, возбуждаемую периодическими колебаниями голосовых связок либо турбулентным шумом.

1. Определение основных параметров формантного синтеза

В основе формантного синтеза лежит аналогия с моделью речеобразования человека. Рассмотрим формирование гласных звуков на модели, представленной на рисунке 1.

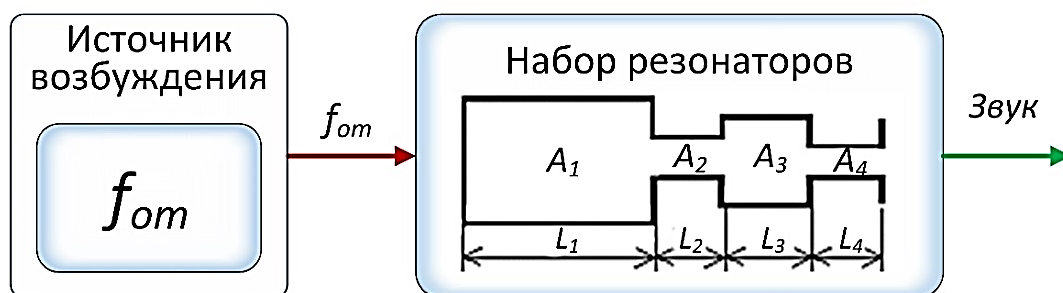


Рис. 1. Четырехтрубная модель речевого тракта при произнесении гласных звуков

Источник возбуждения вырабатывает звуковые колебания с частотой основного тона, причем, частота эта непрерывно меняется в процессе высказывания. Речевой тракт при произнесении гласных звуков работает как набор резонаторов, в которых происходит фильтрация сигнала возбуждения. Первый резонатор (с площадью поперечного сечения A_1 и длиной L_1) имитирует гортань и ротовую полость до сужения, создаваемого языком (задняя полость), второй (A_2 и L_2) – участок сужения между языком и твердым небом, третий (A_3 и L_3) – переднюю ротовую полость и, наконец, четвертый (A_4 и L_4) – проход между губами. Каждому элементарному звуку речи – фонеме – соответствует определенная форма акустических резонаторов, обусловленная положением артикуляторов. При переходе от одного звука к другому форма резонаторов плавно изменяется.

В результате образуется спектральная картина, содержащая ряд максимумов. Максимумы соответствуют резонансам речевого тракта – это форманты. Таким образом, форманты – это некоторая частотная область концентрации энергии в спектре звука. Понятие форманты иллюстрируется примером частотного спектра речи на рисунке 2.

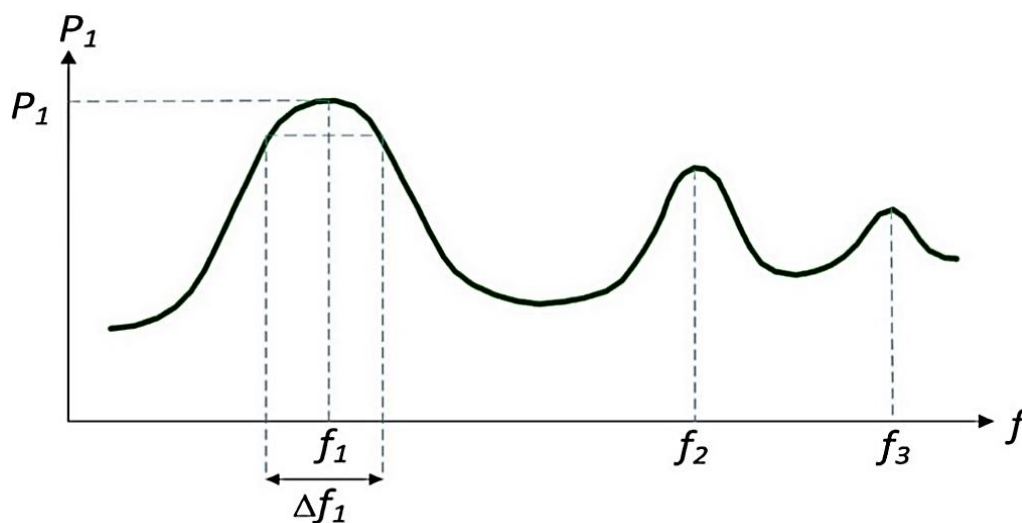


Рис. 2. Пример частотного спектра речи

2. Структурная схема формантного синтезатора речи

Как утверждается в работе [1], для удовлетворительного синтеза речи обычно нужны две – четыре формантные частоты. Первые две (основные) форманты определяют произносимый звук речи, а остальные (вспомогательные) характеризуют индивидуальную для каждого человека окраску, тембр речи. В рамках проводимых исследований было принято решение ограничиться тремя формантами f_1 , f_2 и f_3 , как наиболее информативными. Точным расположением формантных частот в звуковом спектре и определяется звук, который человек интерпретирует как речь.

Приведем таблицу формантных частот для фонем гласных звуков (таблица 1). Путем одновременной генерации формантных частот f_1 , f_2 , f_3 согласно таблице 1 можно получить гласные звуки.

Таблица 1.

Формантные частоты гласных звуков

Фонемы	Формантные частоты		
	f_1	f_2	f_3
<i>a</i>	700	1080	2600
<i>o</i>	535	780	2500
<i>y</i>	300	625	2500
<i>э</i>	440	1800	2550
<i>и</i>	240	2250	3200
<i>ы</i>	300	1100	3200

Формантный синтезатор гласных звуков включает задающий генератор частоты основного тона, полосовые фильтры, перестраиваемые на формантные частоты, соответствующие синтезируемой фонеме, и сумматорах [2]. В спектрограмме выходного сигнала этой схемы содержатся три формантные частоты, идентичные формантным частотам в спектрограмме речи человека, произносящего те же гласные.

Гораздо сложнее формировать согласные звуки. Согласные – звуки речи, при произношении которых в полости рта образуются преграды для выдыхаемого воздуха [3]:

- взрывные согласные – при полном смыкании органов речи – *б, п, т, к*;
- фрикативные согласные – образуется щель – *с, ф, х*;
- сонорные: носовые согласные – *н, м, л* и *р*;
- аффрикаты – согласные звуки, представляющие собой слитное сочетание двух других – *ч -тиц, ц -тс*.

Чтобы расширить диапазон возможностей синтезатора, необходимо ввести источник шума для формирования взрывных и фрикативных согласных, а также аналог носовой резонансной полости, имитирующий носовые согласные. Структура этого расширенного формантного синтезатора приведена на рисунке 3.

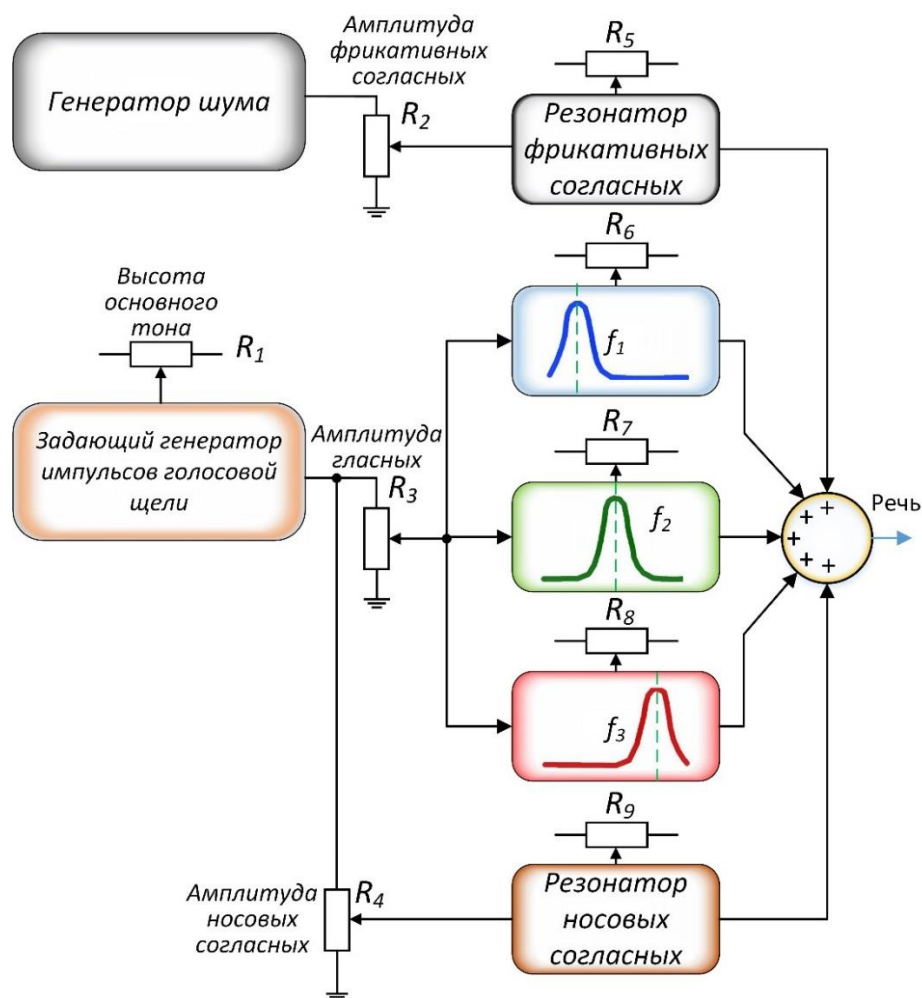


Рис. 3. Структурная схема формантного синтезатора речи

3. Формирование звуковых частот

Реализация функциональной модели с полосовыми фильтрами формант потребует либо применения высокопроизводительного микропроцессора с математическим ядром, позволяющим проводить преобразования Фурье, либо внешней периферии. В данной ситуации логичным видится шаг по переходу от модели фильтров к модели генераторов, вырабатывающих все необходимые частоты речевого сигнала с использованием широтно-импульсной модуляции (ШИМ).

Выбор ШИМ для формирования звуковых колебаний обусловлен тем, что для синтезирования различных звуков может требоваться разная форма сигнала. Так для гласных она будет близка к форме резонансной кривой, а для назальных согласных – к синусоиде. Если использовать прямоугольные импульсы, потребуется внешний фильтр НЧ, «заваливающий» фронты импульсов (что не делает этот вариант неприемлемым, при испытаниях синтезатора он также должен быть исследован, фильтр для ШИМ также нужен, но обычно он существенно проще). Форму для ШИМ можно хранить в виде шаблона – таблицы амплитуд полупериода, например. Кроме того, формирование

выходной частоты при помощи ШИМ позволит осуществить необходимую регулировку амплитуды, что в перспективе позволит упростить схему суммирующего усилителя.

Осциллограммы на рисунке 4 иллюстрируют принцип формирования синусоидального сигнала при помощи ШИМ. Отчетливо видна пропорциональность коэффициента заполнения ШИМ (1-й канал) амплитуде отфильтрованного сигнала (2-й канал).

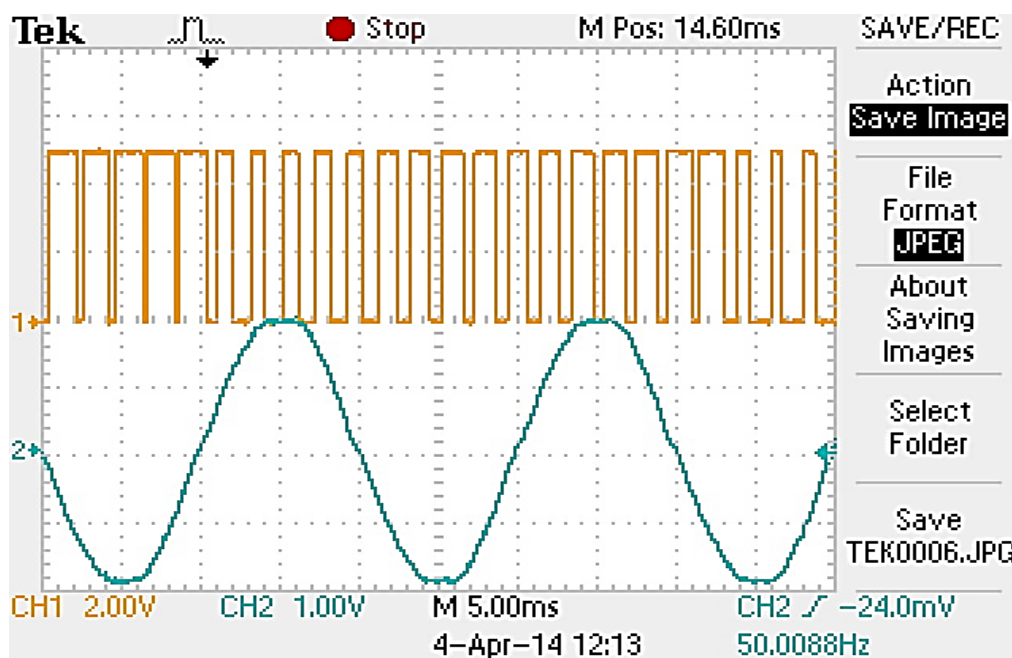


Рис. 4. Принцип формирования сигнала при помощи ШИМ

4. Функциональная схема микропроцессорного синтезатора речи

Используя функциональную схему формантного синтезатора речи, основанную на моделировании строения человеческого речевого тракта, нам удалось определить количество и характер изменения параметров и их регулировок, действующих в этой модели. Собранные материалы по формантным составляющим звуков, принципам их генерации и слияния, делают возможным переход к генераторной модели синтезатора.

