

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
ДОНЕЦКОЙ НАРОДНОЙ РЕСПУБЛИКИ**



ИНФОРМАТИКА И КИБЕРНЕТИКА

1 (31)

Донецк – 2023

УДК 004.3+004.9+004.2+51.7+519.6+519.7

**ИНФОРМАТИКА И КИБЕРНЕТИКА, № 1 (31), 2023,
Донецк, ДонНТУ.**

Выпуск подготовлен по результатам научно-технической деятельности аспирантов, соискателей и научных работников.

Статьи посвящены вопросам приоритетных направлений научно-технического обеспечения в области информатики, кибернетики, вычислительной техники и инженерного образования.

Материалы предназначены для специалистов народного хозяйства, ученых, преподавателей, аспирантов и студентов высших учебных заведений.

Редакционная коллегия

Главный редактор: Павлыш В. Н., д.т.н., проф.

Зам. глав. ред.: Мальчева Р. В., к.т.н., доц.

Ответственный секретарь: Лёвкина А. И.

Члены редакционной коллегии: Аверин Г. В., д.т.н., проф.; Аноприенко А. Я., к.т.н., проф.; Звягинцева А. В., д.т.н., доц.; Зори С. А., д.т.н., доц.; Карабчевский В. В., к.т.н., доц.; Привалов М. В., к.т.н., доц.; Скобцов Ю. А., д.т.н., проф.; Федяев О. И., к.т.н., доц.; Шелепов В. Ю., д.ф-м.н., проф.

Рекомендовано к печати ученым советом ГОУ ВПО «Донецкий национальный технический университет» Министерства образования и науки ДНР. Протокол № 2 от 31 марта 2023 г.

Свидетельство о регистрации СМИ: серия ААА № 000145 от 20.06.2017.

Приказ МОН ДНР № 135 от 01.02.2019 о включении в Перечень рецензируемых научных изданий ВАК ДНР.

Контактный адрес редакции

ДНР, 283001, г. Донецк, ул. Артема, 58, ГОУ ВПО «ДонНТУ»,

4-й учебный корпус, к. 36., ул. Кобозева, 17.

Тел.: +7 (856) 301-07-35, +7 (949) 334-89-11

Эл. почта: infcyb.donntu@yandex.ru

Интернет: <http://infcyb.donntu.ru>

СОДЕРЖАНИЕ

Информатика и вычислительная техника

Разработка приложения для распознавания речи <i>Баляба Я.В., Рычка О.В.</i>	5
Нейросетевая модель автоматизированного перевода <i>Васяева Т.А., Золушкин Ю.А., Мартыненко Т.В., Шуватова Е.А.</i>	12
Вывод древовидной вспомогательной грамматики как инструмент автоматического обобщения алгоритмов <i>Воробьев Л. О., Григорьев А. В.</i>	17
Проектирование архитектуры системы электронных денег <i>Мальцева Р. В., Терещенко К. А.</i>	23
Синтез и моделирование управления нестационарными участками шахтной вентиляционной сети <i>Чередникова О. Ю., Достлев Ю. С.</i>	30
Обзор методов трехмерной реконструкции <i>Якубовская Ю. А.</i>	36
Высокопроизводительный рендеринг трехмерных изображений для систем объемной пространственной визуализации <i>Зори С. А.</i>	42
Анализ эффективности модифицированного алгоритма измерения физических величин для сенсорных систем <i>Койбаш А. А.</i>	50

Инженерное образование

Особенности преподавания программирования в высшем учебном заведении <i>Стрижко М. А., Приходченко Е. И.</i>	59
<u>Об авторах</u>	62
<u>Требования к статьям, направляемым в редакцию научного журнала «Информатика и кибернетика»</u>	64

УДК 621.3.078.4+519.876.5

Разработка приложения для распознавания речи

Я. В. Баляба, О. В. Рычка

Донецкий национальный технический университет, г. Донецк

e-mail: y.baliaba22@gmail.com

Аннотация

Статья посвящена технологии распознавания речи в реальном времени и ее применению в различных сферах жизни и деятельности. Рассмотрены основные подходы, методы и факторы, которые влияют на качество и скорость распознавания речи. Показано, как можно создать тестовое приложение для распознавания речи на основе Microsoft Speech API, которое позволяет пользователю диктовать команды на русском языке и видеть их текстовое представление на экране. В статье описаны основные функциональные требования к приложению. Рассмотрены способы улучшения качества распознавания речи с помощью модели адаптации, которая позволяет увеличить вероятность распознавания определенных слов или фраз. Приведен пример интерфейса системы.

Введение

Распознавание речи в реальном времени - это процесс преобразования звукового сигнала в текстовый формат по мере его поступления. Это позволяет общаться с компьютерами, голосовыми помощниками, переводчиками и другими приложениями с помощью естественного языка. Распознавание речи в реальном времени имеет множество преимуществ для разных категорий пользователей:

Для людей с ограниченными возможностями распознавание речи в реальном времени может облегчить доступ к информации, образованию и развлечениям. Например, слабослышащие люди могут использовать распознавание речи для получения субтитров к фильмам или телефонным разговорам. Слепые люди могут использовать распознавание речи для управления своими устройствами или чтения текста.

Для бизнеса распознавание речи в реальном времени может повысить эффективность, качество работы и удовлетворенность клиентов. Например, колл-центры могут использовать распознавание речи для автоматизации обработки звонков, анализа настроения клиентов и контроля качества обслуживания. Маркетологи могут использовать распознавание речи для создания уникального голоса бренда и персонализации коммуникации с потребителями.

Для образования и науки распознавание речи в реальном времени может способствовать развитию навыков и знаний. Например, студенты могут использовать распознавание речи для диктовки лекций, записи заметок или изучения

иностранных языков. Ученые могут использовать распознавание речи для транскрибирования интервью, экспериментов или конференций.

Распознавание речи в реальном времени основано на сложных алгоритмах машинного обучения, которые анализируют звуковой сигнал на разных уровнях: от отдельных звуков до слов и фраз. Существуют разные подходы к распознаванию речи, такие как скрытые марковские модели, нейронные сети или комбинации этих методов. Для улучшения качества распознавания речи необходимо учитывать разные факторы, такие как шум, акцент, скорость или контекст говорящего.

Целью работы является разработка доступной и удобной системы для распознавания речи.

Обзор существующих систем распознавания речи

В настоящее время на рынке существует множество систем распознавания речи, которые предлагают различные функции и возможности для пользователей. Некоторые из них являются общего назначения, а некоторые специализируются на определенных областях или языках. В этом разделе мы рассмотрим некоторые из наиболее популярных и известных систем распознавания речи и сравним их по следующим критериям: точность, скорость, удобство использования, стоимость и доступность. В таблице 1 приведём характеристику каждой системы.

Таблица 1 – Сравнительная характеристика

Система	Google Speech-to-Text	Microsoft Speech Services	IBM Watson Speech to Text	Яндекс.Распознавание	Amazon Transcribe
Количество поддерживаемых языков и диалектов	Более 120	Более 80	Более 10	русский, английский, турецкий и украинский	Более 30
Используемая технология	Нейронные сети	Нейронные сети	Нейронные сети	Нейронные сети	Нейронные сети
Учет контекста и семантики	Да	Да	Да	Да	Да
Распознавание эмоций и тональности	Нет	Да	Нет	Нет	Нет
Распознавание специфических терминов и жаргона	Нет	Нет	Да	Нет	Да
Режимы работы	Реального времени или с предварительно записанными файлами	Реального времени или с предварительно записанными файлами	Реального времени или с предварительно записанными файлами	Реального времени или с предварительно записанными файлами	Реального времени или с предварительно записанными файлами
Опции для персонализации настройки	Различные	Различные	Различные	Различные	Различные
Доступность	Облачный сервис или локальное приложение	Облачный сервис или локальное приложение	Облачный сервис или локальное приложение	Облачный сервис или локальное приложение	Облачный сервис или локальное приложение
Стоимость	Бесплатна для небольших объемов данных, но требует оплаты за большие объемы или дополнительные функции [1]	Бесплатна для небольших объемов данных, но требует оплаты за большие объемы или дополнительные функции [2]	Бесплатна для небольших объемов данных, но требует оплаты за большие объемы или дополнительные функции [3]	Бесплатна для небольших объемов данных, но требует оплаты за большие объемы или дополнительные функции	Бесплатна для небольших объемов данных, но требует оплаты за большие объемы или дополнительные функции

Из сравнения этих систем распознавания речи можно сделать следующие выводы:

Все системы используют нейронные сети для анализа звукового сигнала и преобразования его в текст, что обеспечивает высокую точность и скорость распознавания.

Все системы учитывают контекст и семантику речи, что повышает их понимание и адаптацию к разным ситуациям и темам.

Все системы поддерживают разные языки и диалекты, что расширяет их применимость к разным группам пользователей.

Все системы предлагают различные опции для персонализации и настройки системы под конкретные нужды пользователя, что увеличивает их удобство использования.

Все системы доступны как облачный сервис или как локальное приложение, что обеспечивает безопасность и доступность данных.

Все системы бесплатны для небольших объемов данных, но требуют оплаты за большие объемы или дополнительные функции. Однако существуют и некоторые различия между этими системами, которые могут влиять на выбор пользователя в зависимости от его целей и предпочтений. Например, Google Speech-to-Text имеет самый широкий выбор языков и диалектов, но также имеет самые высокие цены за большие объемы данных. Microsoft Speech Services имеет самую продвинутую функцию распознавания эмоций и тональности речи, но также имеет самые строгие ограничения на размер и длительность файлов. IBM Watson Speech to Text имеет самую глубокую специализацию по отраслям и областям знаний, но также имеет самую низкую скорость распознавания. Яндекс.Распознавание имеет самую высокую точность для русского языка, но также имеет самую ограниченную поддержку других языков. Amazon Transcribe имеет самую гибкую настройку и персонализацию системы, но также имеет самую сложную интеграцию с другими сервисами.

Таким образом, можно сказать, что каждая из рассмотренных систем распознавания речи имеет свои сильные и слабые стороны, а выбор оптимальной системы зависит от конкретных потребностей и задач пользователя.

Методология разработки системы распознавания речи

В этом разделе мы изложим наш подход к созданию системы распознавания речи на C#, которая удовлетворяет поставленной цели. Мы будем использовать следующие технологии, алгоритмы, модели и инструменты:

.NET 5 SDK: это современный и универсальный фреймворк для разработки приложений на C# и других языках. Он обладает

множеством библиотек и инструментов для работы с данными, машинным обучением и обработкой сигналов. Мы будем использовать .NET 5 SDK для написания кода нашей системы и для взаимодействия с другими сервисами и приложениями [4].

Cognitive Services Speech SDK: это один из самых популярных и мощных SDK для работы с речью от Microsoft. Он позволяет легко интегрировать функции распознавания речи, синтеза речи, перевода речи и понимания речи в приложения на C# и других языках. Мы будем использовать Cognitive Services Speech SDK для преобразования звукового сигнала в текст и для определения намерений говорящего [5].

Language Understanding (LUIS): это сервис от Microsoft, который позволяет создавать и обучать модели понимания речи на основе нейронных сетей. Он поддерживает несколько языков и диалектов, а также способен распознавать разговорную речь, акценты и диалекты. Мы будем использовать LUIS для определения намерений говорящего на основе текста, полученного от Cognitive Services Speech SDK [6].

Для иллюстрации нашей методологии мы приведем следующие диаграммы:

Диаграмма потока данных на рисунке 1: она показывает, как наши данные проходят через различные этапы предобработки, обучения и тестирования нашей системы.

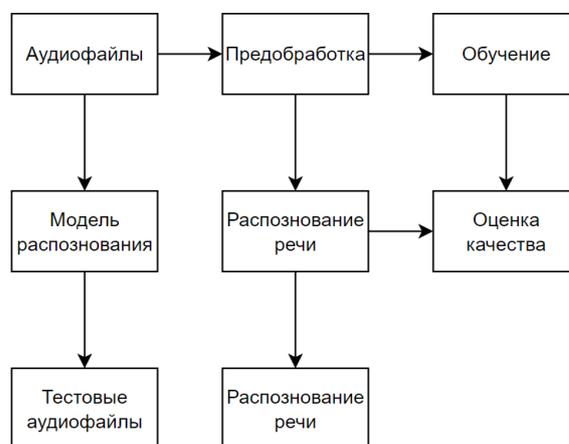


Рисунок 1 – Диаграмма потока данных

Диаграмма архитектуры системы на рисунке 2: она показывает, как наша система распознавания речи состоит из нескольких компонентов и служб, которые выполняют различные функции.

Диаграмма интерфейса системы на рисунке 3: она показывает, как пользователь может взаимодействовать с нашей системой через консольный или графический интерфейс.