При переходе к генераторной модели синтезатора речи число функциональных генераторов должно быть не меньше числа полосовых фильтров резонаторной модели. Также должны быть сохранены все определенные нами параметры и их регулировки, а генераторы должны иметь возможность принимать эти параметры и реагировать на их регулировки. Поскольку речь идет о микропроцессорном устройстве, а генераторы реализуются программным образом, все регулировки должны представляться в цифровом коде.

Всем перечисленным требованиям удовлетворяет функциональная схема генераторного варианта синтезатора речи, представленная на рисунке 5, принятая в данном проекте как рабочая. Разработка этой схемы велась параллельно и взаимосвязано с разработкой алгоритма функционирования акустического процессора и кодовой книгой синтезатора, хотя данные разделы и излагаются последовательно.

Функциональные узлы синтезатора являются частью его микропрограммы, а сама функциональная схема служит лишь для пояснения работы устройства. На вход схемы поступает предварительно подготовленный текст согласно вспомогательной технологии. Затем в работу вступает фонетический процессор. Выходом программного модуля фонетического процессора является фонетическое письмо – текст, преобразованный в фонемы – в индексы кодовой книги, описание которой последует ниже.

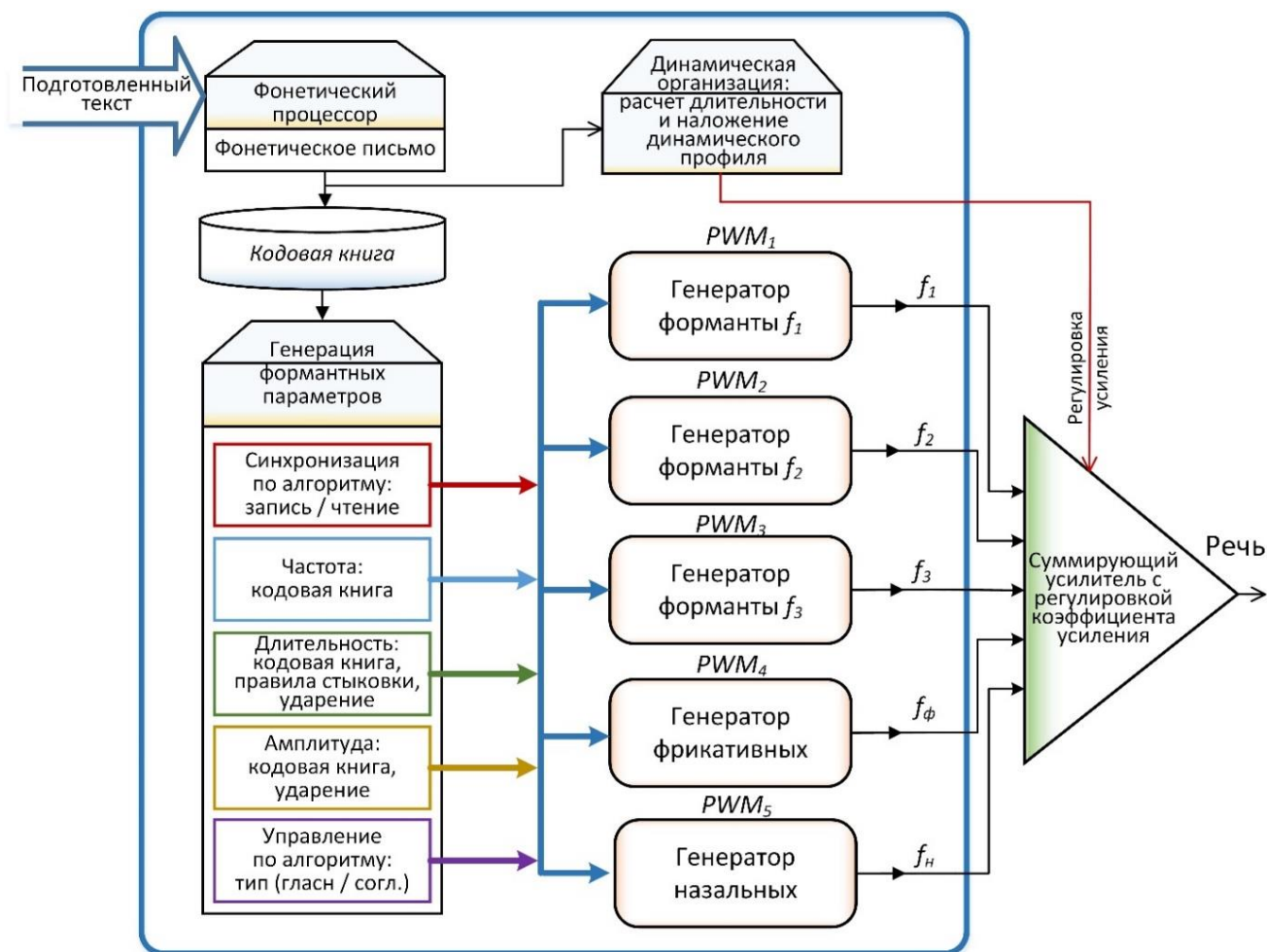


Рис. 5. Микропроцессорный формантный синтезатор речи:
а) функциональная схема

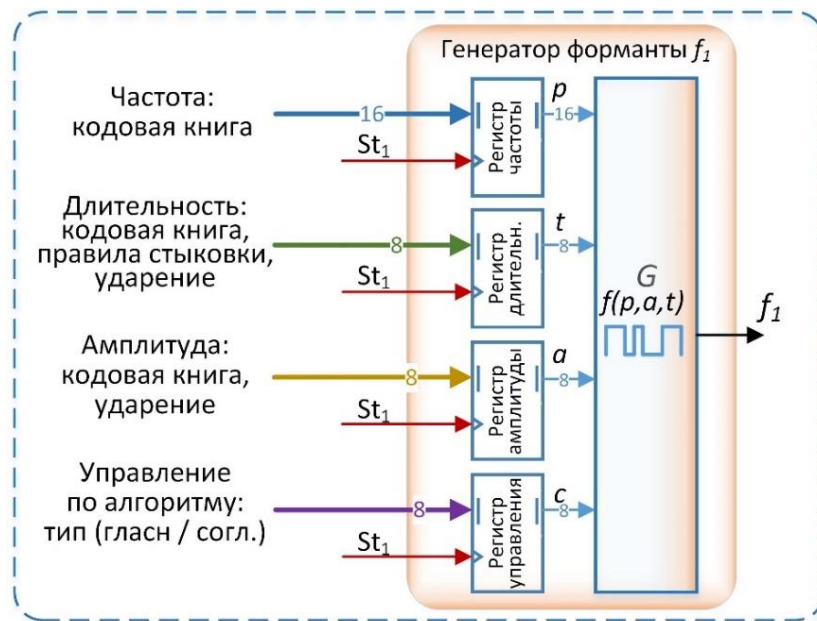


Рис. 5. Микропроцессорный формантный синтезатор речи:
б) генератор ШИМ (PWM1)

Информация из кодовой книги поступает в модуль генерации формантных параметров. Этот модуль осуществляет управление всеми генераторами синтезатора, а именно:

- передает каждому генератору полученную из кодовой книги частоту последовательности;
- вычисляет на основе полученного фонетического письма и записи кодовой книги амплитуду последовательности, действуя по схеме: «гласный (ударный / безударный) – согласный: значение из кодовой книги»;
- вычисляет длительность последовательности (сколько будет звучать данный звук) согласовываясь с данными, полученными из фонетического письма (ударный / безударный), из кодовой книги (длительность по умолчанию), и руководствуясь правилами стыковки звуков.
- выполняет управление всеми генераторами: передает им тип звука, осуществляет своевременную синхронизированную передачу каждому генератору всех требующихся параметров, для генераторов согласных определяет выбор источника шума.

Генераторы частот формант могут быть реализованы с использованием функционала таймеров (счетчики и их прерывания) микропроцессора, или в виде параметрических циклов основной программы. И в том, и в другом случае результатом будет генерация последовательности импульсов с соответствующим коэффициентом заполнения – ШИМ-последовательности – на выбранном выводе микропроцессора. Сигналы должны пройти через фильтры низкой частоты чтобы лишиться высокочастотной составляющей, и поступить на входы усилителя-сумматора с регулируемым коэффициентом усиления.

Помимо объединения частотных каналов, усилитель-сумматор совместно с модулем динамической организации обеспечивает динамическую имитацию интонации генерируемого высказывания [4]. В процессе произнесения фразы производится изменение электрического уровня сигнала на выбранном выводе микропроцессора, для чего может использоваться либо встроенный ЦАП, либо дополнительный канал ШИМ. Изменение потенциала на этом выводе производится согласно выбранному динамическому профилю, и приводит к изменению коэффициента усиления суммирующего усилителя, что, в свою очередь, приводит к изменению громкости фразы по мере ее звучания.

5. Кодовая книга

Микроконтроллер в состоянии надлежащим образом синтезировать изменения формантной частоты путем перенастройки каналов ШИМ. Для этого ему необходимо обращаться к специальной управляющей таблице – кодовой книге, – где хранятся все параметры фонемы, находящейся в обработке: её длительность, тип – гласная, согласная (какая согласная), частоты формант, и амплитуда генерации по умолчанию. Формат записи кодовой книги приведен на рисунке 6.

Имя	Тип	Ампл.	Длит.	Частоты			
				1-я форманта	2-я форманта	3-я форманта	шум
0xFF	0xFF	0xFF	0xFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF

Рис. 6. Формат записи кодовой книги формантного синтезатора речи

Поле имени фонемы имеет размер в один байт и может служить для поиска / адресации в кодовой книге.

Поле типа фонемы введено для передачи модулю генерации информации об адресуемых генераторах, может принимать следующие значения:

- 0x01 – гласная;
- 0x02 – гласная йотированная;
- 0x03 – согласная вокализованная твердая;
- 0x04 – согласная вокализованная мягкая;
- 0x05 – согласная невокализованная твердая;
- 0x06 – согласная невокализованная мягкая;

Решение поместить амплитуду в кодовую книгу было принято исходя из тех соображений, что в естественной человеческой речи амплитуда различных звуков весьма неодинакова (достаточно вспомнить о глухих согласных). Если в генерируемой речи все звуки будут одинаковой амплитуды, то она станет весьма трудно воспринимаемой на слух, особенно – в части согласных. Поэтому в кодовой книге введено поле амплитуды, которая задается в восьмибитном формате в относительных единицах уровня.

Поле длительности фонемы содержит длительность фонемы по умолчанию. Усредненные значения длительности фонем составляют около 0,15с для гласных и около 0,08с для согласных (звук «п» звучит около 0,03с). И, казалось бы, информацию о длительности можно было бы получить на основе самой фонемы. Но, во-первых, указанные выше значения являются все-таки усредненными, и в процессе отладки может потребоваться внести уточнения в значения длительности, а во-вторых – модуль вычисления длительности на основе анализа фонемы займет в памяти микропроцессора место, соизмеримое с суммой дополнительных полей в кодовой книге, а доступ к байтовому полю записи будет существенно быстрее, чем выполнение кода определения длительности на основе анализа фонемы.

Поле каждой частоты имеет разрядность двух байт. Таким образом, для частотного описания гласного звука из трех формант потребуется 6 байт, для частотного описания согласного звука – три форманты и шум – 8 байт. Для унификации в кодовой книге под частотное описание каждой фонемы выделено по 4 поля. Пустующее поле шума для гласных звуков можно будет в дальнейшем использовать, заставляя генератор назальных согласных генерировать четвертую форманту гласных для создания спецэффектов, например. Пустующие поля формант для глухих (невокализованных) согласных остаются невостребованными, но с этим можно смириться в целях унификации и упрощения доступа к записям кодовой книги.

Таким образом, описания одной фонемы в кодовой книге занимает 12 байт, а вся кодовая книга для 42 фонем русского языка займет в памяти микропроцессора 504 байта, что является абсолютно приемлемым для подавляющего большинства современных микропроцессоров.

Записи кодовой книги синтезатора составлены на основе различных источников, содержащих усредненные и часто не совсем точные данные, а также в ходе проведенных экспериментов. Для выравнивания тембральной характеристики генерируемой речи, чтобы все звуки звучали как произнесенные одним речевым аппаратом, потребуется дополнительная работа при отладке синтезатора.

Заключение

Метод синтеза формантных частот речи и его результат довольно специфичен. Если прослушать речь синтезаторов других типов и сравнить ее с речевым выходом синтезатора формантного типа, то их без труда можно различить. По мере совершенствования динамического изменения управляющих параметров речь формантного синтезатора становится более разборчивой. Этот метод синтеза речи весьма перспективен, и развитие проекта будет продолжено. Главное преимущество формантного синтезатора – очень низкая скорость передачи данных, управляющих речевым выходом, и, соответственно, простота схемотехнических решений при создании подобного синтезатора.

Тем не менее, требуется дальнейшее усовершенствование алгоритма акустического процессора в части плавности синтеза речи и имитации интонации. Временные задержки между циклами генерации отдельных звуков, обусловленные передачей данных для нового цикла генерации, создают пусть малозаметные, но мешающие адекватному восприятию синтезированной речи рывки.

В рамках данной работы был собран и обобщен большой объем знаний в смежных областях научных и технологических дисциплин, необходимый для создания формантного синтезатора речи. В частности, разработан алгоритм фонетического процессора, учитывающий взаимовлияние фонем, а также составлена полная таблица формантных частот звуков русского языка. Разработан алгоритм генерации звуков, в котором учтены правила слияния звуков, повышающие разборчивость и естественность речи.

Проведенные эксперименты доказали верность концепции построения синтезатора по формантному принципу генерации речи и перспективность дальнейшего развития проекта.

Список литературы

1. Кейтер Дж. Компьютеры – синтезаторы речи. (Electronically Speaking: Computer Speech Generation, 1983) / Перевод с английского Э.Я. Пастрона под редакцией В.А. Усика. М.: Мир. 240 с.
2. Савчук В.Л. Формантный синтез. [Электронный ресурс]. Режим доступа: <http://masters.donntu.org/2008/fvti/demenko/library/formant.htm> (Дата обращения: 8.05.2018).
3. Бондарко Л.В. Звуковой строй современного русского языка. Учеб. Пособие для студентов педагогических институтов по специальности «Рус. Яз. и литература». М.: «Просвещение». 1977. 175 с.
4. Мещеряков Р.В. Модель системы синтеза параметров речи по печатному тексту с использованием информационных представлений. [Электронный ресурс]. Режим доступа: http://elib.altstu.ru/elib/books/Files/pa1999_2/pages/08/pap_08.html (Дата обращения: 8.05.2018).

Подход к проектированию механизма обновления баз правил системы обнаружения атак

²⁷Пугачев Е. К., ²⁸Лапина Н.А.

МГТУ им. Н.Э. Баумана (НИУ), Москва, Россия

iub.pugachev@yandex.ru

dinka119@yandex.ru

Аннотация: В статье представлен иерархический подход к построению системы обнаружения атак, описан функционал компонентов для каждого уровня иерархии, определён алгоритм взаимодействия уровней. В частности, показаны возможности иерархической структуры при реализации механизма обновления данных и рассмотрен процесс передачи данных между уровнями иерархии. Особое внимание уделено механизму обновления баз правил, используемых при выявлении атак в компьютерной сети.

Ключевые слова: компьютерная сеть, иерархический подход, система обнаружения атак, механизм обновления, база решающих правил, вторжение, атака, уязвимость.

Введение

В результате анализа систем безопасности было определено, что компьютерные сети даже с самой совершенной защитой нельзя назвать абсолютно неуязвимыми. В связи с вышесказанным, повышение уровня безопасности систем остается важной задачей.

В настоящее время актуальное значение приобрела задача создания надежной и автоматизированной системы обнаружения атак. Одной из основных функций такой системы является обнаружение в сетевом трафике действий, направленных на преднамеренный доступ к информации с целью её добывания, уничтожения и искажения.

1. Алгоритм работы системы обнаружения вторжений

При проектировании системы обнаружения вторжений (СОВ) необходимо учесть, что ключевым моментом является автоматизация деятельности служб информационной безопасности. Исследования показали, что основными задачами таких служб должны быть:

²⁷Пугачев Евгений Константинович - кандидат технических наук, доцент кафедры "Компьютерные системы и сети" МГТУ им. Н.Э. Баумана.

²⁸Лапина Надежда Андреевна - магистрант кафедры "Компьютерные системы и сети" МГТУ им. Н.Э. Баумана.

- выявление вторжений в компьютерные сети;
- формирование событий информационной безопасности;
- формирование уведомлений об обнаружениях и мониторинг вторжений в компьютерной сети.

Механизм функционирования разрабатываемой СОВ заключается в следующем: датчики слушают сетевой трафик в компьютерной сети и детектируют угрозы на основе базы решающих правил и базы уязвимостей.

При обнаружении вторжений осуществляется фиксация факта нарушений безопасности в журнале аудита, после чего администратор безопасности уведомляется об этом с помощью графического интерфейса.

Было определено, что типовая атака имеет три этапа: подготовка, реализация и завершение.

На этапе **«подготовка»** злоумышленник собирает информацию. Для этого он изучает структуру сети, окружений, осуществляет сканирование портов и узлов, идентификацию операционной системы (ОС), изучает программное обеспечение (ПО).

На этапе **«реализация»** выполняются зловредные программы, т.е. осуществляется непосредственно атака.

На этапе **«завершение»** происходит чистка логов, ссылок и связей, а также удаление тела программы и своих модулей.

Система обнаружения вторжений должна фиксировать атаку на втором и третьем этапе. При проектировании систем такого класса применяются разные подходы, но все они, как правило, одноуровневые.

2. Иерархическая модель СОВ

В данной статье предлагается подход проектирования СОВ, особенностью которого является иерархичность ее компонентов. Подход выражается в том, что все устройства, слушающие трафик, имеют подчиненность, что повышает надёжность системы в целом. Осуществляется это за счет распределения возможностей каждого уровня иерархии.

Предлагается все компоненты делить на два вида: родительские и дочерние. Также в рамках данной работы были сформулированы правила, которые необходимо использовать при проектировании системы обнаружения вторжений. Ниже приведено описание правил и принципа работы иерархической СОВ.

Компоненты верхних уровней управляют компонентами нижних уровней. При этом компоненты верхних уровней имеют больше возможностей по сравнению с компонентами нижних уровней. В частности, вышестоящий компонент может осуществлять обновление данных.

На рис.1 представлен фрагмент иерархической структуры компонентов системы вторжений. Из рисунка видно, что в иерархии есть корневой компонент. Он является

компонентом самого высокого уровня и непосредственно связан с сервером обновлений. При этом на сервер обновлений поступают обновления всех баз.

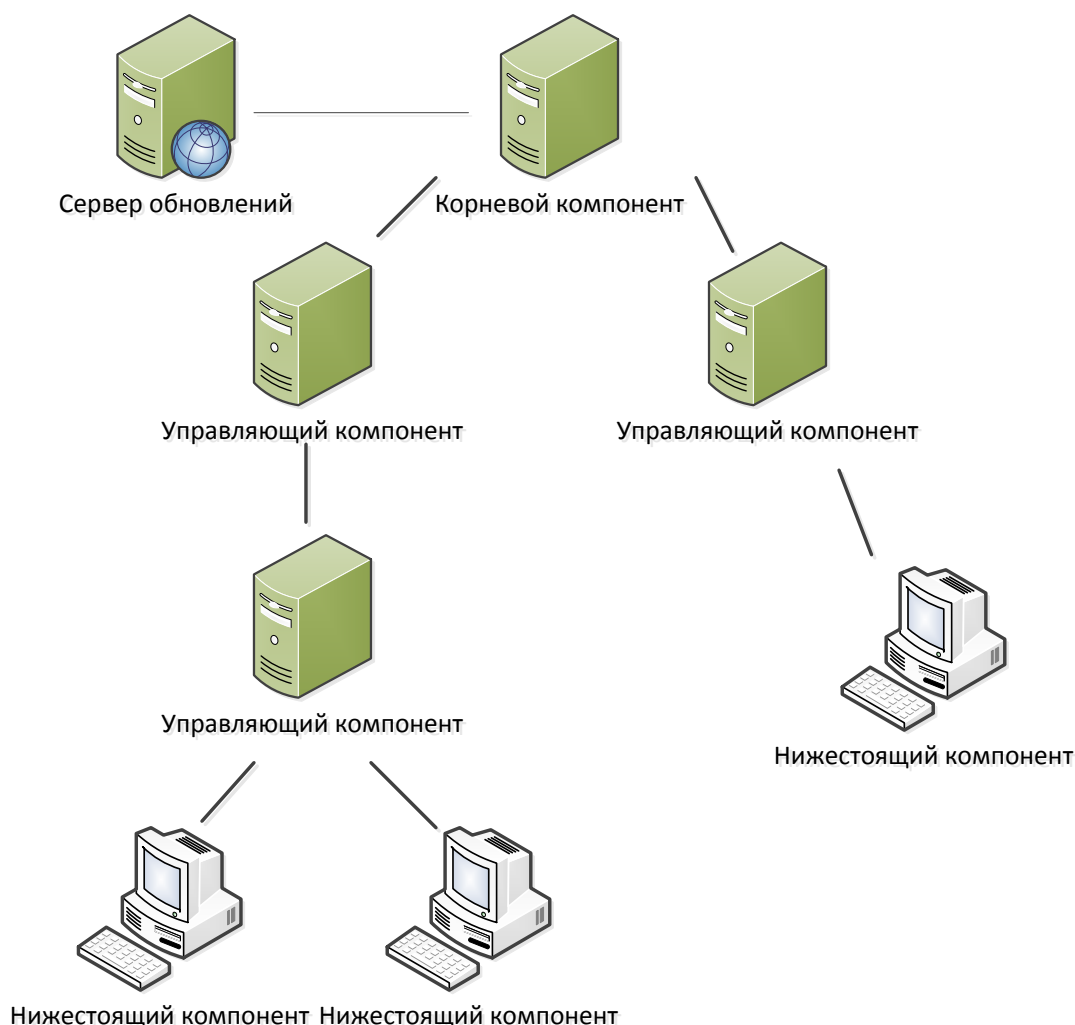


Рис. 1. Фрагмент иерархической структуры компонентов системы

Дочерний компонент устанавливается на рабочее место пользователя. К его функциям относятся:

- Сбор информации о действиях пользователя, его ПО, ОС.
- Регистрация сетевых вторжений за счет анализа сетевого трафика.
- Регистрация событий аудита.
- Сбор и накопление выявленной при анализе информации.
- Передача данных на вышестоящий компонент, который непосредственно осуществляет управление (копию трафика, данные обнаруженных вторжений, события аудита, параметры работы нижестоящего компонента).
- Раскатка обновлений, поступающих от вышестоящего компонента.
- Контроль за свободным дисковым пространством.

Взаимодействие компонентов осуществляется по защищённому каналу связи. Для проверки и защиты от несанкционированного доступа используется цифровые сертификаты.

Управляющие компоненты функционируют на выделенной машине, которая отвечает за выполнение следующих функций:

- поддержка иерархической структуры компонентов;
- отображение в графическом интерфейсе результатов анализа и мониторинга событий информационно безопасности;
- прием данных от нижестоящих подчиненных, передача - на вышестоящие компоненты;
- сбора статистики (формирование профилей хостов);
- регистрация событий аудита;
- настройка параметров своих и подчиненных компонентов;
- взаимодействие с сервером обновлений.

3. Механизм обновления базы правил

Система обнаружения атак решает широкий спектр разнообразных задач. Одной из основополагающих является задача обновления базы с целью поддержки в актуальном состоянии базы решающих правил и базы уязвимостей.

За обновление системы обнаружения атак отвечает отдельный модуль системы с одноимённым названием. Обновление поступает из сервера обновлений на корневой компонент и далее распространяется вниз по иерархии компонентов. Для защиты файлов обновлений от искажений используется контроль целостности.

Обновление представляет собой набор отдельных сжатых файлов. Для контроля версии каждого типа обновления имена файлов содержат текущую дату, номер обновления и тип (под типом имеется в виду обновление базы решающих правил или базы уязвимостей).

Модуль отслеживает появление файлов обновлений в вышестоящем компоненте. При наличии обновлений модуль скачивает сжатые файлы обновлений и распаковывает их. За исключением справочников – они не скачиваются.

При обновлении импортируются в БД:

- база решающих правил сигнатурного анализа,
- база уязвимости.

Для обновления корневого датчика выполняется регистрация на сервере обновлений. Проверка появления обновлений выполняется за счет мониторинга публикаций файлов на сервере обновлений с помощью HTTP-запросов. При наличии обновлений модуль скачивает сжатые файлы обновлений и распаковывает их.

Модуль обновления устанавливает следующие файлы обновлений:

- базы решающих правил сигнатурного анализа,
- ПО для компонентов нижнего уровня.

Справочные данные передаются из БД вышестоящего компонента в БД подчинённого компонента. Таким образом, на вышестоящий компонент поступают файлы обновлений, которые далее «спускаются» вниз по иерархии.

В результате на нижестоящие компоненты передаются обновлённые решающие правила сигнатурного анализа, справочники, база уязвимостей.

Копия трафика создаётся в нижестоящем компоненте в виде файлов, содержащих информацию о сетевом пакете, вызвавшем срабатывание решающего правила. События информационной безопасности также появляются в компонентах самого нижнего уровня. Далее они, как и файлы с копией трафика, передаются вверх по иерархии компонентов. Передача выполняется с учётом приоритета: в первую очередь на вышестоящий компонент посылаются события с более высоким уровнем критичности (этот параметр есть у каждого события).