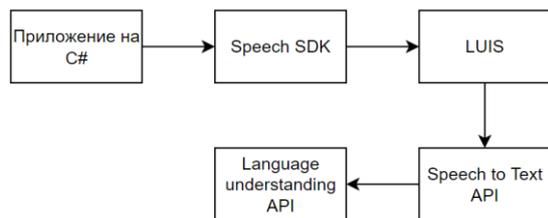


Рисунок 2 – Диаграмма архитектуры системы

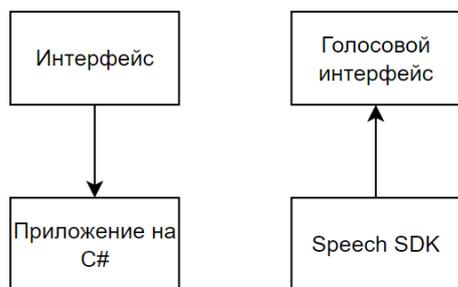


Рисунок 3 – Диаграмма интерфейса системы

Алгоритмы распознавания речи

Для того, чтобы преобразовать звуковой сигнал в текст, наша система использует несколько алгоритмов, которые выполняют различные этапы обработки и анализа речи. В общем виде, эти этапы можно представить следующим образом:

Анализ сигнала

На этом этапе мы применяем различные методы обработки сигналов, такие как фильтрация, усиление, подавление шума и эха, для того, чтобы улучшить качество и четкость звукового сигнала. Это необходимо для того, чтобы уменьшить влияние фоновых шумов и искажений на результат распознавания речи. Для этого мы используем такие алгоритмы, как:

Адаптивный фильтр Винера: это алгоритм, который позволяет подавлять шум в звуковом сигнале на основе статистических свойств сигнала и шума. Он адаптируется к изменению характеристик шума и сигнала во времени и частоте [7].

Алгоритм спектрального вычитания: это алгоритм, который позволяет подавлять шум в звуковом сигнале на основе разности спектров сигнала и шума. Он работает в частотной области и требует оценки спектра шума [8].

Алгоритм эхокомпенсации: это алгоритм, который позволяет подавлять эхо в звуковом сигнале на основе моделирования импульсной характеристики канала передачи. Он работает во временной области и требует адаптации к изменению условий канала [9].

Преимущества анализа сигнала:

- улучшение качества и четкости звукового сигнала
- уменьшение влияния фоновых шумов и искажений на результат распознавания речи

- увеличение точности и надежности распознавания речи

Извлечение признаков

На этом этапе мы преобразуем звуковой сигнал в набор численных характеристик, которые отражают его спектральные и временные свойства. Это необходимо для того, чтобы упростить задачу распознавания речи и выделить наиболее важную информацию из сигнала. Для этого мы используем такие методы, как:

Оконное преобразование Фурье: это метод, который позволяет перевести звуковой сигнал из временной области в частотную область. Он разбивает сигнал на короткие фрагменты (окна) и применяет к ним дискретное преобразование Фурье. Это позволяет получить спектрограмму сигнала - двумерное представление его частотного состава в зависимости от времени [10].

Мел-частотное кепстральное преобразование: это метод, который позволяет получить набор коэффициентов, которые характеризуют форму спектра звукового сигнала. Он применяет к спектрограмме сигнала следующие шаги: логарифмирование амплитуды спектра, применение мел-фильтров для учета нелинейного восприятия частот человеком, обратное дискретное косинусное преобразование для получения кепстра - спектра логарифма спектра. Коэффициенты кепстра называются мел-частотными кепстральными коэффициентами (MFCC) и являются одним из самых распространенных признаков для распознавания речи.

Линейный предсказательный анализ: это метод, который позволяет получить набор коэффициентов, которые характеризуют способность звукового сигнала предсказывать свои будущие значения на основе своих прошлых значений. Он основан на модели авторегрессии, которая представляет сигнал как линейную комбинацию своих предыдущих отсчетов и ошибки предсказания. Коэффициенты авторегрессии называются линейными предсказательными коэффициентами (LPC) и также используются для распознавания речи.

Преимущества извлечения признаков:

- упрощение задачи распознавания речи и выделение наиболее важной информации из сигнала
- уменьшение размерности данных и снижение вычислительной сложности
- учет специфики человеческого восприятия речи и повышение точности распознавания

Распознавание речи: на этом этапе мы сопоставляем набор признаков с набором слов или фраз, которые имеют наибольшую вероятность быть произнесенными. Для этого мы

используем такие методы, как динамическое программирование, скрытые модели Маркова и нейронные сети.

Понимание речи: на этом этапе мы определяем намерение говорящего на основе текста, полученного от распознавания речи. Для этого мы используем такие методы, как анализ естественного языка, извлечение сущностей и классификация намерений.

Распознавание речи в реальном времени основано на сложных алгоритмах машинного обучения, которые анализируют звуковой сигнал на разных уровнях: от отдельных звуков до слов и фраз. Существуют разные подходы к распознаванию речи, такие как скрытые марковские модели, нейронные сети или комбинации этих методов. Для улучшения качества распознавания речи необходимо учитывать разные факторы, такие как шум, акцент, скорость или контекст говорящего.

Скрытые марковские модели (НММ) - это один из наиболее распространенных подходов к распознаванию речи. Они основаны на статистической модели, которая описывает вероятность перехода между скрытыми состояниями и вероятность наблюдения определенных признаков в каждом состоянии. Скрытые состояния соответствуют фонемам - минимальным звуковым единицам языка, а признаки - это численные характеристики звукового сигнала. Для распознавания речи используются специальные виды НММ, такие как трифонные модели или модели с пропусками. Они позволяют учитывать контекст и вариативность произношения.

Нейронные сети (NN) - это другой подход к распознаванию речи, который стал популярным в последние годы. Они основаны на математической модели, которая имитирует работу биологических нейронов и их связей. Нейронные сети способны обучаться на больших объемах данных и автоматически извлекать признаки из звукового сигнала. Для распознавания речи используются разные виды NN, такие как сверточные нейронные сети (CNN), рекуррентные нейронные сети (RNN) или трансформеры. Они позволяют учитывать сложную структуру и динамику речевого сигнала.

Комбинации НММ и NN - это третий подход к распознаванию речи, который объединяет преимущества обоих методов. Он заключается в том, что NN используются для оценки вероятностей наблюдения признаков в НММ или для генерации признаков для НММ. Такой подход позволяет повысить точность и скорость распознавания речи за счет лучшего моделирования звукового сигнала и его статистических свойств.

Для улучшения качества распознавания

речи необходимо учитывать разные факторы, которые могут влиять на произношение и понимание речи. Например:

1. Шум. Шум может мешать распознаванию речи, так как он затрудняет выделение целевого голоса из фоновых звуков. Шум может быть стационарным, например, шум от кондиционера или двигателя, или нестационарным, например, шум от других говорящих или музыки. Чем больше шума и чем ближе он по спектру к целевой речи, тем сложнее распознать речь.

2. Акцент. Акцент может влиять на распознавание речи, так как он отражает различия в произношении звуков, ударениях и интонации между разными говорящими. Акцент может быть связан с родным языком говорящего, его регионом или социальной группой. Чем больше акцент отличается от стандартного или ожидаемого произношения, тем сложнее распознать речь.

3. Скорость. Скорость может влиять на распознавание речи, так как она определяет длительность и четкость звуков и слов. Скорость может варьироваться в зависимости от эмоций, цели или стиля говорящего. Чем быстрее говорит человек, тем сложнее распознать речь, так как звуки могут сливаться или теряться.

4. Контекст. Контекст может влиять на распознавание речи, так как он помогает понять смысл и значение речи. Контекст может быть связан с темой, ситуацией или задачей говорящего и слушающего. Контекст может помогать угадывать незнакомые или нечеткие слова, а также разрешать неоднозначности или ошибки в речи.

Для улучшения качества распознавания речи необходимо учитывать все эти факторы и применять соответствующие методы и алгоритмы для их компенсации или устранения.

Разработка тестового приложения для распознавания речи

Целью данного приложения является демонстрация возможностей распознавания речи на основе Microsoft Speech API. Это приложение позволит пользователю диктовать команды на русском языке и видеть их текстовое представление на экране. Также это приложение может быть использовано для обучения или тестирования распознавания речи в разных условиях.

Функции, которые будет выполнять приложение, включают следующие действия:

- инициализация и настройка объекта SpeechRecognitionEngine или SpeechRecognizer для работы с локальным или облачным движком или сервисом распознавания речи;

- загрузка и применение грамматики для распознавания определенных команд или

запросов на русском языке;

- установка входного аудиоустройства по умолчанию для захвата звукового сигнала от микрофона пользователя;

- запуск и остановка процесса распознавания речи по нажатию кнопок “Старт” и “Стоп” (рис. 4);

- вывод текста распознанной речи в текстовое поле на экране (рис. 5).



Рисунок 4 – Интерфейс тестового приложения



Рисунок 5 – Интерфейс тестового приложения (распознавание текста)

Факторы, которые способствовали разработке приложения, включают:

- рост интереса к технологии распознавания речи и ее применению в разных сферах жизни и деятельности;

- развитие и улучшение движков и сервисов распознавания речи, основанных на статистических моделях и нейронных сетях;

- доступность и простота использования Microsoft Speech API для интеграции функций распознавания речи в приложения для Windows.

Для обучения этого приложения для распознавания слов и предложений на русском языке нужно использовать грамматику и языковой пакет для Microsoft Speech API. Грамматика - это набор правил, которые определяют, какие слова или фразы может распознать приложение. Языковой пакет - это набор файлов, которые содержат модели и словари для определенного языка или диалекта.

Для создания грамматики можно использовать класс Choices, который позволяет добавить список возможных команд или запросов на русском языке. Например:

```
numbers = new Choices();  
numbers.Add("привет", "как дела", "что делаешь", "покажи мне маршрут к парку", "назад");
```

Для загрузки языкового пакета нужно установить его на компьютере и указать культуру ru-ru при создании объекта SpeechRecognitionEngine или SpeechRecognizer. Например:

```
sre = new SpeechRecognitionEngine(new CultureInfo("ru-ru"));
```

Выводы

В статье рассмотрены основные подходы, методы и факторы, связанные с распознаванием речи в реальном времени. Было показано, как можно создать тестовое приложение для распознавания речи на основе Microsoft Speech API и как можно улучшить качество распознавания речи с помощью модели адаптации. Мы также описали библиотеку using Microsoft.Speech.Recognition и ее возможности для работы с разными движками и сервисами распознавания речи. Статья полезна для тех, кто интересуется технологией распознавания речи и хочет использовать ее в своих приложениях.

В заключение можно отметить, что распознавание речи - это динамичная и развивающаяся область, которая предлагает много перспектив и вызовов для исследователей и разработчиков.

Литература

1. Распознавание речи // Википедия. - Режим доступа: https://ru.wikipedia.org/wiki/%D0%A0%D0%B0%D1%81%D0%BF%D0%BE%D0%B7%D0%BD%D0%B0%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5_%D1%80%D0%B5%D1%87%D0%B8 - Загл. с экрана.
2. Speech-to-Text // WIPO. - Режим доступа: https://www.wipo.int/about-ip/ru/artificial_intelligence/speech_to_text.html - Загл. с экрана.
3. Как работают системы распознавания речи // Se7en. - Режим доступа: <https://se7en.ws/kak-rabotayut-sistemy-raspoznavaniya-rechi/> - Загл. с экрана.
4. Speech Recognition Using C# // C# Corner. - Режим доступа: <https://www.c-sharpcorner.com/article/speech-recognition-using-C-Sharp/> - Загл. с экрана.
5. Easy C# Speech Recognition // AssemblyAI. - Режим доступа: <https://www.assemblyai.com/blog/easy-c-speech-recognition/> - Загл. с экрана.
6. How to recognize intents from speech (C#) // Microsoft Learn. - Режим доступа: <https://learn.microsoft.com/en-us/azure/cognitive-services/speech-service/how-to-recognize-intents-from-speech-csharp> - Загл. с экрана.
7. Алгоритмы распознавания речи // Bing. - Режим доступа: <https://bing.com/search?q=%d0%b0%d0%bb%d0%b3%d0%be%d1%80%d0%b8%d1%82%d0%bc%d1%8b+%d1%80%d0%b0%d1%81%d0%bf%d0%be%d0%b7%d0%bd%d0%b0%d0%b2%d0%b0%d0%bd%d0%b8%d1%8f+%d1%80%d0%b5%d1%87%d0%b8> - Загл. с экрана.
8. Как работает распознавание речи в Яндексе // Хабр. - Режим доступа: https://habr.com/ru/companies/ashmanov_net/article/s/523412/ - Загл. с экрана.
9. Как работает распознавание речи в Google // Хабр. - Режим доступа: <https://habr.com/ru/articles/226143/> - Загл. с экрана.
10. Как работает распознавание речи в Nuance // Хабр. - Режим доступа: <https://habr.com/ru/post/135087/> - Загл. с экрана.

Баляба Я.В., Рычка О.В. Разработка приложения для распознавания речи. Статья посвящена технологии распознавания речи в реальном времени и ее применению в различных сферах жизни и деятельности. Рассмотрены основные подходы, методы и факторы, которые влияют на качество и скорость распознавания речи. Показано, как можно создать тестовое приложение для распознавания речи на основе Microsoft Speech API, которое позволяет пользователю диктовать команды на русском языке и видеть их текстовое представление на экране. В статье описаны основные функциональные требования к приложению. Рассмотрены способы улучшения качества распознавания речи с помощью модели адаптации, которая позволяет увеличить вероятность распознавания определенных слов или фраз. Приведен пример интерфейса системы.

Ключевые слова: распознавание речи, нейронные сети, скрытые марковские модели.