Заключение

Рассмотренный механизм обновления баз правил раскрывает особенности работы и управления компонентами в предложенной иерархичной структуре системы. Их иерархичность позволяет разделить полномочия компонентов, разграничить их функционал, упростить реализацию самых нижестоящих компонентов, слушающих сетевой трафик, за счет отсутствия графического интерфейса и средств анализа событий информационной безопасности.

Список литературы

1. Гашимова К.Р., Пугачев Е.К. Подход к разработке модели протокола динамической маршрутизации // Инженерный вестник. Электронный научно-технический журнал. Издатель ФГБОУ ВПО "МГТУ им. Н.Э. Баумана". 2016. №7. С. 501-507. Режим доступа: <http://engsi.ru/doc/843310.html> (дата обращения: 1.02.2018)
2. Васильев В. И., Свечников Л. А., Кашаев Т. Р. Комплексный подход к построению интеллектуальной системы обнаружения атак // Системы управления и информационные технологии. Воронеж: "Научная книга". 2007. №2. С. 76-82.
3. Защита информации. Основные термины и определения: ГОСТ Р 50922-2006. Введ. 1.02.2008. М.: Стандартиформ. 2008. 8 с.
4. Васильев В.И., Свечников Л.А. Архитектура распределенной системы обнаружения атак // Информационная безопасность: Материалы 8-й международной научно-практической конференции. Издатель: ТРТУ. 2006. С. 180-184.

Сравнительный анализ способов конвертирования структур данных между базами данных платформы «1С: Предприятие»

²⁹Ничушкина Т.Н., ³⁰Гузий Е.А., ³¹Федоренков В.В.

МГТУ им. Н.Э. Баумана (НИУ), Москва, Россия

tnich@rambler.ru
zh.guziy@mail.ru
vfedorenkov@mai.ru

Аннотация: Задачей работы является сравнение используемых в настоящее время способов конвертирования структур данных между базами данных платформы «1С: Предприятие», которое необходимо для поддержания баз в согласованном состоянии.

В результате, на основании полученных результатов сравнения двух алгоритмов конвертирования, которые сопоставлялись друг с другом по составленным критериям сравнения, были выявлены достоинства и недостатки каждого из них.

Рассмотрены два существующих в настоящий момент способа конвертирования структур данных, составлена система критериев сравнения алгоритмов. Получены результаты сравнения способов, что влечет постановку задачи о разработке нового способа конвертирования.

Особенностью работы является выявление существенных недостатков используемых алгоритмов обмена данными, которые в большинстве случаев приводят к потере данных в случае возникновения сбоев в работе.

Ключевые слова: платформа 1С (platform 1С), информационные базы (information bases), конвертирование (converting), структуры данных (data structures).

Comparative analysis of ways of converting data structures between databases of the platform "1С: Enterprise"

Nichushkina T.N., Gusiya E.A., Fedorenkov V.V.

Bauman Moscow State Technical University, Moscow, Russia

tnich@rambler.ru
zh.guziy@mail.ru
vfedorenkov@mai.ru

²⁹Ничушкина Татьяна Николаевна – доцент кафедры "Компьютерные системы и сети" МГТУ им. Н.Э. Баумана, SPIN-код 2675-4796.

³⁰Гузий Евгений Андреевич – магистрант кафедры "Компьютерные системы и сети" МГТУ им. Н.Э. Баумана.

³¹Федоренков Владислав Владимирович – магистрант кафедры "Компьютерные системы и сети" МГТУ им. Н.Э. Баумана.

***Abstract:** The task of the paper is to compare the currently used methods of converting data structures between "1С: Enterprise" platform databases, which is necessary to maintain the bases in an agreed state.*

As a result, based on the obtained results of comparison of two conversion algorithms, which were compared with each other according to the compiled comparison criteria, the advantages and disadvantages of each of them were revealed.

Two current methods of converting data structures are considered, a system of criteria for comparing algorithms is compiled. The results of the comparison of methods are obtained, which entails the formulation of the problem of developing a new method of converting.

A special feature of the work is the identification of significant flaws in the used data exchange algorithms, which in most cases lead to loss of data in the event of a malfunction.

***Keywords:** platform 1С, information bases, converting, data structures*

Введение

На сегодня большинство компаний занимается автоматизацией производственных процессов, позволяющей сократить время выполнения того или иного процесса. Одним из вариантов автоматизации является внедрение программных систем, позволяющих перевести обработку данных и передачу документов в электронную форму. К примеру, таким программным продуктом является платформа «1С: Предприятие» производства российской фирмы «1С».

Для автоматизации процессов используются определенные существующие информационные базы, которые работают внутри платформы. В каждой из них используется определенный набор документов, справочников и отчетов, каждый из которых связан со схожим в другой базе. Однако по структуре данных объекты разных информационных баз отличаются друг от друга. Для поддержания согласованности данных внутри баз данных (БД) используются два способа конвертирования – ручной и типовой. Ручной заключается в непосредственном создании или изменении данных в объектах информационных баз пользователями, а типовой – в использовании конфигурации фирмы «1С».

В данной статье будет выполнено сравнение двух указанных способов конвертирования структур данных и выявлены их достоинства и недостатки.

1. Анализ способов конвертирования

1.1 Ручное конвертирование

Ручное конвертирование представляет собой алгоритм, при котором некоторый сотрудник с правами администратора БД вручную создает соответствующие документы в других базах при проведении нового документа в текущей базе. Также, при корректировке данных в старых документах необходимо произвести аналогичную замену данных в соответствующих документах в других базах.

Данный способ является очень трудозатратным. Связано это с тем, что сотруднику компании необходимо все операции производить вручную, при этом при допущении, ошибки существует риск нарушить целостность информационных баз. Другими словами, одна база будет отображать в отчете одни данные, а другая – отличающиеся от предыдущих. При проведении документов с различными данными будет сложно впоследствии восстановить картину происходящих операций в базах и понять, какая ветвь документов верная, а какая была заведена с ошибками.

В тоже время, в компании может создаваться по несколько десятков документов в минуту. При этом в течение дня все созданные документы должны быть синхронизированы между информационными базами. Это влечет за собой проблему обеспечения скорости конвертирования структур данных. Для нормальной работы компании необходимо выделить большое количество сотрудников, которые будут заниматься только этим процессом, при этом на их обучение также придется затратить некоторое время и ресурсы компании. А в связи с монотонностью работы и сложностью обрабатываемых данных, таких как числа, шанс возникновения ошибок велик.

1.2 Типовая система конвертирования

Типовая система конвертирования данных представляет собой полуавтоматизированную подсистему, служащую для организации конвертации данных между конфигурациями любой структуры и сложности [1]. Ее алгоритм отличается от ручного тем, что нет необходимости вручную заносить данные в информационные базы. Большую часть работы выполняет система в автоматическом режиме, участие пользователя требуется для запуска процесса конвертирования и контроля его правильной работы.

Система представляет собой отдельную информационную базу, которая подключается к используемой платформе «1С: Предприятие». При этом в ней задаются настройки конвертации, которые представляют собой пути до файлов с информационными базами, основные настройки процесса конвертации и логин и пароль пользователя, от чьего имени будут выполняться процессы. По наступлению установленного времени система производит загрузку всех данных, которые необходимо конвертировать, и переводит их все в формат XML. После успешной загрузки начинается процесс конвертации, в течение которого каждый загруженный в буфер документ трансформируется в документ другой информационной базы. После того, как буфер опустеет, т.е. все документы будут синхронизированы, происходит процесс выгрузки полученных документов из буфера сконвертированных данных. В течение этого процесса происходит отправка данных из буфера по информационным базам, которые были указаны как получатели конвертированных документов.

Одним из важных недостатков данного способа конвертирования является то, что на любом этапе выполнения процесса конвертации если возникает ошибка, то весь процесс отменяется и все документы, которые были уже сконвертированы и сохранены

в буфер теряются. В тоже время такой способ конвертации оказывает очень сильное влияние на работу серверной части компании, так как сам процесс требует больших вычислительных мощностей процессора и больших пропускных способностей оперативной памяти, шины обмена данными с жестким диском и локальной сети организации.

2. Сравнение алгоритмов конвертирования данных

2.1. Система критериев сравнения способов конвертирования

Для выбора подходящего алгоритма конвертирования структур данных между информационными базами необходимо создать некоторую систему критериев. Затем, согласно этим критериям, провести сравнение алгоритмов и выявить их достоинства и недостатки.

В качестве критериев будем рассматривать как те, что влияют на производительность системы, так и на ее безопасность.

Критерий 1. Максимальное количество документов.

При выборе алгоритма конвертирования данных одним из важнейших критериев является максимальное количество документов, с которыми может нормально функционировать алгоритм. Данное число обеспечивает гарантированную нормальную работу без учета появления возможных ошибок, связанных с другими нештатными ситуациями.

Критерий 2. Поддержка параллельной обработки данных

Данный критерий отображает возможность параллельного выполнения алгоритма. Другими словами, позволяет ли он одновременно обрабатывать сразу несколько документов, не нарушая процесс конвертирования структур данных каждого из них отдельно.

Критерий 3. Наличие механизмов расширения

Наличие механизмов расширения позволяют без выполнения длительных дополнительных работ по настройке алгоритмов конвертирования данных, не учитывая настройку дополнительной базы-участницы процесса, и без внесения изменений в программный код платформы увеличить количество баз, участвующих в конвертировании, или документов, проходящих процесс за один раз.

Критерий 4. Затрачиваемые ресурсы

Критерий, отображающий требуемые вычислительные или рабочие ресурсы, которые необходимы для выполнения конвертирования структур данных между информационными базами.

Критерий 5. Обработка ошибок

Данный критерий показывает, существует ли у алгоритма процесс обработки возникающих во время конвертирования данных ошибок и как решается проблема их появления при дальнейших конвертациях.

Критерий 6. Потеря данных в случае сбоя

Критерий указывает, какое количество данных будет бесследно утеряно при возникновении сбоя в работе алгоритма, и какое количество данных, если такое случится, можно будет восстановить.

Критерий 7. Отказоустойчивость

Обеспечивает ли алгоритм конвертирования данных отказоустойчивость, и на каком уровне.

Критерий 8. Обеспечение целостности информационных баз

Критерий, который показывает, обеспечивает ли алгоритм конвертирования структур данных целостность БД, которая сравнивается с их состояниями до момента запуска процесса синхронизации данных.

2.2. Обоснование выбора алгоритма конвертирования

Для удобства сравнения описанных выше алгоритмов конвертирования все результаты были занесены в таблицу ниже.

Таблица 1.

Результаты сравнения алгоритмов конвертации

Критерий	Ручной алгоритм	Типовой алгоритм
Максимальное количество документов	ограничено возможностями сотрудников	~ 2,5 млн.
Поддержка параллельной обработки	+/- (при наличии нескольких одновременно работающих сотрудников)	-
Наличие механизмов расширения	+ (дополнительные сотрудники)	+/- (ограничение на документы остается)
Затрачиваемые ресурсы	расходы на сотрудников	требует больших мощностей процессора, большой пропускной способности ОЗУ, ЖД и корпоративной сети
Обработка ошибок	ручная замена данных	переконвертирование выбранных документов
Потеря данных в случае сбоя	все данные, которые не были проведены в базах, теряются	все данные, находящиеся в буферах, теряются [2]
Отказоустойчивость	плохая (связана с человеческим фактором)	нормальная (требует правильной настройки)
Обеспечение целостности информационных баз	+/- (не гарантировано)	+ (при наличии правильной настройки)

Таким образом, данные сравнения алгоритмов показывают, что каждый из них имеет свои достоинства и недостатки. При этом ручной алгоритм демонстрирует преимущественно отрицательные стороны.

Для большей наглядности была построена диаграмма, отображающая распределение оценок критериев алгоритмов по осям координат, каждая из которых характеризует критерий. Чтобы получить количественную оценку степени соответствия

алгоритма конвертирования тому или иному критерию была введена характеристика оценки, которая показана в таблице ниже.

Таблица 2.

Характеристика оценок алгоритмов

Оценка критерия	Характеристика алгоритма
«0»	Не поддерживает условие
«1»	Поддерживает условие, но в определенных случаях
«2»	Поддерживает условие в любом случае при некоторых ограничениях
«3»	Полностью поддерживает условие

При оценке критериев в случаях, когда изменение параметров можно достигнуть простым путем, присваивалось «2» балла, а если изменения можно достичь, но общее ограничение остается – «1» балл.

Итоговая диаграмма, полученная после подсчета общих баллов, представлена на рисунке ниже.

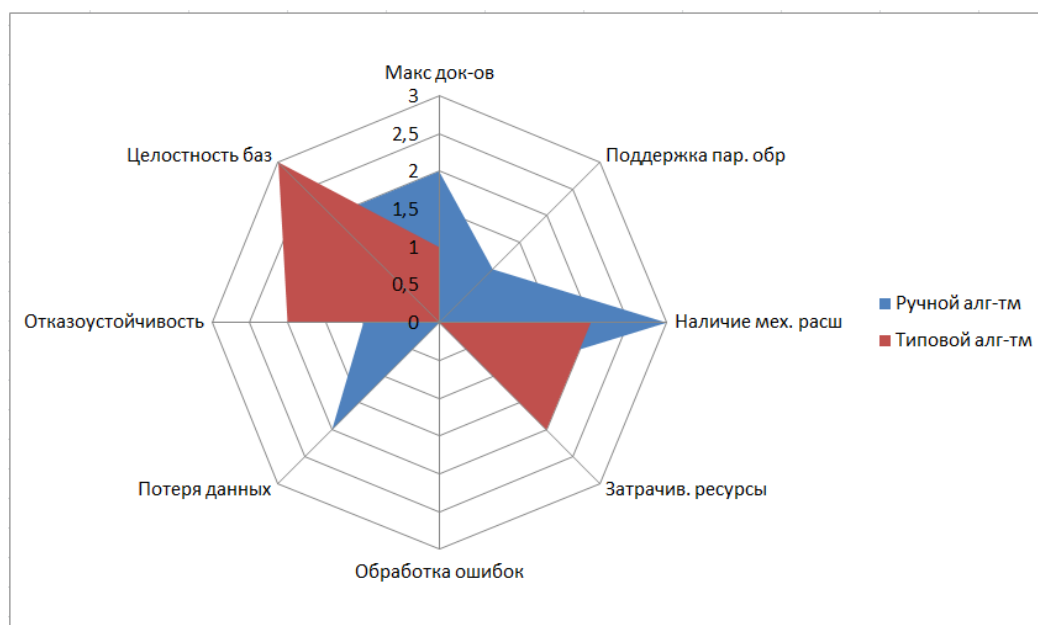


Рис. 1. Диаграмма сравнения алгоритмов

Преимущества ручного алгоритма заключаются в том, что:

- нет необходимости выполнять начальные настройки для развертывания процесса конвертирования данных;
- не нужно предусматривать мощное оборудование для выполнения синхронизации. Достаточно только обучить сотрудников, а это не займет много времени, выделить им персональные рабочие станции и запустить процесс.

Дополнительным плюсом ручного алгоритма является то, что в отличие от типового алгоритма конвертации теряется только те данные, которые не были сохранены в базу. Если ввести правило для сотрудников, что после создания документа его необходимо сразу провести и сохранить, то процент потерь сведется к минимуму.

Однако наряду с положительными сторонами алгоритма есть и отрицательные. Например, количество обрабатываемых документов ограничено возможностями сотрудников. Для обработки большого количества документов требуется большой штат работников, которые будут монотонно создавать новые документы в базах или исправлять текущие. А монотонная работа будет нести за собой и повышенный шанс опечаток и, соответственно, рассинхронизации данных в базах. Таким образом, может произойти нарушение целостности информационных баз, что допускать невозможно. Если на параллельную обработку данных можно не обращать внимания, так как главное, чтобы процесс заканчивался в срок, а параллельно или последовательно выполнялась конвертация – это дело второстепенное, то отсутствие простых механизмов расширения сказываются заметно. При резком возрастании объема обрабатываемых данных или появлении новой базы, алгоритм не будет позволять выполнять работу в срок и, следовательно, опять произойдет нарушение целостности. В одной базе будут использоваться новые данные, а в остальных они еще не будут обновлены, в связи с чем, отчеты будут ссылаться на старые данные. Таким образом, ручной алгоритм конвертирования структур данных не подходит.

Типовой алгоритм обеспечивает автоматический режим конвертации данных, однако тоже имеет существенные недостатки:

- требует больших вычислительных мощностей сервера компании;
- имеет ограничение на объем обрабатываемых данных, хотя у него и имеется возможность быстрой настройки процесса конвертирования с новой базой, однако общий объем данных должен не превосходить заданное значение.

В дополнение к этому, важной отрицательной стороной является полная потеря данных при возникновении сбоя. Даже в случае, когда все документы кроме последнего успешно пройдут конвертирование и добавятся в выходной буфер, а при обработке последнего документа возникнет ошибка, все данные будут утеряны. В итоге необходим повторный запуск процесса обработки данных.

Заключение

Сравнение двух существующих алгоритмов конвертирования позволяет сделать вывод, что для компаний, ежедневно обрабатывающих большое количество документов, необходимо использовать другой, новый алгоритм, который не будет иметь всех существенных недостатков рассмотренных алгоритмов.

Поэтому, следующим шагом будет разработка нового способа конвертирования структур данных, который будет реализован в виде подсистемы конвертирования. Она позволит обеспечить отсутствие недостатков, выявленных у двух используемых в настоящий момент алгоритмов, увеличенную скорость обработки данных, наличие механизмов обработки ошибок. Подсистему можно будет использовать в любой компании, которая применяет в своей деятельности информационные базы платформы «1С: Предприятие» и выполняет обмен данными между базами.

Список литературы

1. Организация конвертации данных и обмена данными с помощью конфигурации «Конвертация данных 2.0»: [Электронный ресурс]. Режим доступа: <https://its.1c.ru/db/metod8dev#content:2943:hdoc> (дата обращения: 2.02.2018)
2. Бояркин В.Э., Филатов А.И. 1С: Предприятие 8. Конвертация данных: обмен данными между прикладными решениями. М.: Изд-во 1С. 2008. 179 с.

Сравнение мирового и российского опыта противодействия отмывания денежных средств и финансового терроризма

³²Пугачев Е.К., ³³Хорунжина К.С.

МГТУ им. Н.Э. Баумана (НИУ), Москва, Россия

iub.pugachev@yandex.ru

horunzhina.kristina@yandex.ru

Аннотация: Под отмыванием денег подразумевают процесс легализации денежных средств или других материальных ценностей, которые были получены незаконным путем, в том числе вследствие совершения преступления. Легализируют эти ценности для того, чтобы в будущем иметь возможность открыто их использовать. Эта процедура помогает выводить деньги из теневой экономики в законную, тем самым обманывая государство. Причем разные способы отмывания денег оказывают разное влияние на экономику страны. В работе сравниваются законодательные акты России и Европы, и на их основе показаны основные плюсы и минусы двух разных подходов, стремящихся улучшить качества расследования отмывания денежных средств.

Ключевые слова: директива, регламент, денежная проводка, бенефициар, отмывание средств, финансовый терроризм, преступность, мониторинг, легализация

Comparison of the world and Russian experience of combating money laundering and financial terrorism

Pugachev E.K., Khorunzhina K.S.

Bauman Moscow State Technical University, Russia

iub.pugachev@yandex.ru

horunzhina.kristina@yandex.ru

Abstract: Money laundering involves the process of legalizing money or other material assets that have been obtained illegally, including as a result of the commission of a crime. Legalize these values in order to be able to openly use them in the future. This procedure helps to withdraw money from the shadow economy into the legal economy, thereby deceiving the state. Moreover, different methods of money laundering have different effects on the economy of the country. The paper compares the legislative acts of Russia and Europe, and on their basis shows the main pros and cons of two different approaches that seek to improve the quality of the investigation of money laundering.

³²Пугачев Евгений Константинович - кандидат технических наук, доцент кафедры "Компьютерные системы и сети" МГТУ им. Н.Э. Баумана.

³³Хорунжина Кристина Сергеевна – ассистент кафедры "Компьютерные системы и сети" МГТУ им. Н.Э. Баумана.

Keywords: directive, regulation, cash post, beneficiary, money laundering, financial terrorism, crime, monitoring, legalization.

Введение

В настоящее время цифровые технологии и развитие интернет услуг шагнули далеко вперед, особенно в банковской сфере. Осуществлять мониторинг банковских операций в масштабе международных транзакций становится гораздо сложнее, поэтому необходимо скоординировать усилия и максимально унифицировать регулирование денежных проводок.

Борьба с отмыванием денежных средств в каждой стране ведется по-своему и стала своеобразным витком политики, который предполагает:

- внедрение новых законов в правовую систему, которые поспособствуют противодействию преступности;
- формирование новых органов власти, как рычага борьбы с финансовыми махинациями;
- налаженный мониторинг финансовых институтов;
- международное сотрудничество в сфере противодействия отмыванию денежных средств.

В России чтобы осуществлять борьбу с незаконным оборотом капитала, компетентные органы используют такие инструменты, как:

- отчетность физических и юридических лиц с указанием их уровня дохода и его происхождения;
- периодические аудиты финансовых учреждений, включая частные фирмы;
- сотрудничество с местными регулирующими органами власти;
- сбор сведений третьих лиц;
- отчетность по всем банковским операциям в Росфинмониторинг.

Но все эти инструменты по-прежнему работают весьма неэффективно в России, и вся система противодействия отмывания денежных средств требует новых идей и нововведений.

1. Европейское законодательство по противодействию денежной преступности

В Европе примером регулирования стало введение Директивы ЕС 2015/849 о противодействии использованию финансовой системы для отмывания денег и финансирования терроризма и Регламент (ЕС)2015/847 об информации, сопровождающей переводы денежных средств.

Российское законодательство выстраивает схему борьбы с финансовыми преступлениями и отмыванием денег на внутригосударственном уровне, Европа же

разрабатывает регулирование, затрагивающее персональные данные и частную жизнь клиентов.

Такой подход нельзя осуждать, потому что глобальные проблемы терроризма и отмывания денежных доходов требуют комплексных мер и единого регулирования. Но не стоит забывать, что во всем должна присутствовать пропорциональность между защитой персональных данных и прозрачностью в бизнесе.

Для минимизации рисков отмывания денег важно понимать, кто реально контролирует управление в компании, занимает руководящие посты или владеет основным пакетом ее акций. В 2015 году Европейская комиссия руководствовалась именно этим принципом, выступая за создание единого национального реестра с данными о конечных бенефициарах с последующей передачей такой информации в единый общеевропейский реестр.

Бенефициарами признают лиц, владеющих долей в уставном капитале компании или 25% акций, именно такие люди могут влиять на деятельность компании, но значение процента может быть разным в каждой стране. Все это реализует принцип «знай своего клиента», а, следовательно, международные стандарты и лучшие коммерческие практики.

Согласно Директиве кредитные организации должны хранить следующие данные о своих клиентах:

- ФИО;
- дата и год рождения;
- гражданство;
- страна постоянного проживания;
- природа владения;
- размер доли в компании.

Необходимо проводить комплексную проверку в следующих случаях:

- а) если начинаете деловые отношения с организацией впервые;
- б) если заключаемая сделка превышает 10 000 евро, учитывая, как одноразовые переводы, так и связанные сделки;
- в) если денежная проводка превышает 2000 евро и связана с игорным бизнесом;
- г) если кредитная организация подозревается в совершении финансового терроризма или отмывании денежных средств;
- д) если ранее от клиентов поступали сведения, компрометирующие данную банковскую организацию.

Особенно важным нововведением считается комплексная клиентская проверка. Интересно, что если есть высокий риск или подозрения (подкрепленные обоснованными доводами) в отмывании денег, то проверка разрешена и с привлечением информации о иностранных клиентах (например, очень крупные сделки, не имеющие под собой экономических и законных оснований или сделки с участием влиятельных политических деятелей). Влиятельные политические лица (ВПЛ) - это физические лица, которые

выполняют или ранее выполняли какие-либо государственные функции (главы государств или правительств, политические деятели, судьи и сотрудники вооруженных сил, менеджеры крупных государственных компаний, представители крупных политических партий, а также члены семей названных лиц).

Одним словом, Директива не делает различия между клиентами европейских стран и лиц из третьих стран, тем самым обеспечивая всеобъемлющую клиентскую проверку в отношении влиятельных политических лиц.

Второй важный момент нововведений мониторинг информации, сопровождающей переводы средств (Регламент (ЕС) 2015/847). Регламент устанавливает мониторинг переводов, полученных или отправленных платежных сервисом, расположенным в Европе или европейской экономической зоне (ЕЭЗ). Требования распространяются на сопровождающую переводы информацию в рамках всех платежных систем, которые участвуют в цепочке переводов. Если перевод разовый (а не осуществляется на регулярной основе) и сумма перевода высока, то именно такие проводки должны проходить тщательные проверки и мониторинга.

Так же новый Регламент требует предоставления данных и о получателе денег, в отличие от старого регламента 2006 года, где проверки подвергалась информация только о плательщике.

Оператор платежной системы плательщика обязуется предоставлять при переводе денежных средств следующую информацию:

- 1) наименование плательщика;
- 2) номер счета плательщика;
- 3) адрес плательщика;
- 4) наименование получателя платежа;
- 5) номер счета получателя платежа.

Установление единых стандартов для расследования случаев финансового терроризма и мониторинга платежных операций по всему ЕС, включая ЕЭЗ, это главная цель установки таких порядков.

Но есть одна проблема. Передача данных должна обеспечиваться с соблюдением российского законодательства. Требования Регламента тесно связаны с передачей персональных данных (далее — ПДн), то есть провайдерам платежных систем необходимо передавать информацию о российских гражданах за рубеж.