Balyaba Ya., Rychka O.V. Development of a speech recognition application. The article is devoted to real-time speech recognition technology and its application in various spheres of life and activity. The main approaches, methods and factors that affect the quality and speed of speech recognition are considered. It shows how you can create a test application for speech recognition based on the Microsoft Speech API, which allows the user to dictate commands in Russian and see their text representation on the screen. The article describes the main functional requirements for the application. Ways to improve the quality of speech recognition using an adaptation model that allows you to increase the probability of recognizing certain words or phrases are considered. An example of the system interface is given.

Keywords: speech recognition, neural networks, hidden Markov models.

Статья поступила в редакцию 20.02.2023
Рекомендована к публикации профессором Зори С.А.

Нейросетевая модель автоматизированного перевода

Т.А. Васяева, Ю.А. Золушкин, Т.В. Мартыненко, Е.А. Шуватова
ГОУВПО «Донецкий национальный технический университет»
vasyaeva@gmail.com, illuzium1999@gmail.com,
tatyana.v.martynenko@gmail.com, mauritia88@gmail.com

Аннотация

В статье рассмотрены подходы к реализации автоматизированных систем перевода, современные примеры таких систем и принципы их работы. Проанализированы актуальные проблемы автоматизированного перевода, показано, что нейросетевые модели имеют преимущества. Реализована нейросетевая модель автоматизированного перевода с использованием рекуррентных нейронных сетей. Для разработки и обучения нейронной сети использован язык программирования Python, Обучение модели и описанные эксперименты выполнены на корпусе данных WMT. Реализованная модель имеет архитектуру кодер-декодер.

Введение

Поскольку мир становится все более глобализированным, потребность в точном и эффективном переводе как никогда высока. Качественный перевод важен в различных областях:

1. Глобальный бизнес. В работе на международном уровне, позволяет легко переводить документы, контракты, сообщения, электронные письма, а также речь участников конференции или деловой встречи.

2. Туризм. Помогает туристам, приехавшим в другую страну ориентироваться на местности, понимая информационные объявления.

3. Онлайн-переводчики могут быстро и точно переводить тексты с сайтов, онлайн-чатов и социальных сетей.

4. Медицина. Используется при переводе медицинских документов и инструкций на разные языки.

Анализ предметной области

Исследования в области автоматизированного перевода, одного из направлений работы в области обработки естественного языка, ведутся очень давно. Первые модели и разработанные на их основе системы были реализованы еще в 1950-х годах. Решением задачи машинного перевода могут быть: правилые системы (rule-based) (1950-е годы); статистические модели (statistics-based) (1960-2010); нейронные модели (neural-based) (2010-е годы - настоящее время).

В последние годы эффективным решением стал нейросетевой перевод, использующий

возможности искусственного интеллекта. Рассмотрим некоторые из наиболее известных современных систем автоматизированного перевода и основные принципы их работы.

1. Google Translate – одна из самых популярных систем автоматизированного перевода; запущена в 2006 году и на сегодняшний день поддерживает более 100 языков; основана на статистическом подходе, также использует нейронную сеть для улучшения производительности и качества перевода.

2. Yandex.Translate – поддерживает более 90 языков и использует схожий подход, что и Google Translate, но использует свой собственный словарь, который базируется на большом количестве текстов на разных языках.

3. Microsoft Translator – поддерживает более 60 языков и также основывается на статистическом подходе. Чтобы улучшить качество перевода, использует технологию глубокого обучения и нейронные сети.

4. DeepL – относительно новая система автоматизированного перевода, запущенная в 2017 году; основана только на технологии глубокого обучения, использует нейронные сети; может переводить более чем 40 языков. В большинстве случаев перевод более качественный, в сравнении с системами, которые основываются на статистическом подходе.

5. Systran – является одной из более ранних систем автоматизированного перевода и основывается на гибридном подходе, который использует как статистические данные, так и лингвистические правила; поддерживает более 50 языков и используется многими компаниями и государственными учреждениями для перевода текстов.

Постановка проблемы

Несмотря на хорошие результаты нейросетевого перевода в настоящее время продолжают исследования по следующим направлениям:

1. Качество перевода. В настоящее время качество перевода нейросетевыми моделями существенно лучше, но не всегда идеально.

2. Многоязычность. Перевод на несколько языков, используя одну и ту же модель, является еще одной актуальной и нерешенной проблемой.

3. Данные. Доступность и качество параллельных текстов (текстовых данных, которые содержат пары с предложением на одном языке и его переводом на другом языке) являются довольно серьезной проблемой. Особенно если данные получены автоматически, например, субтитры или технические описания.

4. Нестабильность. Нейросетевые модели имеют тенденцию к переобучению и плохой устойчивости к изменениям в данных. Это приводит к потере качества перевода и необходимости постоянного переобучения модели.

5. Скорость. Нейросетевые модели по-прежнему обладают довольно высокими требованиями к вычислительным ресурсам и времени. Работа с такими моделями может быть крайне медленной, что не всегда приемлемо в реальном мире.

6. Локализация. Нейросетевой перевод достаточно плохо работает с сочетаниями языков, содержащих локальные диалекты или специфические термины.

В связи с этим, дальнейшие исследования в области построения языковых моделей и реализации на их основе систем автоматизированного перевода являются актуальными и перспективными.

Постановка задачи

Задача машинного перевода формально описывается так: у нас есть *source* предложение – это предложение на языке, с которого мы должны перевести и *target* предложение – это предложение на языке, на который мы должны привести.

$$\begin{aligned} source &= x_1, x_2, \dots, x_n, \\ target &= y_1, y_2, \dots, y_n. \end{aligned}$$

Задача машинного перевода найти наиболее вероятную последовательность на *target* языке при условии входящей последовательности на *source* языке.

$$\widehat{target} = \underset{target}{\operatorname{argmax}} P(target | source, \theta).$$

Target предсказанной модели перевода это argmax по всем *target* то есть по всем переводам, которые вообще могут соответствовать предложению *source*. То есть самый вероятный перевод предложения:

$$\begin{aligned} P(target | source) &= P(y_1, y_2 \dots y_m | source) = \\ &= P(y_1 | source) * P(y_2 | y_1, source) \dots \\ &= P(y_m | y_1, \dots, y_{m-1}, source) \end{aligned}$$

Необходимо получить переводы *source* предложения на *target* язык, который может сгенерировать наша модель и выбирать из них с максимальной вероятностью. То есть тот, который вероятнее всего является действительно переводом предложения *source* на *target*.

Подготовка набора данных

Одним из основных аспектов, влияющих на качество моделей машинного перевода, является выбор набора данных (датасета). Рассмотрим наиболее современные и открытые датасеты параллельных текстов, использующиеся при создании моделей обработки естественного языка (табл. 1).

Таблица 1 - Сравнительные характеристики современных датасетов параллельных текстов

Название датасета	Разработчик	Объем
WikiMatrix	Meta Research	135 миллионов параллельных предложений
CCMatrix	Meta Research	4.5 миллиарда параллельных предложений
News-Commentary	Barrault et al.	109 текстов
WMT	Vojar et al.	42.3 миллионов параллельных предложений
MASSIVE	Amazon	1 миллион текстов

Следует отметить, что нацеленность датасета MASSIVE [1] на парадигму MMNLU (massively multilingual natural-language understanding) делает набор данных чрезмерно тяжеловесным.

WikiMatrix [2] наиболее часто применяются при обучении систем машинного перевода между разными языками без необходимости переходить на английский язык. WMT [3] является наиболее часто используемым для построения моделей машинного обучения, содержит тексты на русском и английском языках, и использован для исследований в данной работе.

Разработка нейросетевой модели перевода

Перед разработкой модели, необходимо выполнить следующие этапы: токенизация; нормализация; фильтрация; векторизация.

В [4] последовательно рассмотрены перечисленные этапы обработки естественного языка, выполнено их описание, реализация и тестирование. Также в статье [4] проведен анализ методов преобразования слов, таких как Embedding и One-hot-encoding. Для решения задачи нейронного переводчика было выбрано векторное представление слов Embedding.

Получение векторных представлений слов, на корпусе параллельных данных WMT 2020, с использованием word2vec описано в [5]. В работе [5] выполнена реализация на языке программирования Python с использованием облачного сервиса на основе Jupyter Notebook – Google Colab.

Нейросетевая модель для перевода, это языковая модель с ограничениями, т.е. модель не просто генерирует нормальные предложения, а они являются переводом конкретного предложения. Наиболее распространенными моделями являются модели кодер-декодер (рис. 1), которые обычно используют рекуррентную нейронную сеть (RNN).

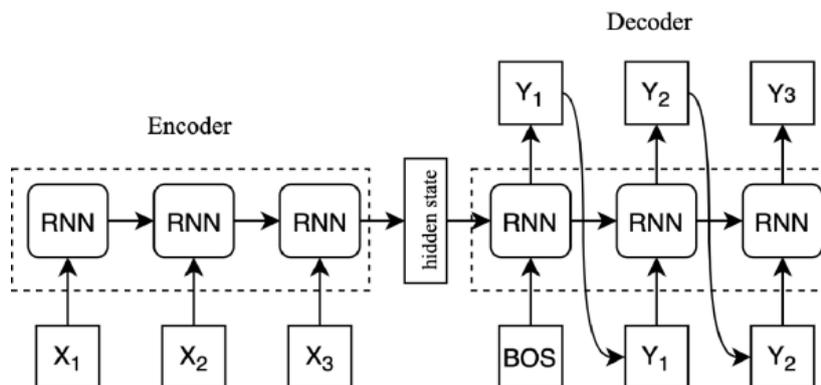


Рисунок 1 – Sequence-to-sequence модель для машинного перевода

Нейросетевая модель для перевода состоит из двух частей: энкодера и декодера. Энкодер (нейросетевая языковая модель) собирает информацию о предложении на source языке. Декодер использует информацию, собранную энкодером, чтобы сгенерировать предложение-перевод на target языке.

Входное/исходное предложение X вводится в кодер по одному слову за раз. Необходимо добавить маркеры начала последовательности ($\langle \text{sos} \rangle$) и конца последовательности ($\langle \text{eos} \rangle$) к началу и концу предложения соответственно. На каждом шаге входом в кодер RNN является как текущее слово, x_t , так и скрытое состояние из предыдущего шага, h_{t-1} , и кодер RNN выводит новое скрытое состояние h_t . Скрытое состояние в упрощенном понимании является векторным представлением предложения. RNN можно представить как функцию как от x_t , так и от h_{t-1} :

$$h_t = \text{EncoderRNN}(x_t, h_{t-1}).$$

В качестве RNN может быть выбрана любая рекуррентная архитектура, такая как LSTM [6] или GRU.

$$X = \{x_1, x_2, \dots, x_T\},$$

где $x_1 = \langle \text{sos} \rangle$, $x_2 =$ первое слово в предложении, и т. д.

Начальное скрытое состояние, h_0 , обычно либо инициализируется нулями, либо значением, полученным при обучении.

Как только последнее слово x_T передано в RNN, скрытое состояние h_T используется в качестве вектора контекста, то есть $h_T = z$. Это векторное представление всего исходного предложения.

После получения вектора контекста z , необходимо начать его декодирование, чтобы получить целевое предложение. Как и в описанном выше случае, необходимо добавить маркеры начала и конца последовательности к предложению. На каждом шаге входом в декодер RNN является текущее слово y_t , а также скрытое состояние предыдущего шага s_{t-1} , где начальное скрытое состояние декодера s_0 представляет собой вектор контекста, $s_0 = z = h_T$, т. е. начальное скрытое состояние декодера является конечным скрытым состоянием кодера. Таким образом, подобно кодеру, декодер можно представить как:

$$s_t = \text{DecoderRNN}(y_t, s_{t-1}).$$

В декодере нужно перейти от скрытого состояния к реальному слову, поэтому на каждом шаге необходимо использовать st для предсказания (пропуская его через линейный

слой), что является следующим словом в последовательности \hat{y}_t .

$$\hat{y}_t = s_t.$$

Маркер <eos> используется для первого ввода в декодер y_t . Для последующих вводов $y_{t>1}$, в некоторых случаях необходимо использовать следующее слово в последовательности y_t , а иногда использовать слово, предсказанное декодером \hat{y}_{t-1} .

Данный подход соответствует обучению с подкреплением (teacher forcing).

Параметры для обучения нейронной сети приведены в табл. 2.

Реализация структуры нейронной сети на языке python представлена на рис. 2. Результаты обучения представлены на рисунках 3 и 4.