2. Российское законодательство по противодействию денежной преступности и персональные данные

Обратимся к ч. 1 ст. 12 Федерального закона от 27.07.2006 № 152-ФЗ «О персональных данных», который регламентирует обязанность оператора убедиться, что страна, куда передаются персональные данные, обеспечивает достаточный уровень защиты этих данных, установленных в Российской Федерации. Необходимо помнить о существовании Конвенции по защите прав физических лиц при автоматизированной

обработке персональных данных от 28.01.1981. Но не все страны, входящие в ЕС, признают и ратифицируют этот документ.

Реестры бенефициаров вызывают еще больше вопросов с точки зрения защиты ПДн, а именно факт их публичности. Поэтому доступ к реестру бенефициарных владельцев могут получить только лица, демонстрирующие легитимный интерес. Но доступ к такому роду информации надо предоставлять в свободное пользование с осторожностью, то есть на индивидуальной основе, избегая рисков мошенничества, шантажа, возможностей похищения, насилия или запугивания ее владельцев. Персональные данные должны использоваться для достижения законных целей и по правилам, заранее оговоренным. В России это правило регламентируется п. 2 ст. 5 Федерального закона «О персональных данных». В Европе это называется принципом пропорциональности: собираются и обрабатываются только те данные, которые были необходимы для достижения заранее определенных целей обработки данных. При всей нужности таких реестров, которые будут способствовать выявлению подозрительной финансовой и незаконной деятельности, связанной с отмыванием денег, до конца не понятно, как законодательные акты разных стран будут взаимодействовать друг с другом на практике. Существует риск, что излишняя открытость приведет к оттоку денежных средств в другие страны, обеспечивающие хоть какой-то минимум конфиденциальности, а, следовательно, спровоцирует движение инвестиций за границу и налоговое обложение европейских стран.

На данный момент в Российской Федерации такого реестра нет. Согласно Федеральному закону от 07.08.2011 № 115-ФЗ «О противодействии легализации (отмыванию) доходов, полученных преступным путем, и финансированию терроризма» российские компании обязаны хранить, обновлять и сообщать информацию о конечных бенефициарных собственниках (с учетом нескольких исключений), а именно:

- 1) включать информацию о конечных бенефициарных собственниках в финансовую отчетность;
- 2) передавать такую информацию госорганам по их запросу.

Заключение

Подводя итог выше сказанного, видно, что европейские страны стремятся к централизованному ведению клиентской базы, в России же каждая компания ведет свой собственный реестр конечных бенефициаров, без отправки в единый реестр.

Вместе с тем российским предпринимателям необходимо учитывать требования европейского законодательства о предотвращении отмывания денег и финансировании терроризма, в особенности при регистрации компаний в европейских юрисдикциях. Мировые тенденции диктуют новые правила и одно из них максимальная прозрачность. Российское законодательство может пойти на поводу у Европейских стран или задать новый виток в политике урегулирования финансового терроризма, но ясно одно России необходимо перенимать «мировой опыт» и внедрять современные технологии в

свои инструменты мониторинга операций, тем самым улучшая законодательство в рамках противодействия отмывания денежных средств.

Список литературы

1. Директива Европейского Парламента и Совета Европейского союза от 20.05.2015 № 2015/849 «О предотвращении использования финансовой системы для целей отмывания денег или финансирования терроризма, об изменении Регламента (ЕС) 648/2012 Европейского Парламента и Совета ЕС и об отмене Директивы 2005/60/ЕС Европейского Парламента и Совета ЕС и Директивы 2006/70/ЕС Европейской комиссии».
2. Федеральный закон "О противодействии легализации (отмыванию) доходов, полученных преступным путем, и финансированию терроризма" от 07.08.2001 N 115-ФЗ
3. Евразийская группа по противодействию легализации преступных доходов и финансированию терроризма: [Электронный ресурс] // Официальный сайт ФАТФ. URL: <https://eurasiangroup.org/ru>. (Дата обращения: 18.03.2018)
4. Глааб С., Дель Е. Предписание (ЕС) 2015/847 об информации, сопровождающей переводы средств: новые требования к платежам // Международные банковские операции. 2017. № 2.
5. Delorenzi D. EU Anti-Money Laundering legal framework: the race has started again: [Электронный ресурс] // European Area of Freedom Security & Justice. URL: <https://free-group.eu/2015/09/22/eu-anti-money-laundering-legal-framework-the-race-has-started-again>. 22.09.2015. № 2.

Сравнительный анализ контейнеров библиотек Qt и STL по критерию времени

³⁴Гуренко В.В., ³⁵Букуша К.В., ³⁶Цаплин С.Т.

МГТУ им. Н.Э. Баумана (НИУ), Москва, Россия

wgurenko@bmstu.ru

kenbxitn@gmail.com

zap05@inbox.ru

Аннотация: Работа посвящена сравнительному анализу быстродействия контейнеров объектных библиотек Qt и STL. Получены экспериментальные результаты, отражающие время выполнения базовых контейнерных операций на различных структурах данных. Измерения проводились для контейнеров, аналогичных по назначению и внутренней структуре. На основе полученных данных сделаны выводы о предпочтительном использовании конкретных контейнеров для реализации основных операций над структурами данных. Результаты могут быть использованы программистами-разработчиками для создания эффективных по критерию времени программных продуктов с использованием контейнерной обработки в библиотеках Qt и STL.

Ключевые слова: объектная библиотека, контейнер, Qt, STL, структуры данных, время обработки.

Comparative analysis of containers of Qt and STL libraries by the time criterion

Gurenko V.V., Buksha K.V., Tsaplin S.T.

Bauman Moscow State Technical University, Moscow, Russia

wgurenko@bmstu.ru

kenbxitn@gmail.com

zap05@inbox.ru

Abstract: This article is dedicated to a comparative analysis of the performance of the containers of QT and STL object libraries. Experimental results were obtained that reflect the time of basic container operations on various data structures. Measurements were made for the containers similar in purpose and internal structure. Based on the data obtained, the preferred use of specific containers for the implementation of the basic operations on data

³⁴Гуренко Владимир Викторович – доцент кафедры "Компьютерные системы и сети" МГТУ им. Н.Э. Баумана, SPIN-код 2675-4796.

³⁵Букуша Кирилл Владимирович – студент бакалавриата кафедры "Компьютерные системы и сети" МГТУ им. Н.Э. Баумана.

³⁶Цаплин Сергей Тимофеевич – студент бакалавриата кафедры "Компьютерные системы и сети" МГТУ им. Н.Э. Баумана.

structures conclusions were made. The results can be used by software developers to create time-efficient software products while using container processing in the QT and STL libraries.

Keywords: object library, container, Qt, STL, data structure, processing time.

Введение

В настоящее время для эффективной разработки программных приложений нередко используются стандартные библиотеки, предоставляющие разнообразные встроенные средства разработки. Примерами таких часто используемых библиотек являются Qt и STL. Перед программистом нередко встает вопрос об обоснованном выборе инструментальных средств. В качестве критерия принятия решения недостаточно принять только функциональность той или иной библиотеки. Одним из основных показателей качества программного продукта является его быстродействие. Компонентами библиотек, вносящими существенную задержку в работу создаваемых с их помощью программных продуктов, являются встроенные в библиотеку контейнеры. Несмотря на схожие модели контейнерного хранения данных, реализованные в библиотеках, время выполнения базовых операций (поиск элемента, вставка элемента и прочих) может значительно различаться. Имеющиеся в литературе сведения о времени выполнения контейнерных операций неполны, и вопрос, в целом, изучен недостаточно [1–3].

Целью настоящей работы поставлен сравнительный анализ средств контейнерной обработки в указанных библиотеках по времени, затрачиваемому на различных операциях с различными структурами данных.

1. Сравнение контейнеров по времени обработки различных структур данных

Для осуществления сравнительного анализа контейнеров библиотек Qt и STL по критерию времени был проведен ряд экспериментальных исследований. Рассматривались одностипные операции как в той, так и в другой библиотеке на следующих структурах данных:

- динамический массив;
- список;
- бинарное дерево;
- хэш-таблица;
- множество.

Во внимание были приняты следующие основные операции:

- добавление в контейнер элемента определенной структуры данных;
- удаление элемента из контейнера;
- вставка элемента в контейнер;
- сортировка контейнера;

– поиск заданного элемента в контейнере.

В качестве данных, хранящихся в контейнерах, использовались программно сгенерированные псевдослучайные числа. Начальный объем числовой последовательности (далее N) был принят равным 1000. При последующих исследованиях значение N кратно увеличивалось до 1000000. Программный продукт, написанный для проведения эксперимента, содержит множество подпрограмм, каждая из которых измеряет время выполнения той или иной операции над контейнером и возвращает это время в основную программу. Для замера скорости работы подпрограммы использовался компонент стандартной библиотеки C «time.h», в частности, функция «clock», возвращающая количество временных тактов, прошедших с начала запуска программы [4, 5]. Для получения наиболее объективных результатов подпрограмма вызывалась несколько раз, и в таблицу заносилось усредненное время работы.

Для тестирования использовалась среда Qt Creator 4.6.0 с компилятором MinGW 5.3.0 32bit. Вычисления производились на компьютере под управлением ОС Windows 7 Ultimate, оснащенный процессором Intel Core i5–4670K 3.40GHz и оперативной памятью объемом 16 ГБ. В процессе получения экспериментальных данных выполнялось построение графических зависимостей времени выполнения основных операций от объема контейнера.

Проведем анализ полученных данных на основных контейнерных операциях.

Обработка динамических массивов. Контейнеры Qt QVector и STL vector реализованы как динамические массивы, элементы которых расположены в памяти последовательно, что обеспечивает высокую производительность и возможность индексного доступа. Для этих контейнеров замерялось время добавления/удаления, вставки в уже заполненную структуру, сортировки и поиска элементов. Основные результаты измерений представлены в табл. 1, а также наглядно отображены на рис. 1.

Таблица 1.

Время выполнения основных операций над динамическими массивами

Количество элементов	Время, с					
	10^3		10^5		10^6	
Библиотека	Qt	STL	Qt	STL	Qt	STL
Добавление/ Удаление	$5.20 * 10^{-5}$	$3.00 * 10^{-5}$	$4.80 * 10^{-3}$	$2.70 * 10^{-3}$	$4.90 * 10^{-2}$	$2.80 * 10^{-2}$
Вставка	$4.18 * 10^{-7}$	$4.19 * 10^{-7}$	$1.45 * 10^{-5}$	$1.39 * 10^{-5}$	$1.51 * 10^{-4}$	$1.52 * 10^{-4}$
Сортировка	$1.19 * 10^{-4}$	$1.65 * 10^{-4}$	$1.77 * 10^{-2}$	$2.70 * 10^{-2}$	$1.95 * 10^{-1}$	$2.98 * 10^{-1}$
Поиск	$2.10 * 10^{-6}$	$6.05 * 10^{-6}$	$6.38 * 10^{-5}$	$1.91 * 10^{-4}$	$6.80 * 10^{-5}$	$2.04 * 10^{-4}$

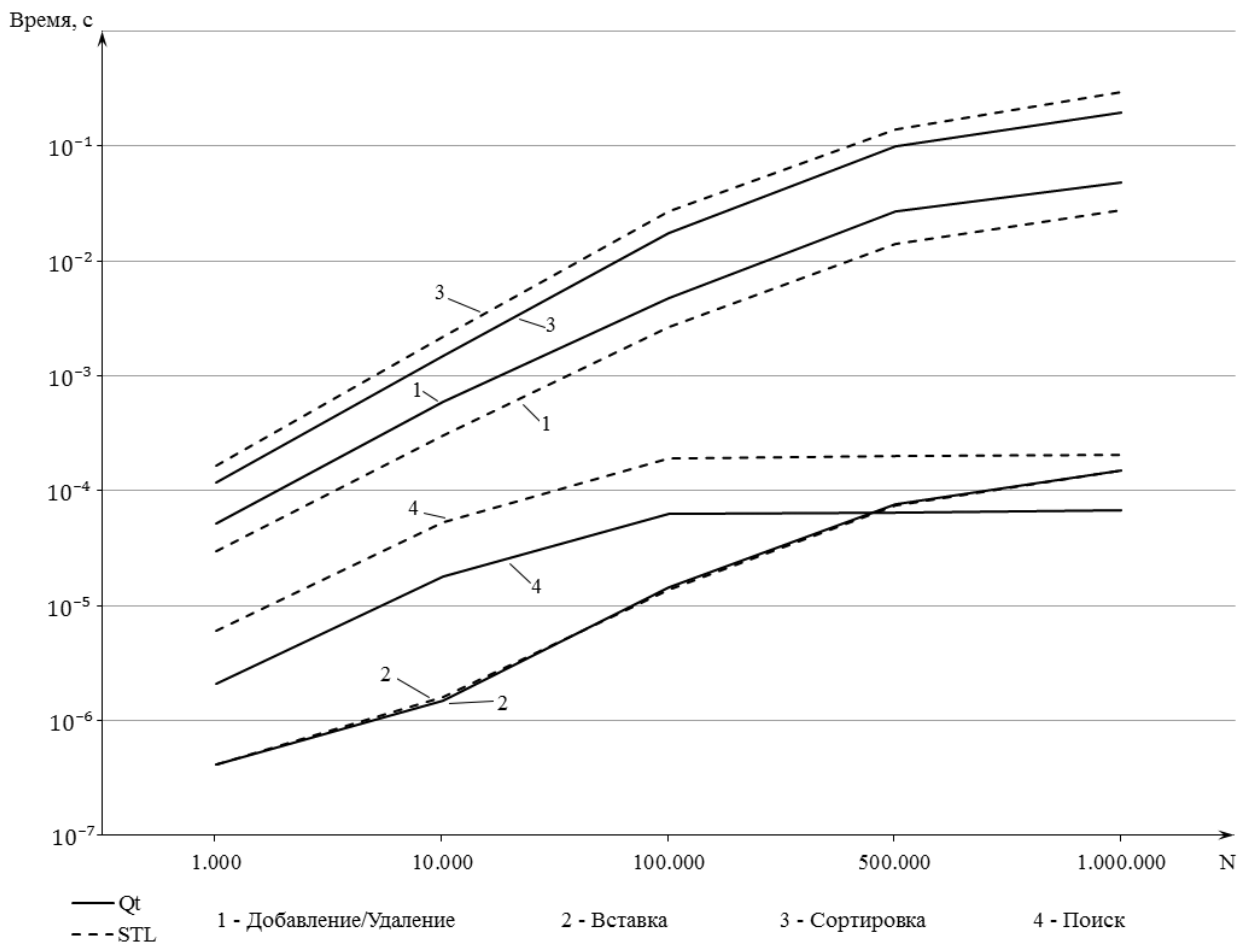


Рис. 1. Графики времени выполнения основных операций над динамическими массивами

Из таблицы 1 и рисунка 1 видно, что формирование динамического массива Qt Qvector происходит в среднем в два раза дольше, чем STL vector, но время сортировки и поиска элемента для контейнера Qt значительно меньше аналогичных операций для массива библиотеки STL. Так, для $N = 10^5$ имеем время сортировки для Qt QVector $1.77 \cdot 10^{-2}$ с, для STL vector $2.70 \cdot 10^{-2}$ с; время поиска – соответственно $6.38 \cdot 10^{-5}$ и $1.91 \cdot 10^{-4}$.

Обработка списков. Контейнеры Qt QList и STL list реализованы как двухсвязные списки и размещают данные в «куче». Эти контейнеры позволяют осуществлять быструю вставку элементов, однако не обладают возможностью обращения к элементам по индексам. Для них замерялось время добавления/удаления, вставки и поиска элементов. Полученные данные приведены в таблице 2, графическая интерпретация данных дана на рисунке 2.

Таблица 2.

Время выполнения основных операций над списками

Количество элементов, N	Время, с					
	10^3		10^5		10^6	
Библиотека	Qt	STL	Qt	STL	Qt	STL
Добавление/ Удаление	$1.27 * 10^{-4}$	$1.10 * 10^{-4}$	$1.38 * 10^{-2}$	$1.20 * 10^{-2}$	$1.32 * 10^{-1}$	$1.21 * 10^{-1}$
Вставка	$9.00 * 10^{-10}$	$1.00 * 10^{-9}$	$1.00 * 10^{-9}$	$1.00 * 10^{-9}$	$1.10 * 10^{-9}$	$1.10 * 10^{-9}$
Поиск	$5.79 * 10^{-6}$	$8.42 * 10^{-6}$	$1.88 * 10^{-4}$	$2.63 * 10^{-4}$	$1.97 * 10^{-4}$	$2.78 * 10^{-4}$

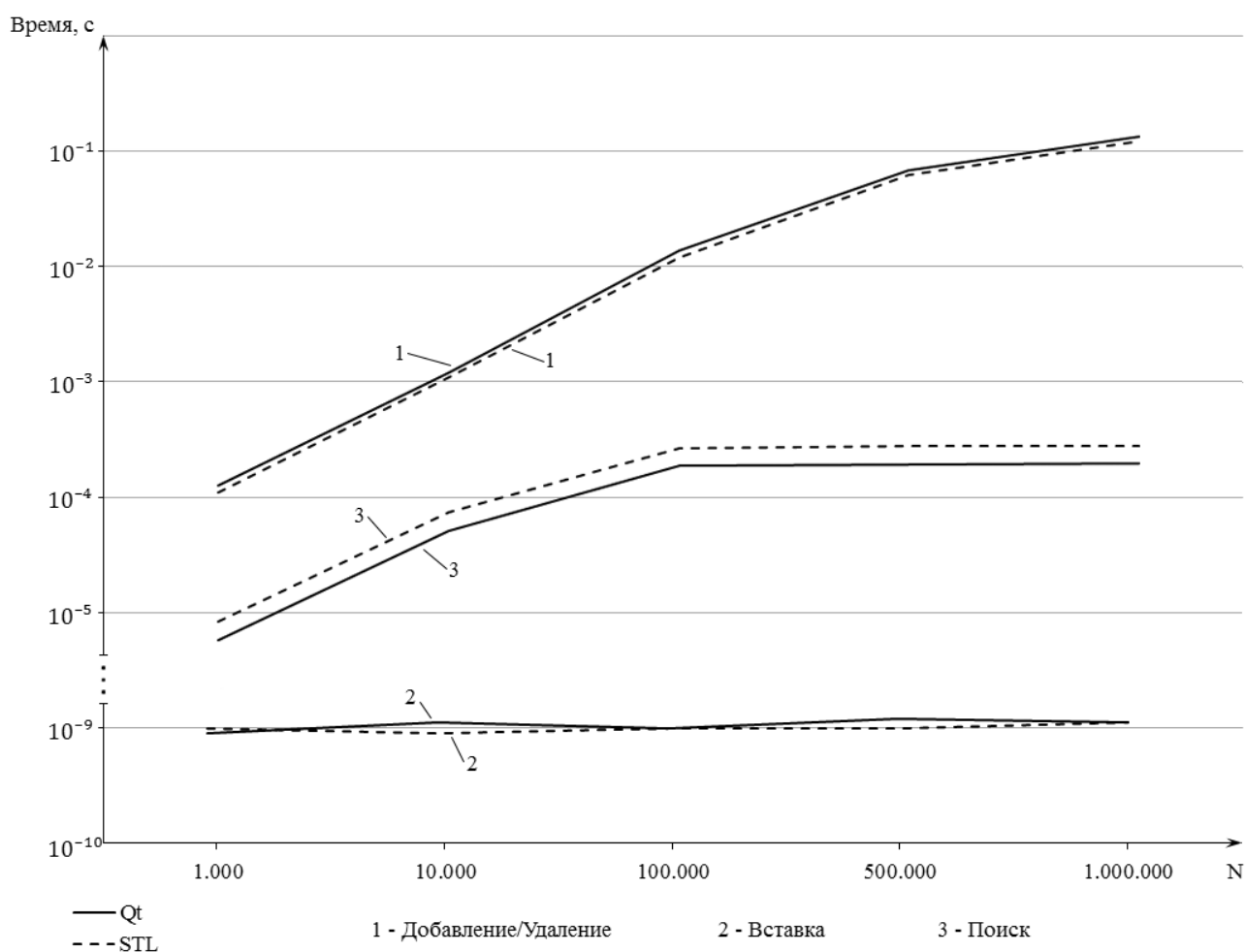


Рис. 2. Графики времени выполнения основных операций над списками

Полученные результаты свидетельствуют о том, что Qt QLinkedList затрачивает незначительно большее время для операций добавления и удаления. Так, при $N = 10^6$ разница времени обработки составляет менее 10%. При этом время выполнения поиска элементов в контейнере Qt значительно ниже, чем у STL list.

Обработка бинарных деревьев. Контейнеры Qt QMap, Qt QMultiMap, STL map и STL multimap реализованы как сбалансированные бинарные деревья. Основным отличием Qt QMultiMap от Qt QMap и STL multimap от STL map является то, что эти контейнеры позволяют хранить и обрабатывать несколько значений с одним ключом [6, 7]. Контейнер хранит свои данные в паре «ключ – значение». Так как деревья отсортированы по ключу, то поиск элемента, ассоциированного с ключом, осуществляется быстро. Для контейнеров замерялось время добавления/удаления и поиска элементов. Результаты измерений показаны в табл. 3. и на рис 3.

Таблица 3.

Время выполнения основных операций над деревьями

Qt QMap и STL map						
	Время, с					
Количество элементов	10^3		10^5		10^6	
Библиотека	Qt	STL	Qt	STL	Qt	STL
Добавление/ Удаление	$5.12 * 10^{-4}$	$8.98 * 10^{-4}$	$5.61 * 10^{-2}$	$1.12 * 10^{-1}$	$3.58 * 10^{-1}$	$7.10 * 10^{-1}$
Поиск	$1.54 * 10^{-7}$	$2.74 * 10^{-7}$	$2.82 * 10^{-7}$	$4.87 * 10^{-7}$	$3.05 * 10^{-7}$	$5.77 * 10^{-7}$
Qt QMultiMap и STL multimap						
	Время, с					
Количество элементов	10^3		10^5		10^6	
Библиотека	Qt	STL	Qt	STL	Qt	STL
Добавление/ Удаление	$5.50 * 10^{-4}$	$8.78 * 10^{-4}$	$9.09 * 10^{-2}$	$1.36 * 10^{-1}$	1.29	1.59
Поиск	$1.41 * 10^{-7}$	$2.81 * 10^{-7}$	$3.64 * 10^{-7}$	$6.13 * 10^{-7}$	$9.59 * 10^{-7}$	$1.16 * 10^{-6}$

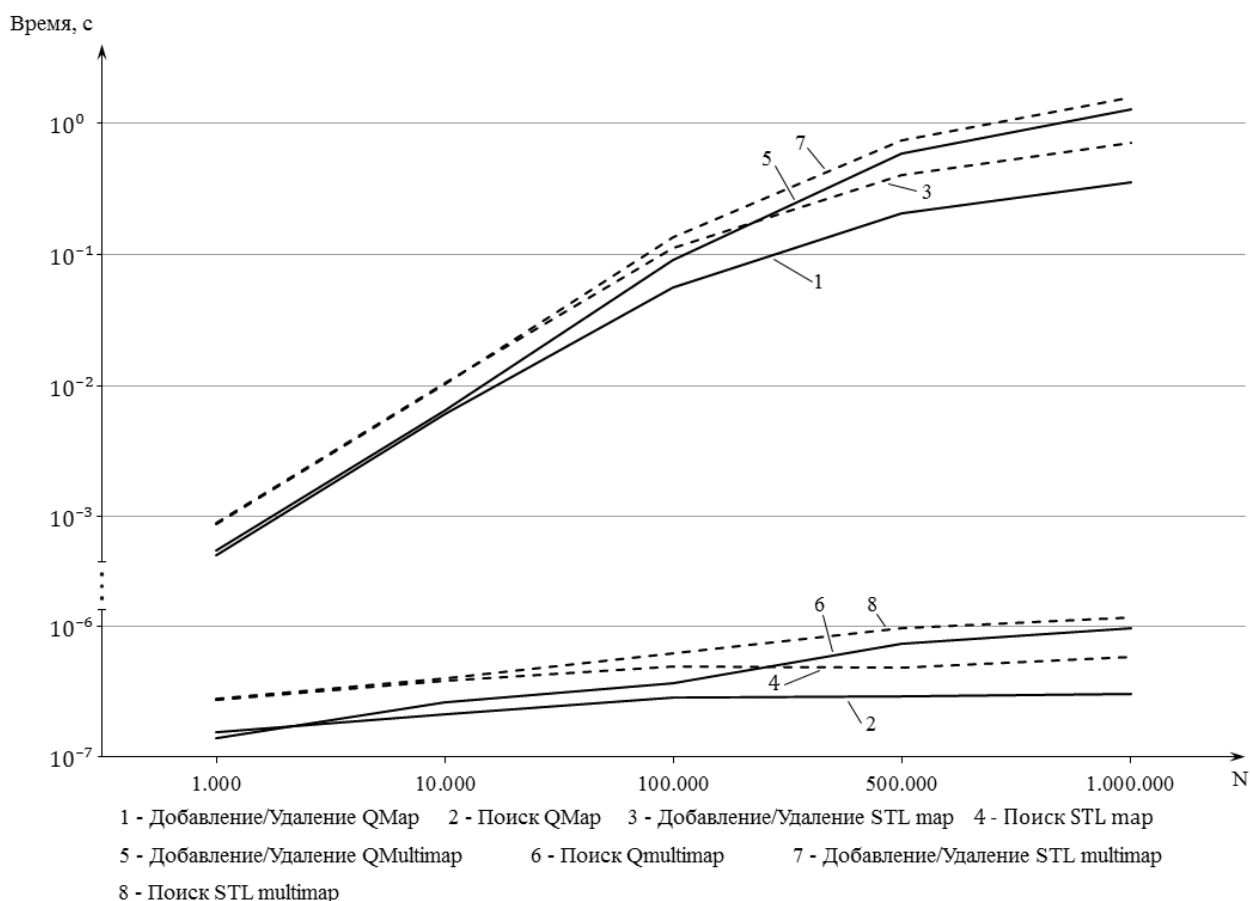


Рис. 3. Графики времени выполнения основных операций над деревьями

Qt QMap стабильно превосходит STL map во всем примененном диапазоне количества элементов. Разница во времени выполнения операций двукратная. Кроме того, следует отметить, что Qt QMap обладает более широким спектром встроенных функций для работы с элементами контейнера. Примерами могут служить методы QMap.values() и QMap.keys() [4]. Отличия в затраченном на операции времени у контейнеров Qt MultiMap и STL multimap не так значительны, причем с увеличением числа элементов контейнера разница становится менее заметной.