Таблица 2. – Параметры обучения нейронной сети seq2seq

Название параметра		Значение
Оптимизатор		Adam
Количество эпох		40
Критерий (loss)		Кросс энтропия
Размер батча		64
Размер эмбединга	кодера	256
	декодера	256
Размер скрытого слоя		512
Дропаут	кодера	0.5
	декодера	0.5
Коэффициент обучения с подкреплением		0.5

```
Seq2Seq(
  (encoder): Encoder(
    (embedding): Embedding(148755, 256)
    (rnn): LSTM(256, 256, num_layers=2, dropout=0.5, bidirectional=True)
    (dropout): Dropout(p=0.5, inplace=False)
  )
  (decoder): Decoder(
    (embedding): Embedding(75949, 256)
    (rnn): LSTM(256, 512, num_layers=2, dropout=0.5)
    (out): Linear(in_features=512, out_features=75949, bias=True)
    (dropout): Dropout(p=0.5, inplace=False)
  )
)
```

Рисунок 2 – Структура нейронной сети

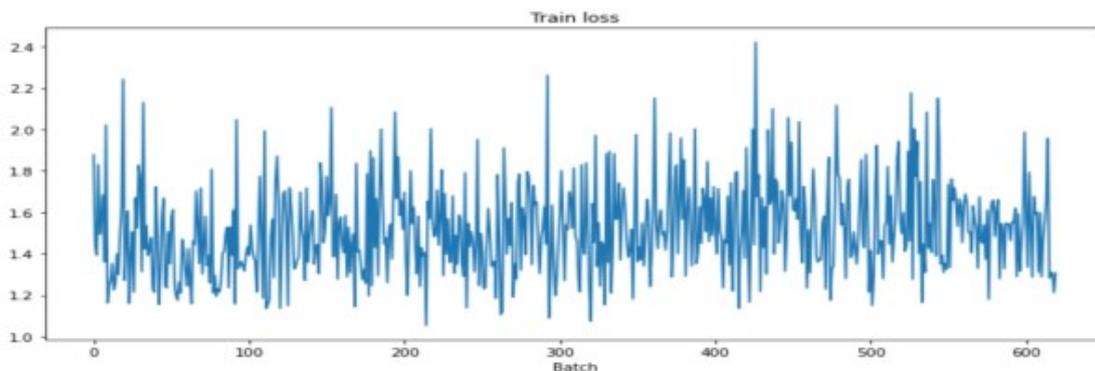


Рисунок 3 - График изменения ошибки во время обучения нейронной сети

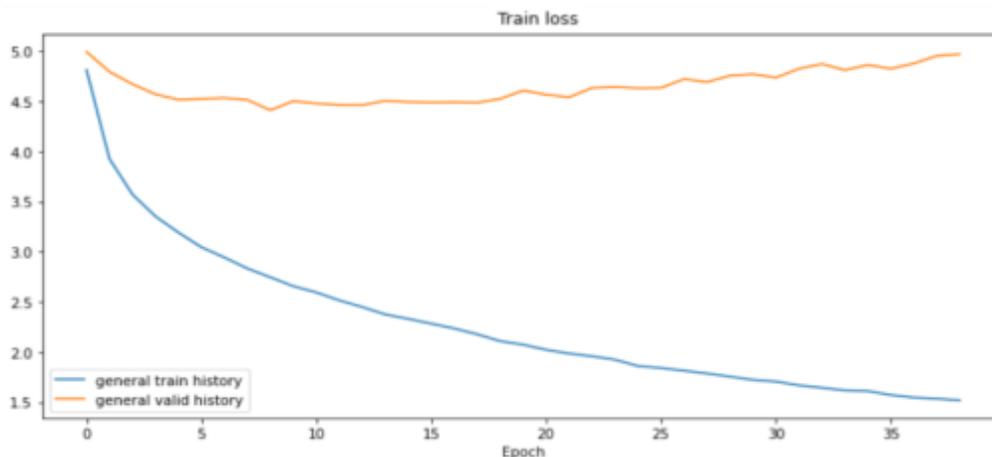


Рисунок 4 - График изменения ошибки обучения от количества эпох

Оценка качества полученного перевода осуществлялась с использованием метрики BLEU [7].

Заключение

Результатом работы является разработанная нейросетевая модель автоматизированного перевода. Выполнена математическая постановка задачи к переводу текстов. Выбран корпус данных для обучения модели. В работе использованы реализованные ранее [4] этапы обработки текста и реализация модуля получения векторных представлений слов [5]. Разработан модуль нейросетевого перевода, обучена модель для перевода текстов.

Литература

1. MASSIVE: A 1M-Example Multilingual Natural Language Understanding Dataset with 51 Typologically-Diverse Languages – Jack FitzGerald, Christopher Hench, Charith Peris, Scott Mackie, Kay Rottmann, Ana Sanchez, Aaron Nash, Liam Urbach, Vishesh Kakarala, Richa Singh, Swetha Ranganath, Laurie Crist, Misha Britan, Wouter Leeuwis, Gokhan Tur, Prem Natarajan [Электронный ресурс]. – Режим доступа: <https://arxiv.org/abs/2204.08582>
2. WikiMatrix: Mining 135M Parallel Sentences in 1620 Language Pairs from Wikipedia –vy, Vishrav Chaudhary, Shuo Sun, Hongyu Gong, Francisco Guzmán [Электронный ресурс]. – Режим доступа: <https://arxiv.org/abs/1907.05791>

3. Findings of the 2020 Conference on Machine Translation (WMT20) – Ondřej Bojar, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Philipp Koehn, and Christof Monz [Электронный ресурс]. – Режим доступа: <https://www.statmt.org/wmt20/pdf/2020.wmt-1.pdf>

4. Золушкин, Ю. А. Обработка естественного языка / Ю. А. Золушкин, Т. А. Васяева, А. А. Малицкая // Информатика, управляющие системы, математическое и компьютерное моделирование (ИУСМКМ-2021): Материалы XII Международной научно-технической конференции в рамках VII Международного Научного форума Донецкой Народной Республики к 100-летию ДонНТУ, Донецк, 26–27 мая 2021 года. – Донецк: Донецкий национальный технический университет, 2021. – С. 71-78.

5. Разработка векторных представлений слов для нейросетевой языковой модели / Ю. А. Золушкин, Т. А. Васяева, Т. В. Мартыненко, Е. А. Шуватова // Информатика, управляющие системы, математическое и компьютерное моделирование (ИУСМКМ-2022): Материалы XIII Международной научно-технической конференции в рамках VIII Международного Научного форума Донецкой Народной Республики, Донецк, 25–26 мая 2022 года. – Донецк: Донецкий национальный технический университет, 2022. – С. 219-223.

6. Manaswi N.K. RNN and LSTM. In: Deep Learning with Applications Using Python. Apress, Berkeley, CA Recurrent Neural Networks(2018)

7. Papineni, K.; Roukos, S.; Ward, T.; Zhu, W. J., BLEU: a method for automatic evaluation of machine translation (PDF). ACL-2002: 40th Annual meeting of the Association for Computational Linguistics. pp. 311–318.

***Васяева Т.А., Золушкин Ю.А., Мартыненко Т.В., Шуватова Е.А. Нейросетевая модель автоматизированного перевода.** В статье рассмотрены подходы к реализации автоматизированных систем перевода, современные примеры таких систем и принципы их работы. Проанализированы актуальные проблемы автоматизированного перевода, показано, что нейросетевые модели имеют преимущества. Реализована нейросетевая модель автоматизированного перевода с использованием рекуррентных нейронных сетей. Для разработки и обучения нейронной сети использован язык программирования Python. Обучение модели и описанные эксперименты выполнены на корпусе данных WMT. Реализованная модель имеет архитектуру кодер-декодер.*

***Ключевые слова:** автоматизированный перевод, нейросетевая модель, кодер-декодер, рекуррентные нейронные сети, глубокое обучение, машинное обучение, обучение с подкреплением, параллельные тексты, Python*

***Vasyaeva T.A., Zolushkin Yu.A., Martynenko T.V., Shuvatova E.A. Neural network model of automated translation.** The article discusses approaches to the implementation of automated translation systems, modern examples of such systems and the principles of their work. The actual problems of automated translation are analyzed, it is shown that neural network models have advantages. Implemented neural network model of automated translation using recurrent neural networks. The programming language Python was used to develop and train the neural network. Model training and the described experiments were performed on the WMT data corpus. The implemented model has an encoder-decoder architecture.*

***Keywords:** automated translation, neural network model, encoder-decoder, recurrent neural networks, deep learning, machine learning, reinforcement learning, parallel texts, Python*

Статья поступила в редакцию 25.02.2023

Рекомендуется к публикации профессором Скобцовым Ю.А.

УДК: 004.855.5

Вывод древовидной вспомогательной грамматики как инструмент автоматического обобщения алгоритмов

Л. О. Воробьёв, А. В. Григорьев
Донецкий национальный технический университет, г. Донецк
e-mail: lev.vorobyov@rambler.ru , grigorievalvl@gmail.com

Аннотация.

В статье рассматривается проблема автоматического обобщения алгоритмов в процедурных языках программирования. Предлагается уникальный способ объединения семантики алгоритмов путём представления программы в виде рекуррентной функции в обратной польской нотации и вывод древовидной вспомогательной грамматики (TAG) с образованием И-ИЛИ дерева с помощью алгоритма теоретико-множественных операций над формальными грамматиками. Приводится описание И-ИЛИ графа внутреннего представления алгоритма и основные этапы обобщения программ. В настоящее время алгоритм не апробирован.

Введение

Основы теории алгоритмов и формальных грамматик описаны в [1, 2] соответственно.

В предлагаемой работе рассматривается вопрос синтеза алгоритмов решения общей задачи по примерам решений частных задач путём объединения формальных грамматик. Задача объединения формальных грамматик подробно описана в [3], реализации некоторых алгоритмов обобщения грамматик на языке Python приведены в [4].

При написании программы на императивном языке программирования (ИЯП) иногда возникает задача объединить несколько алгоритмов с одинаковым интерфейсом (набором аргументов). В результате объединения получается обобщённая реализация алгоритма. Инструментами обобщения алгоритмов могут быть условное выполнение операторов и шаблонные функции [5]. Существующие решения задачи автоматического обобщения программ [6, 7] используют вероятностные методы. Детерминированные алгоритмы обобщения программ в литературе освещены слабо, что обуславливает **актуальность** формализации процесса обобщения программ.

Введём следующее определение. **Обобщённая программа** — такая программа, которая решает все задачи из некоторого класса. Например: пусть есть множество задач C и есть два собственных подмножества A и B , при этом может быть $A \cap B \neq \emptyset$ и обычно $A \cup B \neq C$. Задача обобщения состоит в том, чтобы из двух программ, решающих частные задачи из множества A и B соответственно, составить программу, которая решает все задачи из множества C . Это и будет обобщённая программа.

Мы предполагаем, что обобщённые программы относятся к одному классу, т.е.

программы, решающие сходную задачу. Таким образом, считается, что обобщённая программа не может включать алгоритмы из различных классов. В таком случае, для каждого класса задач создаётся отдельная обобщённая программа. Классификация задач — это предварительный этап, который нужно выполнять перед обобщением программ.

Цель работы: автоматизировать процесс обобщения алгоритмов.

Исходя из поставленной цели предполагается решить следующие задачи:

- привести справочный материал по выводу грамматик, древовидным грамматикам и автоматическому программированию;
- разработать математическую форму представления обобщённых алгоритмов;
- проверить полноту по Тьюрингу предлагаемого математической формы представления алгоритма;
- описать процедуру обобщения алгоритмов во внутреннем представлении;
- описать процедуру перевода текстов программ во внутреннее представление и обратно.

Краткий анализ существующих решений

Задача обобщения программ относится к более общей задаче автоматического программирования. Вероятностные методы автоматического программирования описаны в [8, 9]. Они основаны на случайной генерации и оценке возможных решений по некоторому критерию. Обычно для оценки возможных решений в автоматическом программировании программу проверяют на наборе примеров входов и выходов. Здесь же рассматривается вывод обобщённой программы по примерам

программ. Поскольку программа состоит из слов (термов), то она является сентенциальной формой из некоторого формального языка. Таким образом, обобщение программ есть частный случай грамматического вывода (grammatical inference, GI).

Индуктивный вывод грамматики - это разновидность задачи машинного обучения по примерам, когда в качестве примеров предоставляется набор синтаксически правильных цепочек термов, и строится распознающий автомат, или продукционные правила порождения этих примеров. Алгоритм синтеза распознающего автомата с помощью обучения рекуррентной нейронной сети [10] относится к вероятностным методам обучения, и позволяет вывести грамматику ограниченной сложности (не более 5 состояний автомата), и подходит только для вывода регулярной грамматики. Вывод формальной грамматики, по сути является сжатием информации. Ведь исходные сентенциальные формы имеют больший размер в совокупности, чем продукционные правила порождения этих форм, потому что сентенциальные формы обычно содержат повторяющиеся фрагменты, которые выявляет алгоритм вывода грамматики.

Существует также детерминированный алгоритм построения продукционных правил формальной грамматики по примерам порождаемых ею слов, впервые описанный в [11]. Проблема этого алгоритма состоит в том, что он не создаёт грамматику с рекурсивными правилами. Рекурсивные правила — это правила, в которых в левой и правой части содержится один и тот же нетерминальный символ [2 с. 19]. Поскольку количество операторов в одной программе обычно ничем не ограничено то для описания формальной грамматики, сентенциальные формы которой являются программами, нужны рекурсивные правила.

Древовидная вспомогательная грамматика (tree-adjunct grammars, TAG) - это система переписывания деревьев [12], введённая Аравиндом Джоши в 1975 г. для описания естественного языка. Формально TAG является пятёркой (T, V, I, A, S) , где: T — множество терминальных символов; V — множество нетерминальных символов; $S \in V$ — начальный символ; I — множество начальных деревьев, корень которых помечен S , листья — терминальными символами, а остальные вершины — нетерминальными символами; A — множество вспомогательных деревьев, листья которых помечены терминальными и нетерминальными символами, а промежуточные узлы - только нетерминальными символами.

Особенностью вспомогательных деревьев TAG является наличие одного листа, который помечен тем же символом, что и корень дерева.

Этот лист называется ногой. Порождение дерева в TAG осуществляется путём замещения поддерева от нетерминального узла вспомогательным деревом, у которого корень соответствует тому же нетерминальному символу. А исходное поддерево присоединяется к ноге вспомогательного дерева. Это позволяет порождать бесконечные языки, что нужно для описания грамматики обобщённых программ. В [12] описывается метод автоматического программирования AntTAG, использующий TAG в качестве способа представления программы, который позволяет представлять дерево программы, как линейную последовательность термов для автоматического программирования путём муравьиной оптимизации.

Ещё одной формой линейного представления программы является польская запись. Польская запись - это запись, в которой операторы следуют за или перед операндами. Бывает прямая и обратная польская запись. Примером прямой польской записи является язык Lisp.

Если представить вспомогательное дерево TAG, как рекурсивную функцию, где листья — это операнды, промежуточные узлы — это операции, а корень — это точка входа в программу, и использовать обратную польскую запись, то получим линейную последовательность термов. Это позволяет использовать алгоритм теоретико-множественных операций над формальными грамматиками [11], как способ вывода вспомогательных деревьев TAG для обобщения программ.

Предлагаемый подход

Предлагаемое решение состоит в том, что обобщаемые примеры программ представляются в виде сентенциальной формы некоторой формальной грамматики. Из набора сентенциальных форм выводится эта формальная грамматика, которая затем переводится в набор обобщённых подпрограмм.

Обобщаемые примеры программ назовём прототипами. Прототип имеет номер в базе данных. Из прототипов методом синтаксического и семантического анализа можно извлечь алгоритм. Для внутреннего представления алгоритма используем И-ИЛИ граф, как способ хранения атрибутивной грамматики. Узлы И-ИЛИ графа помечены номерами прототипов, а дуги от ИЛИ-узлов — значениями параметров.

Для обобщения программ нужно исходные прототипы различать с помощью некоторого критерия. Для указания, какими параметрами обладает конкретный прототип, используется два И-ИЛИ дерева: дерево выбора прототипов и дерево порождения прототипов.

И-ИЛИ дерево порождения прототипов задаёт множество синтаксически правильных выражений, определённых на алфавите терминальных символов. Подмножеством множества синтаксически правильных выражений является семантически правильные выражения, которые задаются номерами прототипов в ИЛИ-узлах. При нисходящем выводе пользователь выбирает одну из альтернатив в дереве выбора прототипов. ИЛИ-узлы с номерами прототипов, записанными в остальных альтернативах, в дереве порождения прототипов удаляются [13].

Рассмотрим пример обобщённого алгоритма сложения n чисел, выраженного в виде рекурсивной функции f от переменного числа аргументов:

$$\left\{ \begin{array}{l} f(x_1) = x_1; \\ f(x_1, x_2, \dots, x_n) = x_1 + f(x_2, \dots, x_n). \end{array} \right. \quad (1)$$

Здесь параметром, от которого зависит выбор вида функции, является количество аргументов x_i . Таким образом, одна из исходящих из корневого ИЛИ-узла дерева на рис. 1 дуг будет помечена значениями параметра $n = 1$, а вторая не будет помечена, и соответствует всем остальным значениям этого параметра.

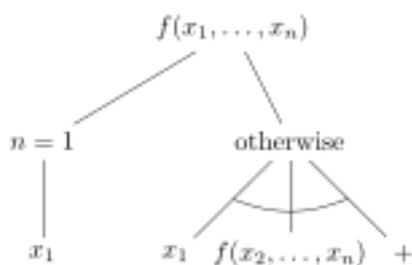


Рисунок 1 — Представление алгоритма в виде И-ИЛИ дерева

Здесь циклы устраняются путём добавления новых символов операций в алфавит и превращения узла, создающего циклическую связь в лист. Впоследствии, при генерации программы этот лист заменяется на рекурсивный вызов подпрограммы. Таким образом, на рис. 1 изображен пример вспомогательного дерева для TAG, полученной в результате вывода грамматики из И-ИЛИ дерева прототипов.

ИЛИ-узлы транслируются в блоки условного выполнения операторов, а И-узлы — в последовательность выполнения операторов в обратной польской нотации.

И-ИЛИ граф есть структурный граф программы [14], т.е. связный помеченный ориентированный граф, такой, что у него есть вершина, из которой достижима любая вершина; и есть хотя бы одна вершина, которая имеют нулевую степень исхода. Каждая вершина И-

ИЛИ графа, с ненулевой степенью исхода, является либо И-вершиной, либо ИЛИ вершиной. Для представления алгоритмов применяется И-ИЛИ граф со следующими особенностями:

- узлы графа с нулевой степенью исхода помечаются терминальными символами языка записи рекуррентных формул в обратной польской нотации, т.е. переменные и операции;
- исходящие из И-узла дуги связаны с вершинами, которые помечены порядковым номером;
- порядок следования подчинённых узлов по И определяет порядок записи выражений в рекуррентной формуле;
- исходящие из ИЛИ-узла дуги связаны с узлами, помеченными списком номеров прототипов, и называются ИЛИ-альтернативами [13];
- ИЛИ-узел трактуется как рекуррентная формула с набором альтернативных случаев;
- если из И-узла исходят дуги, связанные с узлами с нулевой степенью исхода, то этот И-узел трактуется как рекуррентная формула, иначе это последовательность рекуррентных формул;
- циклические связи между узлами означают рекуррентное вычисление формул.

Между ИЛИ-узлами разных деревьев установлены отношения соответствия с помощью номеров прототипов. При выборе альтернативы в одном дереве все альтернативы в другом дереве, не связанные с выбранной альтернативой номерами прототипов, удаляются.

Это позволяет определить отношение ЕСЛИ-ТО между двумя языками: языком спецификаций и языком прототипов.

По факту, продукционная база знаний, то есть набор продукций ЕСЛИ-ТО реализуется на алгоритмическом языке программирования с помощью конструкции if-else.

Каждому ИЛИ-узлу в дереве спецификаций соответствует хотя бы один узел в дереве прототипов.

Проверка полноты И-ИЛИ графа для представления алгоритмов

Теорема 1. Представленная выше нотация записи алгоритмов является полной по Тьюрингу.

Доказательство: рекуррентные формулы являются полными по Тьюрингу вариантами записи алгоритмов. Следовательно, И-ИЛИ дерево, задающее в компактном виде набор рекуррентных формул, является полным по Тьюрингу.

Следствие. Программу на универсальном языке программирования можно перевести в описанное выше И-ИЛИ дерево и наоборот.

Описанное выше задание алгоритма соответствует принципам, которые положены в основу функционального программирования и языка Haskell.

Задача состоит в том, чтобы исходные примеры программ преобразовать в И-ИЛИ деревья синтаксического разбора, разделить их на три И-ИЛИ дерева для внешней границы, состава блоков и связей, затем обобщить полученные деревья, получив обобщённые три И-ИЛИ дерева, которые затем объединить в одно И-ИЛИ дерево алгоритма и перевести его в программу.

Этапы обобщения примеров программ:

- кластеризация примеров по признаку наличия и отсутствия параметров.
- синтаксический и семантический анализ примеров с построением набора рекурсивных функций в обратной польской нотации;
- поиск одинаковых фрагментов в цепочках термов, и создание для каждого из них нового продукционного правила до тех пор, пока не получится один нетерминальный символ;
- представление грамматики в виде И-ИЛИ дерева, выявление повторяющихся поддеревьев и вспомогательных деревьев TAG;
- перевод вспомогательных деревьев полученной TAG в рекурсивные функции и генерация программы.

Пример обобщения набора программ

Алгоритм обобщения функций получает набор программ в виде цепочек терминальных символов, и значения некоторого параметра, от которого зависит вид этой программы. Затем алгоритм производит выведение общего алгоритма и возвращает полученный алгоритм в форме рекуррентного выражения. Например, пусть необходимо вывести общий алгоритм сложения N чисел (не важно, какого типа). В исходном виде функции представляют собой цепочки из элементов массива и знаков сложения, как показано в табл. 1.

Таблица 1 - Входной набор прототипов и их свойств

№	Выражение для $f(X,N)$	Свойство N
1	x_1	1
2	$x_1 + x_2$	2
3	$x_1 + x_2 + x_3$	3
4	$x_1 + x_2 + x_3 + x_4$	4
5	$x_1 + x_2 + x_3 + x_4 + x_5$	5

Здесь в качестве параметра, от которого зависит программа, выступает N — количество

слагаемых. В результате обобщения нужно получить рекуррентную функцию, форма которой не зависит от количества слагаемых. Эта функция имеет вид:

$$f(\{x_i\}_i^n, n) = \begin{cases} f(\{x_i\}_i^{n-1}, n-1) + x_n, & n > 1; \\ x_1, & n = 1. \end{cases} \quad (2)$$

Сначала свойства прототипов используются для построения дерева решений по алгоритму С4.5. В результате получаем следующее дерево, в листьях которого находятся номера прототипов, ИЛИ-узлы помечены свойствами, в дуги от ИЛИ-узлов — значениями свойств.

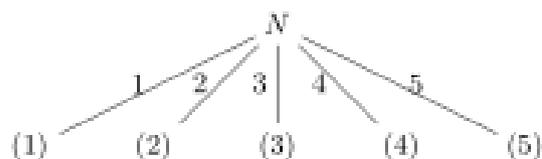


Рисунок 2 — Дерево решения для выбора прототипов

Параллельно исходные прототипы переводятся в обратную польскую запись:

Таблица 2 - Промежуточное представление прототипов

№ прототипа	Обратная польская запись
1	x_1
2	$x_1, x_2, +$
3	$x_1, x_2, x_3, +, +$
4	$x_1, x_2, x_3, x_4, +, +, +$
5	$x_1, x_2, x_3, x_4, x_5, +, +, +, +$

На основе данных примеров строится грамматика в виде И-ИЛИ дерева (рис. 3).

Это недообученное И-ИЛИ дерево, потому что оно включает только те примеры, которым его обучили, а нужно создать такую модель, которые включают все возможные варианты задач данного класса. Алгоритм унификации должен выявить сходство и построить более простую порождающую грамматику.

Индукция проводится с помощью метода, описанного в [11]. Алгоритм индуцирует регулярную грамматику с не более чем 5 состояниями. В данном случае прототипы в обратной польской записи определяются следующей КЗ грамматикой $G = (VN, VT, P, S)$ (3).

Это атрибутная грамматика. Здесь каждому нетерминальному символу соответствует атрибут N , и альтернативные правила применяются при наличии определённого значения атрибута.

Конечный автомат, распознающий эту грамматику, содержит ровно пять состояний.

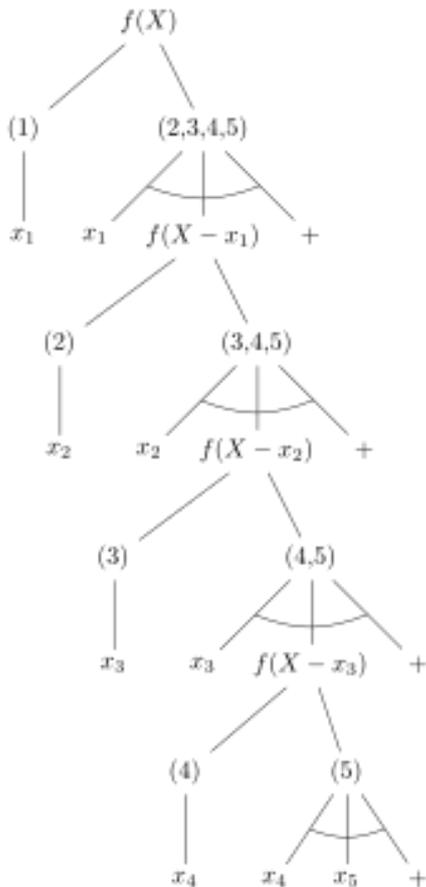


Рисунок 3 — Недообученное И-ИЛИ дерево прототипов

$$\begin{aligned}
 VN &= \{S, A, B, C, D\}; \\
 VT &= \{x_1, x_2, x_3, x_4, x_5, +\}; \\
 P &= \begin{cases} S \rightarrow x_1 \vee x_1 A; \\ A \rightarrow x_2 + x_2 B +; \\ B \rightarrow x_3 + x_3 C +; \\ C \rightarrow x_4 + x_4 D +; \\ C \rightarrow x_5 +. \end{cases} \quad (3)
 \end{aligned}$$

Тот же метод, предложенный в [10], вполне способен индуцировать такую грамматику, но он пригоден только для грамматики регулярного типа с числом состояний не больше пяти.

В результате выполнения алгоритма унификации получается И-ИЛИ дерево:



Рисунок 4 — Дообученное И-ИЛИ дерево прототипов

В скобках указывается номера прототипов. Это И-ИЛИ дерево соответствует алгоритму, выраженному в форме рекуррентной функции в (1).

Выражение в блоке if вычисляется исходя из свойств прототипов, номерами которых помечены ИЛИ-альтернативы.

Обобщение программы для всего класса задач, а не только для объединения подмножеств решаемых примерами задач, достигается путём анализа отрицательных примеров.

Заключение

Авторами предложен новый способ обобщения функций в алгоритмических языках программирования. Предлагаемый алгоритм опирается на метод вывода регулярной грамматики.