Обработка хэш-таблиц. Контейнеры Qt QHash, Qt QMultiHash, STL unordered_map и STL unordered_multimap реализованы в виде хэш-таблиц. Они обладают схожим функционалом с Qt QMap и STL map, однако обеспечивают более высокую скорость поиска элементов. Основным отличием Qt QMultiHash от Qt QHash и STL unordered_multimap от STL unordered_map является то, что эти контейнеры позволяют хранить и обрабатывать несколько элементов с одним значением ключа. Как и в предыдущих случаях, для этих контейнеров замерялось время добавления/удаления и поиска элементов. Результаты измерений представлены в табл. 4.

Таблица 4.

Время выполнения основных операций над хэш-таблицами

Qt QHash и STL unordered_map						
	Время, с					
Количество элементов	10^3		10^5		10^6	
Библиотека	Qt	STL	Qt	STL	Qt	STL
Добавление/ Удаление	$2.09 * 10^{-4}$	$5.80 * 10^{-4}$	$1.26 * 10^{-2}$	$4.56 * 10^{-2}$	$6.85 * 10^{-2}$	$3.75 * 10^{-1}$
Поиск	$5.50 * 10^{-8}$	$1.03 * 10^{-7}$	$4.10 * 10^{-8}$	$9.70 * 10^{-8}$	$4.30 * 10^{-8}$	$9.30 * 10^{-8}$
Qt QMultiHash и STL unordered_multimap						
	Время, с					
Количество элементов	10^3		10^5		10^6	
Библиотека	Qt	STL	Qt	STL	Qt	STL
Добавление/ Удаление	$2.10 * 10^{-4}$	$6.11 * 10^{-4}$	$2.45 * 10^{-2}$	$6.30 * 10^{-2}$	$3.21 * 10^{-1}$	$6.67 * 10^{-1}$
Поиск	$5.20 * 10^{-8}$	$1.08 * 10^{-7}$	$4.30 * 10^{-8}$	$1.20 * 10^{-7}$	$4.40 * 10^{-8}$	$1.16 * 10^{-7}$

Контейнеры библиотеки STL обладают большим временем обработки по основным операциям по сравнению с контейнерами библиотеки Qt. Возможность Qt QMultiHash и STL unordered_multimap хранить несколько элементов с одним ключом влечет за собой значительное увеличение затрачиваемого на обработку времени.

Обработка множеств. Контейнеры Qt QSet и STL unordered_set реализованы как множества элементов, причем повтор элементов не допускается. Для этих контейнеров вновь замерялось время добавления/удаления и поиска элементов. Результаты измерений представлены в табл. 5.

Таблица 5.

Время выполнения основных операций над множествами

	Время, с					
Количество элементов	10^3		10^5		10^6	
Библиотека	Qt	STL	Qt	STL	Qt	STL
Добавление/ Удаление	$2.08 * 10^{-4}$	$5.30 * 10^{-4}$	$1.33 * 10^{-2}$	$2.79 * 10^{-2}$	$7.52 * 10^{-2}$	$1.62 * 10^{-1}$
Поиск	$5.00 * 10^{-8}$	$9.40 * 10^{-8}$	$4.80 * 10^{-8}$	$8.00 * 10^{-8}$	$5.00 * 10^{-8}$	$7.90 * 10^{-8}$

Для каждой операции Qt QSet по скорости превосходит свой аналог почти в два раза. Так, уже при $N = 1000$ контейнер библиотеки Qt затрачивает на операцию добавления/удаления $2.08 \cdot 10^{-4}$ с, а STL unordered_set – $5.30 \cdot 10^{-4}$ с. Поиск элемента в множестве – достаточно распространённая операция, и такая большая разница во времени может оказаться критичной.

Данные из табл. 4 и 5 отражены на графиках рис. 4.

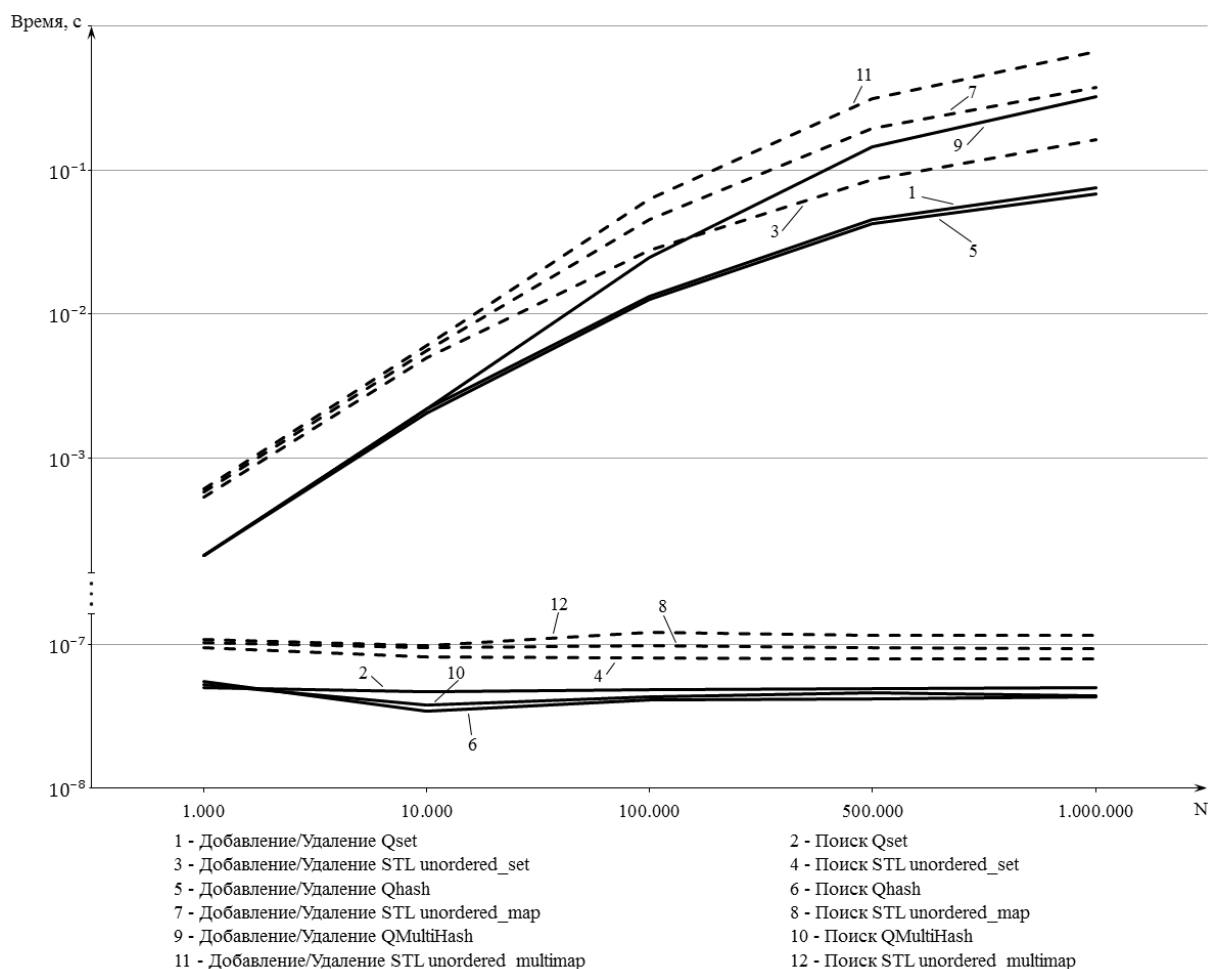


Рис. 4. Графики времени выполнения основных операций над деревьями

Из графиков рис. 4. видно, что время выполнения операции поиска для хэш-таблиц библиотеки Qt почти не отличается от времени выполнения той же операции над контейнером Qt QSet, реализующим множество. Контейнеры библиотеки STL требуют заметно большего времени на выполнение операций добавления/удаления. Например, при $N = 10^5$ время, затрачиваемое Qt QMultiHash, составляет $2.45 \cdot 10^{-2}$ с, а для STL unordered_multimap требуется $6.30 \cdot 10^{-2}$ с.

2. Итоговая качественная оценка контейнеров

Полученные данные позволяют качественно оценить каждую библиотеку по времени контейнерной обработки. Оценка выполнялась по трехбалльной шкале (+, -, =). Знаком «+» отмечены контейнеры, для которых указанные операции выполняются за

меньшее время, знаком «→» – за большее, а знаком «≈» – за примерно одинаковое время. Результаты сравнения представлены в табл. 6.

Таблица 6.

Качественный анализ результатов экспериментов

Контейнер		Операция				
Структура данных	Наименование	Добавление/ Удаление	Поиск	Вставка	Сортировка	
Динамический массив	Qt QVector	–	+	=	+	
	STL vector	+	–		–	
Связный список	Qt LinkedList	–	+	=	исследования не проводились	
	STL list	+	–			–
Бинарное дерево	Qt Map	+	+	исследования не проводились		
	STL map	–	–			
	Qt MultiMap	+	+			
	STL multimap	–	–			
Хэш-таблица	Qt QHash	+	+			исследования не проводились
	STL unordered_map	–	–			
	Qt QMultiHash	+	+			
	STL unordered_multimap	–	–			
Множество	Qt Set	+	+		исследования не проводились	
	STL unordered_set	–	–			

Анализ данных табл. 6 показывает, что контейнеры библиотеки Qt в целом получили более высокую оценку по всем операциям, чем их аналоги из библиотеки STL. Так, контейнеры Qt получили в итоге 13 «плюсов», а контейнеры STL только 2. Исключение составляют контейнеры QVector и QLinkedList, для которых операция добавления/удаления занимает большее время, чем для STL vector и STL list соответственно.

Заключение

Проведенный сравнительный анализ показал, что исследованные контейнеры имеют следующие предпочтительные области применения в зависимости от выполняемых на них операций.

Для операций добавления/удаления применительно к динамическим массивам и спискам предпочтительно использовать контейнеры STL vector и STL list. На операциях поиска элемента все исследованные контейнеры Qt показали лучшие результаты, чем их аналоги из библиотеки STL. Для операций вставки элементов контейнеры обеих библиотек затратили практически одинаковое время обработки. На операциях сортировки контейнер Qt QVector более предпочтителен, чем его аналог STL vector.

Использование контейнеров Qt на операциях поиска при общем количестве элементов, превышающем 100000, заметно выгоднее. На малых ($N < 1000$) объемах данных исследованные контейнеры по критерию времени почти равноценны.

Сделанные выводы могут быть учтены при разработке сложных программных продуктов с использованием рассмотренных библиотек. В дальнейшем авторы планируют продолжить исследования в направлении восполнения и анализа недостающих данных по времени контейнерных операций для других часто используемых библиотек, а также исследования возможностей и параметров параллельной контейнерной обработки данных.

Список литературы

1. Mutz M. Understand the Qt Containers. Режим доступа: <https://marcmutz.wordpress.com/effective-qt/containers/> (дата обращения 13.05.2018).
2. Сравнение скорости работы range-based for, foreach(Qt) и STL при подсчете суммы элементов контейнеров. Режим доступа: <https://habr.com/post/144905/> (дата обращения 13.05.2018).
3. Анализ производительности std::vector, std::list & std::deque. Режим доступа: <http://microfork.com/stdvector-stdlist-stdqueue-preformance-analisys/> (дата обращения 13.05.2018).
4. Время и работа с ним. Режим доступа: <http://www.excode.ru/art6061.html> (дата обращения 13.05.2018).
5. P.J. Plauger. Standard C++ Library Reference. Режим доступа: https://www.ibm.com/support/knowledgecenter/en/ssw_ibm_i_61/books/sc094949.pdf (дата обращения 13.05.2018).
6. Шлее М. Qt 5.3. Профессиональное программирование на C++. СПб.: БХВ-Петербург. 2015. 928 с.
7. Qt Documentation: Container Classes. Режим доступа: <http://doc.qt.io/qt-5/containers.html> (дата обращения 13.05.2018).

Отпечатано в типографии ООО «Первая оперативная типография»
Россия, Москва, 109457, ул. Зеленодольская, д.3, кв.28
Тел.: (495) 668-04-30, (495) 994-72-71;
<http://24print.ru/>; e-mail: print@24print.ru

Подписано в печать 10.07.2018. Заказ № 00000
Формат 60×84/16. Гарнитура Times New Roman.
Усл. печ. л. 8,1. Тираж 150 экз.