Отличия состоят с следующим:

- сентанциальные формы являются рекуррентными функциями в обратной польской нотации;
- алгоритм обрабатывает исходные линейные примеры и строит грамматику;
- для устранения рекуррентных связей программы представляются в виде вспомогательных деревьев TAG.

Данные отличия позволяют использовать алгоритм вывода грамматики, рассчитанный на вывод нерекурсивных правил.

В настоящее время этот алгоритм не был апробирован. Планируется проверить предлагаемый подход в программной реализации на языках Perl и PL/SQL. Грамматический анализатор примеров планируется реализовать с помощью инструментов lex и yacc.

Литература

1. Поляков, В. И. Основы теории алгоритмов / В. И. Поляков, В. И. Скорубский. – СПб : СПб НИУ ИТМО, 2012.
2. Лаздин, А. В. Формальные языки, грамматики, автоматы. / А. В. Лаздин. – СПб : Университет ИТМО, 2019.
3. de la Higuera C. Grammatical Inference: Learning Automata and Grammars. / C. de la Higuera. – Cambridge : Cambridge University Press, 2010.
4. Wojciech, W. Grammatical Inference. Т. 673 / W. Wojciech. – 2017.
5. Вандевурд, Д. Шаблоны C++. Справочник разработчика. / Д. Вандевурд, Н. М. Джонаттис, Д. Грегор. – Изд. 2. – СПб. : ООО «Альфа-книга», 2018. – 848 с.
6. Helmuth, T. Improving generalization of evolved programs through automatic simplification / T. Helmuth, N. Mcphee, E. Pantridge, L. Spector. – 2017. – С. 937-944.

7. Xu D. Neural Task Programming: Learning to Generalize Across Hierarchical Tasks / D. Xu, S. Nair, Y. Zhu [и др.]. – 2017.

8. Elleuch, S. From Metaheuristics to Automatic Programming / S. Elleuch, B. Jarboui, P. Siarry. – 2022. – С. 3-38.

9. O'Neill, M. Automatic programming: The open issue? / M. O'Neill, L. Spector // Genetic Programming and Evolvable Machines. – 2020. – Т. 21.

10. Грачев, П. Г. Генерация автоматов на основе рекуррентных нейросетей и автоматического выбора кластеризации [Текст: электронный] / П. Г. Грачев, С. Б. Муравьев, А. А. Фильченков, А. А. Шалыто // Информационно-управляющие системы, 2020. – № 1 (104). – URL: <https://cyberleninka.ru/article/n/generatsiya-avtomatov-na-osnove-rekurrentnyh-neyrosetey-i-avtomaticheskogo-vybora-klasterizatsii>

11. Григорьев, А. В. Алгоритм выполнения теоретико-множественных операций над грамматиками в среде специализированной оболочки для создания интеллектуальных САПР / А. В. Григорьев // Наукові праці Донецького

національного технічного університету : Проблеми моделювання та автоматизації проектування. – Донецьк : ДонНТУ, 2002. – С. 83-93.

12. Abbass, H. AntTAG: a new method to compose computer programs using colonies of ants / H. Abbass, N. Hoai, R. McKay. – 2002. – Т. 2. – С. 1654-1659.

13. Григорьев, А. В. Комплекс средств и методов работы с формальными грамматиками в семиотической концептуальной модели предметной области интеллектуальных САПР / А. В. Григорьев // Информатика и кибернетика. – Донецк: ДонНТУ, 2017. – № 1(7). – С. 46-72.

14. Крицкий, С. П. Реализация оптимизирующих преобразований программ с помощью структурных предикативных грамматик / С. П. Крицкий, Б. Ю. Тапкинов. – Текст: электронный // Известия вузов. Северо-Кавказский регион. Серия: Естественные науки. – 2006. – № S1. – URL: <https://cyberleninka.ru/article/n/realizatsiya-optimiziruyuschih-preobrazovaniy-programm-s-pomoschyu-strukturnyh-predikativnyh-grammatik>.

Воробьев Л. О., Григорьев А. В. Вывод древовидной вспомогательной грамматики как инструмент автоматического обобщения алгоритмов. В статье рассматривается проблема автоматического обобщения алгоритмов в процедурных языках программирования. Предлагается уникальный способ объединения семантики алгоритмов путём представления программы в виде рекуррентной функции в обратной польской нотации и вывод древовидной вспомогательной грамматики (TAG) с образованием И-ИЛИ дерева с помощью алгоритма теоретико-множественных операций над формальными грамматиками. Приводится описание И-ИЛИ графа внутреннего представления алгоритма и основные этапы обобщения программ. В настоящее время алгоритм не апробирован.

Ключевые слова: древовидная вспомогательная грамматика, вывод грамматики, автоматическое программирование, И-ИЛИ дерево.

Vorobyov L. O., Grigoriev A. V. Tree adjunct grammar inference as a tool for automatic generalization of algorithms. The article considers the problem of automatic generalization of algorithms in procedural programming languages. A unique way of combining the semantics of algorithms is proposed by representing the program as a recurrent function in reverse Polish notation and deriving a tree adjunct grammar (TAG) with the formation of an AND-OR tree using the algorithm of set-theoretic operations on formal grammars. A description of the AND-OR graph of the internal representation of the algorithm and the main stages of program generalization are given. At present, the algorithm has not been tested.

Keywords: tree adjunct grammar, grammar inference, automatic programming, AND-OR tree.

Статья поступила в редакцию 22.02.2023
Рекомендована к публикации профессором Зори С.А.

Проектирование архитектуры системы электронных денег

Р.В. Мальчева, К.А. Терещенко
ГОУВПО «Донецкий национальный технический университет»
E-mail: malcheva.raisa@yandex.ru

Аннотация

Рассмотрены базовые онлайн и офлайн архитектуры систем для внедрения механизмов и протоколов электронных денег. Рассмотрены основные схемы передачи платежей и проанализированы их узкие места. Предложен альтернативный подход, состоящий в распределении банковских обязанностей. Описана сервис-ориентированная архитектура SOA – многоуровневая архитектура, позволяющая оптимизировать как внутренние, так и внешние процессы. Выполнено моделирование бизнес-процессов в соответствии с принятой архитектурой. Определены ключевые факторы в модели облачных вычислений для систем электронных денег.

Введение

Новые технологии являются самым заметным признаком изменения экономических систем, в т.ч. банковских [1]. Некоторые авторы [2] предполагали, и с этим трудно не согласиться, что в 2018-2020 гг. закончится индустриальная фаза роста мировой экономики, и ее дальнейшее развитие будет осуществляться под все большим воздействием когнитивных факторов и производств, основанных на принципах аддитивных, нано- и биотехнологий. Основными направлениями такого развития будут [3, 4]:

- реализация концепции электронного правительства;
- воплощение идеи «цифрового города», что обусловлено комплексной информатизацией транспорта, ЖКХ и др.;
- воплощение идеи строительства «умного» и экологически безопасного дома, что потребует большого объема новых отделочных и строительных материалов;
- распространение разного рода альтернативных и свободных форм занятости в таких сферах, как бухгалтерские услуги, программирование, творческая деятельность и др.;
- создание многочисленных профессиональных сетей, где потенциальный работодатель размещает заказы.

Современная банковская система развивается в рамках общей стратегии развития и цифровизации Российской Федерации, которая подразумевает [3, 5]:

- создание ИТ решений на базе открытых, модульных платформ;
- создание целевых платформ (ППС, ЕИСПД и др.);

- внедрение современных интеграционных ИТ решений и корпоративной интеграционной сервисной шины;

- внедрение инструментов автоматизации бизнес-процессов (BPM);

- развитие каналов взаимодействия, форматов и интерфейсов;

- организацию полнофункциональных личных кабинетов для клиентов Банка России и внедрение единых стандартов обмена (XBRL и др.).

До сих пор основное внимание в исследованиях электронных денег уделялось удовлетворению требований безопасности; вопросам архитектуры и реализации уделялось ограниченное внимание.

Ключевые требования, которые необходимо учитывать при проектировании распределенной архитектуры, включают: масштабируемость, открытость, неоднородность, безопасность, доступность, и производительность.

Целью данной статьи является анализ особенностей функционирования объектов и процессов компьютерной сети банка и проектирование архитектуры системы электронных денег.

Анализ архитектур систем электронных денег

Важным аспектом проектирования архитектуры системы электронных денег является применимость к реальному контексту и задачам. По сути, чтобы новая схема оплаты была принята профессионалами, она должна либо обеспечить значительное снижение затрат, чтобы оправдать инвестиции в необходимую

новую инфраструктуру, либо предоставить клиентам заметные дополнительные услуги.

Базовая онлайн-архитектура

Базовый стиль онлайн-архитектуры можно рассматривать как своего рода архитектуру по умолчанию, которая систематически использовалась во многих работах для внедрения механизмов и протоколов электронных денег. Базовая онлайн-архитектура обычно включает три типа ролей: Банк, Платательщик и Получатель платежа (рис. 1).

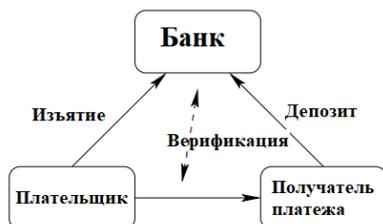


Рисунок 1 – Базовая онлайн-архитектура

В этой конфигурации потребителем является платательщик, который владеет монетами и использует их для покупки товаров или услуг. Получатель платежа получает монеты в обмен на оказанную услугу. Банк означает финансовое учреждение, которое выпускает и продает монеты потребителям, оказывает поддержку в проверке подлинности и целостности монет и выкупает монеты по запросу продавцов.

Электронные деньги, предлагаемые в большинстве современных онлайн-схем, могут быть представлены, в целом, в виде пары $(s, s0)$, где s — ключ, который держится в секрете и $s0$ это уникальный номер, который может быть обнародован и часто упоминается в качестве серийного номера. Обычно $s0$ получается из s с использованием однонаправленной хеш-функции.

Для типичного изъятия платательщик генерирует $(s, s0)$ и отправляет $s0$ вместе с суммой монеты и дополнительной информацией в банк. Банк проверяет личность потребителя и остаток на счете. Если средств достаточно, банк дебетует счет и обновляет базу данных монет, перечисляя $s0$ в качестве эталона для действительной монеты.

Для оплаты заказчик отправляет монету $(s, s0)$ получателю платежа, который проверит действительность монеты, отправив ее в банк. Это позволяет избегать двойных расходов. В случае, если клиент пытается повторно использовать электронные деньги в другом месте, операция будет отклонена, потому что пароль будет недействителен.

Таким образом, получение права собственности на монету гарантирует получателю монеты, что ее действительность подтверждается банком, и что он является единственным владельцем.

Базовая онлайн-архитектура по своей сути является централизованной архитектурой, в которой большая часть коммуникаций проходит через банк. Постоянное участие банка делает его узким местом в производительности и единой точкой отказа, что является важным недостатком в распределенной среде. С другой стороны, это дает преимущество с точки зрения безопасности, поскольку позволяет обнаруживать двойные траты в режиме реального времени. Однако добиться невозможности отслеживания платежей с помощью базовой архитектуры может оказаться очень сложной задачей.

В некоторых предложениях базовая архитектура расширена за счет введения четвертой роли, которую играет доверенная третья сторона, например Центральный банк. В этих случаях некоторые обязанности банка передаются центральному банку, например, выпуск или публикация монет. Иногда такое распределение ответственности не только повышает эффективность системы за счет снижения нагрузки на банк, но и позволяет реализовать некоторые необходимые свойства, такие как анонимность или справедливость.

Процедура оплаты, используемая в этой схеме, аналогична описанной выше, с той разницей, что в этом случае монеты выпускаются отдельной организацией, отличной от банка - Эмитентом. Но сговор между банком и эмитентом, что не исключено, может привести к раскрытию личности потребителя.

То же самое можно сказать и о свойстве справедливости, которое зависит от готовности участников, таких как банк или эмитент, соблюдать протокол. Кроме того, хотя предложенная схема позволяет повторно использовать монеты в нескольких платежах перед их погашением, участие банка в этих операциях ставит под сомнение утверждение о переносимости.

Можно смягчить ограничения онлайн-платежей (например, единая точка отказа и узкое место в производительности) путем распределения накладных расходов на проверку монет между несколькими Эмитентами.

В целом базовая онлайн-схема предоставляет несколько практических преимуществ, когда речь идет о реализации ключевых функций безопасности, таких как предотвращение двойных расходов в режиме реального времени.

Однако это связано с потерей эффективности и масштабируемости из-за центральной роли, которую играет банк. Альтернативы могут состоять, например, в распределении роли, которую играет банк, между несколькими игроками или просто в устранении или уменьшении центрального положения, которое играет банк.

Базовая оффлайн-архитектура

Базовые автономные схемы возникли в результате попытки устранить некоторые недостатки, отмеченные ранее в онлайн схемах, за счет большей гибкости и автономии транзакций с электронными деньгами. Базовые автономные схемы также основаны на трехсторонней модели с теми же ролями, что и в предыдущей модели. Основное различие между обеими моделями заключается в том, что платежные операции происходят только между получателем и плательщиком, без участия банка.

Таким образом, банк больше не является узким местом в производительности и единой точкой отказа. Однако двойные траты могут быть обнаружены только постфактум, что может иметь некоторые негативные последствия.

Поскольку повышенная гибкость и автономия базовых автономных схем достигается за счет безопасности, заключающейся в обнаружении двойных расходов. Основное внимание в этих работах уделялось разработке надежных механизмов безопасности (например, схема со слепой подписью Чаумана [7] или улучшение через протоколы Шнорра, основанные на использовании доказательств с нулевым разглашением), которые устраняли бы это ограничение.

К сожалению, в целом эти механизмы безопасности оказались громоздкими и сложными, что фактически «свело на нет» прирост производительности и масштабируемости, достигнутый за счет предложенных схем. Таким образом, в распределенной обработке, где одинаково важны как безопасность, так и эффективность, необходимо найти правильный баланс, используя соответствующие компромиссы при разработке и развертывании базовых автономных схем.

Базовая переносимая архитектура

В базовой переносимой архитектуре участвуют как минимум четыре типа участников (рис. 2): Первоначальный плательщик, Конечный получатель, Платежный посредник, и Банк.

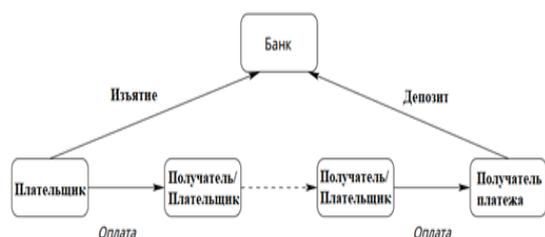


Рисунок 2 – Базовая переносимая архитектура

Первоначальный плательщик, Конечный получатель, и Банк играют ту же роль, что и в предыдущих схемах. Платежный посредник - это агент, который будет играть роль Получателя платежа или Плательщик по типу сделок.

Например, одна из первых переносимых схем включает банк B и n физических лиц $\{C_i / 1 \leq i \leq n\}$, которые могут играть роли потребителей или торговцев. C_1 берет монету из банка B и покупает какой-то предмет с C_2 , передав ему монету. Позже, C_2 повторно использует ту же монету при покупке с C_3 . Тот же процесс повторяется несколько раз, переходя от C_i к C_{i+1} . Окончательно после получения монеты последний человек в цепочке C_n , положит ее в банк.

Следует обратить внимание, что все промежуточные шаги в этой последовательности транзакций не используют обращение в банк. Хотя этот подход предлагает подходящую почву для автономной и распределенной обработки, он может обеспечить значительную масштабируемость и проблемы с производительностью.

Основная проблема при разработке переводимой схемы заключается в обнаружении двойных расходов. Поскольку передаваемые схемы по своей сути работают в автономном режиме, обнаружение двойных расходов происходит постфактум. Поэтому, если нет ограничения на количество разрешенных переводов, стоимость мошеннических транзакций (обнаруженных постфактум) может стать значительной. Таким образом, передаваемые схемы нуждаются в некоторых механизмах отслеживания для выявления мошенников и, как таковые, не могут одновременно гарантировать полную анонимность и безопасность.

Для решения данной проблемы можно использовать модель, состоящую из двух частей:

- фиксированный компонент, подписанный эмитентом, основанный на традиционном формате электронных денег;
- переменный компонент, подписанный во время транзакции, который записывает информацию о транзакции для обеспечения прослеживаемости.

Переменная составляющая позволяет защититься от двойных расходов за счет того, что личность каждого участника фиксируется в монете, в списке транзакций. Каждый элемент в списке транзакций состоит из фиксированного числа пар, полученных в результате разделения секрета на основе личности участника.

Для обеспечения конфиденциальности и выявления возможных двойных расходов, список случайным образом скрывается таким образом, что будущие участники не могут узнать личность предыдущих участников. Кроме того, в случае двойной траты вероятность того, что одни и те же предметы были случайно скрыты, будет низкой. С одной парой на элемент транзакции вероятность обнаружения мошенничества составляет 50 %. В то же время для достижения

вероятности обнаружения мошенничества, равной 98 %, потребуется 6 пар.

Этот способ обнаружения двойных расходов очень интересен, но для его реализации требуется доверенная третья сторона, называемая в модели устройством точки продажи (POS). Оно отвечает за сокрытие и подписание второй части монеты. Рассмотренную модель можно отнести к категории работающих в автономном режиме в том смысле, что присутствие банка не требуется, но все же необходимо присутствие внешней стороны, которая не является ни плательщиком, ни получателем платежа.

Таким образом, защита от двойных расходов может быть достигнута только за счет компромисса с анонимностью. Кроме того, при ее реализации могут резко возрасти основные проблемы масштабируемости и производительности.

Распределенная архитектура

Учитывая, что в онлайн-схемах банк является основным узким местом, а офлайн-платежи представляют собой способ устранить узкое место, созданное банком, предложен альтернативный подход, который заключается в распределении банковских обязанностей (рис.3).

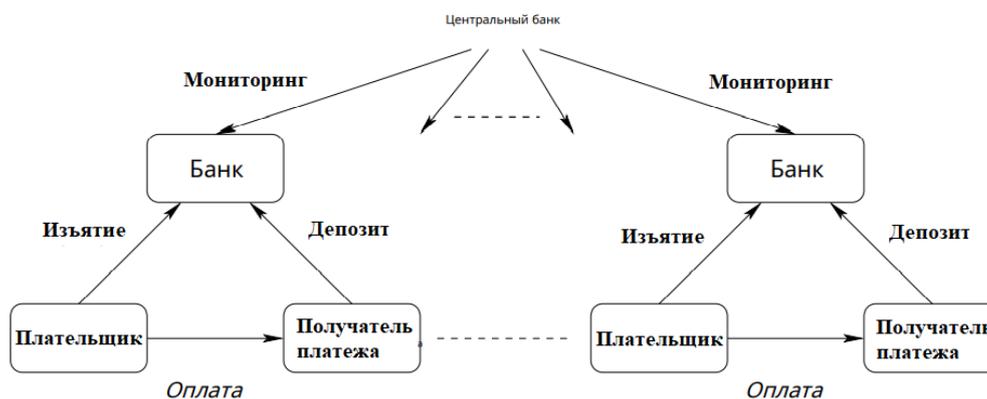


Рисунок 3 – Распределенная архитектура

Модель распределенного банкинга основана на том, что электронные платежные системы в настоящее время разрабатываются и управляются банками отдельно и независимо друг от друга. Т.е. каждый банк поддерживает свой собственный платежный шлюз и инфраструктуру аутентификации [8]. Это не лучший способ оптимизировать использование ресурсов и, как правило, усложняет клиринговые операции, необходимые для сверки транзакций, выполняемых в разных банках.

Модификация базовой трехсторонней модели с учетом того факта, что транзакции могут осуществляться через большую группу банков, контролируемый центральным банком (см. рис. 3) предполагает расширение схемы групповой подписи. Схемы групповых подписей позволяют члену группы подписывать от имени остальных членов группы, не раскрывая личность подписавшего и делая невозможным связать две разные подписи, выданные одним и тем же членом группы. Однако в случае разногласий в схеме участвует назначенный член группы, который может определить лицо, подписавшее документ.

Положительная особенность групповых подписей заключается в том, что проверку подписи можно выполнять с использованием одного открытого ключа группы. К сожалению, при начальных схемах размер открытого ключа

имеет тенденцию к росту вместе с размером группы, что неприемлемо в ситуациях, когда требуется масштабируемость.

Позже была разработана система, позволяющая избегать роста размера ключа. Предложенная схема отличается не только тем, что позволяет нескольким банкам распределять электронные деньги, но и позволяет скрыть не только личность потребителя, но и его банк.

В предлагаемой схеме рассматриваются четыре роли: потребитель, продавец, банки, образующие группу, и менеджер группы, которым может быть, например, центральный банк. Чтобы купить монету, потребитель сначала генерирует монету и отправляет ее в свой банк для подписи. Банк снимает номинал монеты со счета потребителя, вслепую подписывает монету и отправляет ее потребителю.

Существует еще целый ряд архитектур, позволяющих существенно оптимизировать данный процесс.

Сервис-ориентированная архитектура (SOA) [9] возникла как метод проектирования, который позволяет поставщикам программного обеспечения создавать/обертывать различные программные системы в виде доступных сервисов, которые могут быть легко опубликованы и доступны для деловых партнеров и клиентов и, следовательно, освобождают различные модули и системы от

одинаковых функций. По сути, в SOA всегда задействованы три основных участника: поставщик услуг, потребитель услуг, и сервисный реестр.

Поставщик услуг отвечает за создание, публикацию и обслуживание открытых сервисов.

Потребитель услуг несет ответственность за поиск услуг, отвечающих его деловым/техническим потребностям, с целью их использования.

Реестр услуг действует как посредник (база данных), который позволяет поставщику услуг регистрировать созданные услуги со всей необходимой информацией метаданных и инструкциями, которые могут помочь потребителям легко найти и использовать необходимую услугу.

SOA архитектура зависит от использования способов авторизации и веб-сервисов для совместной интеграции доступных (современных и устаревших) систем. Процесс интеграции разделен на две задачи [9]:

- интеграция интерфейсных систем вместе путем предоставления внутренних функций в виде набора веб-приложений;

- интеграция клиентских систем с серверной системой через интеграционную шину, которая будет действовать как единая точка доступа для всех запросов, поступающих от фронтенд-систем.

На рис. 4 приведен пример модели бизнес-процесса «Клиент выбирает продукт или услугу».

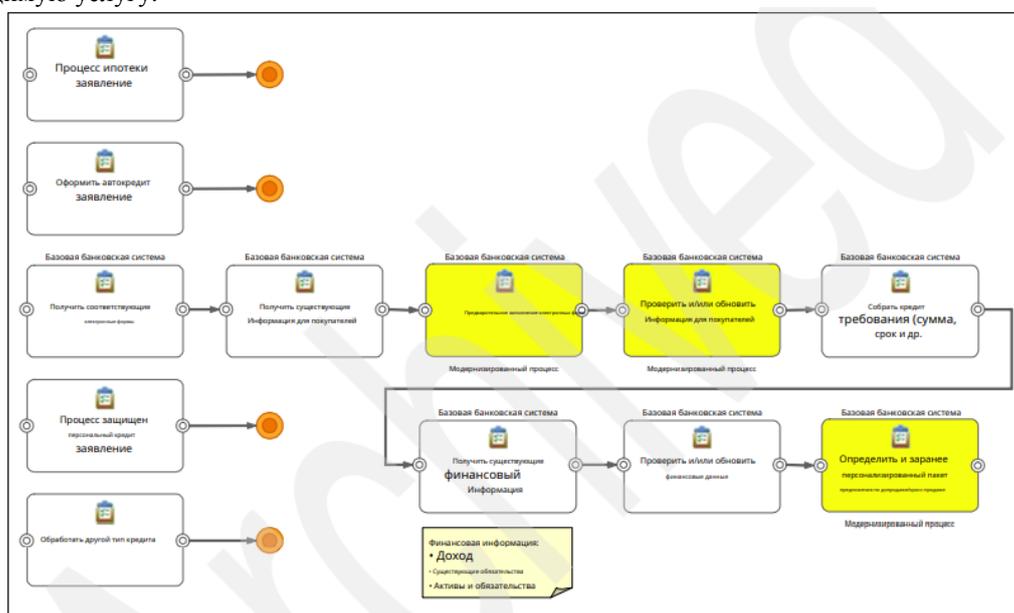


Рисунок 4 – Пример модели бизнес-процесса «Клиент выбирает продукт или услугу»

Использование облачных технологий

С точки зрения инфраструктуры можно выделить возможности, относящиеся к сервисной инфраструктуре и инструментам, которые могут предложить техническую основу для внедрения облака. В облачной инфраструктуре можно назвать такие элементы, как сервисы, предоставление и модель упаковки. С другой стороны, с точки зрения архитектуры есть возможности, которые определяют всю архитектуру и различные указания для включения в облако.

Бизнес- и стратегический компонент модели облачных вычислений позволяет реализовать облачную инициативу. Он охватывает такие элементы, как бизнес-инновации, желаемые выгоды, принципы координации, ожидаемые затраты.

Возможности, которые приобретают важное значение в облачной инициативе, представлены соглашениями о выборе услуг и

уровне обслуживания. Ключевые факторы в модели облачных вычислений могут быть определены в таких компонентах, как: гибкость, снижение барьера для входа, стоимость и эффективность.

Традиционное развертывание приложения с административной точки зрения предполагает указание, закупку, настройку и развертывание аппаратного компонента, развертывание базы данных и приложения, а также настройку параметров.

Облачное развертывание можно охарактеризовать с точки зрения платформы, управляемой пользователем (платформа как развертывание службы) и с точки зрения портала развертывания: запрос на развертывание приложения, корректировка мощности по мере изменения спроса, отключение приложения, когда оно не требуется.

Облачное развертывание является распределенным, поскольку облачные системы

являются важной платформой для распределенных приложений.

В 2011 г. Р. Mell и Т. Grance предложили четыре модели развертывания как часть облачной модели для доступа к общему пулу вычислительных ресурсов: Частное облако, Облако сообщества, Публичное облако, Гибридное облако [10].

Для большинства банков первым крупным шагом в облачные вычисления станут частные облака. Созданы частные облака услуги связи для внутренних бизнес-пользователей. Эти сервисные компоненты очень эластичны и непрерывно расширяются. Банки могут заключать контракты по мере необходимости для удовлетворения требований к уровню обслуживания.

Технологии, лежащие в основе облака и относящиеся к ресурсам пула, представлены виртуализацией, кластеризацией и сеткой. Виртуализация серверов и кластеризация обеспечивают объединение ресурсов и эластичную масштабируемость. Виртуализация определяет один ресурс как несколько виртуальных ресурсов.

Технология кластеризации делает один ресурс одним виртуальным ресурсом. Преимуществами этих технологий являются высокая эффективность, качество обслуживания и гибкость.

У клиентов есть выбор: развертывать эти технологии публично или использовать частные облака. Облачная технология упрощает динамическую замену узлов в кластере. Это позволяет легко и эффективно поддерживать уровни обслуживания при дальнейшем снижении затрат за счет улучшения оптимизация.

В то время как некоторые крупные банки с оптимизмом смотрят на внедрение облачных технологий, другие проявляют осторожность и ждут ответов на вопросы безопасности и регулирования.

В ходе одного из опросов специалистов по банковским технологиям [11] банкиров, выразивших интерес к облачным вычислениям, спросили, почему их это волнует. Большинство, 73%, указали на способность быстро удовлетворять потребности пользователей и масштабироваться в облаке.

Крупные организации могут воспользоваться этими услугами так же, как малые или начинающие предприятия. В то время как преимущество доступа к вычислительным услугам без больших капитальных затрат является очевидной привлекательностью для небольших компаний, крупные компании также могут извлечь выгоду из переноса некритичных для бизнеса вычислительных мощностей и приложений, не требующих большого объема

данных, в облако. Некоторые из основных отмеченных преимуществ:

- лучший денежный поток и большая финансовая прозрачность, так как основное внимание уделяется операционным расходам, а не капитальным затратам;
- быстрое предоставление и эластичное масштабирование услуг;
- позволяет ИТ-отделу сосредоточиться на компетенциях, которые являются ключевыми для бизнеса;
- экологические преимущества [11].

Заключение

Модель распределенного банкинга пытается устранить узкое место в производительности, созданное банком, путем перераспределения и оптимизации выполняемых задач. Она обеспечивает лучшую масштабируемость по сравнению с базовой онлайн-моделью, сохраняя при этом тот же уровень безопасности.

Подход SOA в процессе интеграции означает не только создание набора веб-сервисов, но и способ, которым эти веб-сервисы будут встроены, чтобы позволить максимальному количеству потребителей их использовать, т. е. какие компоненты будут представлены как сервисы, структура этих услуг и то, как потребители смогут ими пользоваться.

Литература

1. Мальчева, Р. В. Компьютерные технологии – основа цифровой экономики // Бизнес-инжиниринг сложных систем: модели, технологии, инновации. Сборник материалов III международной научно-практической конференции. – Донецк: ДОННТУ, 2018. - С. 102-105.
2. Паньшин, Б. Цифровая экономика: особенности и тенденции развития [Электронный ресурс]. – Режим доступа: <https://cyberleninka.ru/article/n/tsifrovaya-ekonomika-osobennosti-i-tendentsii-razvitiya>
3. Терещенко, К. А. Анализ особенностей функционирования и развития объектов и процессов компьютерной сети банка / К. А. Терещенко, Р. В. Мальчева // Материалы XIII Международной научно-технической конференции в рамках VIII Международного Научного форума Донецкой Народной Республики ИУСМКМ-2022. - Донецк, ДОННТУ, 2022. – С. 392-395.
4. Дремач, К. Как банки подстраиваются под ситуацию [Электронный ресурс]. – Режим доступа: https://www.tadviser.ru/index.php/Конференция:Конференция_ИТ_в_банках_2020?
5. Основные направления Стратегии ИТ Банка России [Электронный ресурс]. – Режим

доступа: https://ib8.ib-bank.ru/files/files/2016/03_kruchkov.pdf

6. Чеботарев, В. А. Проектирование компьютерной системы рейтингования деятельности банков [Электронный ресурс] / В. А. Чеботарев, Н. П. Путивцева // Инновационные аспекты развития науки и техники : материалы X Международной научно-практической конференции. – Режим доступа: <https://cyberleninka.ru/article/n/proektirovanie-kompyuternoy-sistemy-reytingovaniya-deyatelnosti-bankov/>

7. Молдовян А. А. Криптография: скоростные шифры / А. А. Молдовян и др. - СПб.: БХВ-Петербург, 2002. - 496 с.: ил. – URL: https://books.4nmv.ru/books/kriptografiya_skorostnye_shifry_3642800.pdf

8. Терещенко, К. А. Проектирование распределенной архитектуры компьютерной сети банка / К. А. Терещенко, Р. В. Мальчева // Информационное пространство Донбасса:

проблемы и перспективы : материалы V Респ. С междунар. Участием науч.-практ. Конф., 27 окт. 2022 г. – Донецк : ГО ВПО «ДонНУЭТ», 2022. – С. 131 -134.

9. Service-oriented architecture [Electronic resource] – URL: <https://learn.microsoft.com/en-us/dotnet/architecture/microservices/architect-microservice-container-applications/service-oriented-architecture>

10. Mell, P. The NIST Definition of Cloud Computing [Electronic resource] / P. Mell, T. Grance // Computer Security Division Information Technology Laboratory National Institute of Standards and Technology, Gaithersburg, MD. - URL: <http://dx.doi.org/10.6028/nist.sp.800-145>.

11. BS&T Survey: Banks Take to Cloud Computing [Electronic resource] – URL: <https://banktech.com/infrastructure/bsandt-survey-banks-take-to-cloud-computing/d/d-id/1293984d41d.html>

Мальчева Р. В., Терещенко К. А. Проектирование архитектуры системы электронных денег. Рассмотрены базовые онлайн и офлайн архитектуры систем для внедрения механизмов и протоколов электронных денег. Рассмотрены основные схемы передачи платежей и проанализированы их узкие места. Предложен альтернативный подход, состоящий в распределении банковских обязанностей. Описана сервис-ориентированная архитектура SOA – многоуровневая архитектура, позволяющая оптимизировать как внутренние, так и внешние процессы. Выполнено моделирование бизнес-процессов в соответствии с принятой архитектурой. Определены ключевые факторы в модели облачных вычислений для систем электронных денег.

Ключевые слова: архитектура, электронные деньги, моделирование, облачные технологии

Malcheva R. V., Tereshchenko K. A. Designing the architecture of the electronic money system. The basic online and offline architectures of systems for the implementation of mechanisms and protocols of electronic money are considered. The main payment transfer schemes are considered and their bottlenecks are analyzed. An alternative approach is proposed, consisting in the distribution of banking responsibilities. The service-oriented architecture of SOA is described – a multi-level architecture that allows optimizing both internal and external processes. Modeling of business processes in accordance with the accepted architecture is performed. The key factors in the cloud computing model for electronic money systems are identified.

Keywords: architecture, electronic money, modeling, cloud computing

*Статья поступила в редакцию 05.03.2023
Рекомендуется к публикации профессором Аноприенко А.Я.*

УДК 62-5, 681.5.015, 004.942

Синтез и моделирование управления нестационарными участками шахтной вентиляционной сети

О. Ю. Чередникова, Ю. С. Достлев
Донецкий национальный технический университет
e-mail: olga.donntu@gmail.com

Аннотация

Предложен универсальный закон управления участками шахтной вентиляционной сети, газодинамические свойства которых описываются минимально- и неминимально-фазовыми дифференциальными уравнениями. Синтезирована математическая модель системы управления участками. Определены оптимальные значения параметров настройки предложенного регулятора расхода воздуха системы управления. Аналитическими исследованиями и методами математического моделирования совместно и согласовано показано, что предложенный закон управления обеспечивает корректную обработку заданной уставки при отсутствии загазовывания участков в переходном процессе.

Введение

Важнейшей задачей оптимизации работы шахтной вентиляционной сети является автоматизация управления вентиляцией участков [1,2]. К текущему моменту времени имеется множество различных постановок задачи поиска оптимального воздухораспределения, разработано множество различных подходов и методов оптимизации воздухораспределения.

Шахтными экспериментами и теоретическими исследованиями показано, что динамические свойства участков значительно изменяются при развитии горных работ на участках. На незначительных объемах выработанного пространства динамические свойства участка как объекта управления описываются дифференциальными уравнениями минимально-фазового типа [3, 4].

Установлено, что при значительной отработке пласта и больших объемах выработанного пространства на участке динамические свойства участка описываются дифференциальными уравнениями неминимально-фазового типа [3, 5].

Шахтными экспериментами и теоретическими исследованиями показано, что эффективное и безопасное по концентрации метана управление на участках минимально-фазового типа обеспечивается при использовании классических законов управления [3]. Однако актуальной научной задачей является синтез безопасной по концентрации метана системы управления участками неминимально-фазового типа, решению которой посвящено настоящее исследование.

Постановка задачи

В области минимально-фазовых характеристик переходные процессы по концентрации метана на участках описываются дифференциальным уравнением вида [4]:

$$T \frac{dS}{dt} + S = -\tau \frac{dG}{dt} - G, \quad (1)$$

где $S = (C - C_o) / C_o$, $G = (Q - Q_o) / Q_o$ – безразмерные относительные значения концентрации метана и расхода воздуха на выходе участка, соответственно; C_o , Q_o – концентрация участка и расход воздуха в установившемся режиме, соответственно; T , τ – постоянная времени участка и коэффициент пропорциональности дифференцирующей составляющей расхода воздуха – управляющего воздействия, соответственно.

Математической модели (1) равносильно дифференциальное уравнение:

$$\frac{d}{dt}(TS + \tau G) = -G. \quad (2)$$

В силу этого при $G(t < 0) = 0$, $G(t \geq 0) = G_K = const$, $S(t < 0) = 0$, согласно (2), получим:

$$S(t = 0) = -(\tau/T) \cdot G_K, \quad (3)$$

где $S(t = 0)$ – скачок (или всплеск) концентрации метана при форсированном изменении дебита воздуха на участке $G(t \geq 0) = G_K$.

После фиксации дебита воздуха на уровне $G = G_K = const$ при $t > 0$ переходные процессы по

концентрации метана участка (1) описываются дифференциальным уравнением:

$$T(dS/dt) + S = -G_K, \quad (4)$$

где, согласно (3), $S(t=0) = -(\tau/T) \cdot G_K$.

Следовательно, при скачкообразном изменении режима вентиляции на участке (1) переходные процессы по концентрации метана на участке, согласно (4), описываются выражением [6]:

$$S(t \geq 0) = -(\tau/T) \cdot G_K - G_K(1 - \exp(-t/T)), \quad (5)$$

где $S(t \rightarrow \infty) = (-G_K)$ – установившееся значение концентрации метана на участке при расходе воздуха $G(t \geq 0) = G_K$; $(-\tau/T) \cdot G_K$ – начальное значение концентрации метана $S(t=0)$ при скачке дебита воздуха $G(t=0) = G_K$;

$$\text{Sign } S(t=0) = \text{Sign } S(t \rightarrow \infty). \quad (6)$$

Таким образом, согласно (5) и (6), на участках минимально-фазового типа даже при скачкообразном изменении расхода воздуха переходные процессы по концентрации метана имеют однополярный характер, что гарантирует безопасное управление участком при произвольном законе изменения дебита воздуха.

В области неминимально-фазовых параметров участков, переходные процессы по концентрации метана описываются дифференциальным уравнением [5]:

$$T \frac{dS}{dt} + S = \tau \frac{dG}{dt} - G. \quad (7)$$

Уравнению (7) равносильно соотношение

$$\frac{d}{dt}(TS - \tau G) = -G.$$

Следовательно, при $G(t < 0) = 0$, $S(t < 0) = 0$, $G(t \geq 0) = G_K = \text{const}$ получим:

$$S(t=0) = (\tau/T) \cdot G_K, \quad (8)$$

где $S(t=0)$ – всплеск концентрации метана на участке при скачкообразном изменении расхода воздуха; $S(t < 0) = 0$, $G(t < 0) = 0$ – начальные уровни концентрации метана и дебита воздуха на участке.

После фиксации дебита воздуха на постоянном уровне $G = G_K$ переходные процессы по концентрации метана на участке (7) описываются, согласно (8), дифференциальным уравнением:

$$T \cdot (dS/dt) + S = -G_K, \quad (9)$$

где $S(t=0) = ((\tau/T) \cdot G_K)$ – начальное условие.

Таким образом, согласно (9), при форсированном изменении расхода воздуха

временные диаграммы участка описываются выражением [6]:

$$S(t \geq 0) = ((\tau/T) \cdot G_K) \cdot \exp(-t/T) - G_K(1 - \exp(-t/T)), \quad (10)$$

где $((\tau/T) \cdot G_K)$ – начальный всплеск концентрации метана при скачке дебита воздуха $G(t \geq 0) = G_K$; $S(t \rightarrow \infty) = (-G_K)$ – конечный уровень концентрации метана на участке;

$$\text{Sign}((\tau/T) \cdot GK) \neq \text{Sign}(t \rightarrow \infty).$$

Следовательно, при изменении расхода воздуха на участках неминимально-фазового типа, согласно (10), всплеск и установившиеся значения концентрации метана являются разнополярными, что может вызвать загазирование участка даже при увеличении расхода воздуха. В силу этого на участках неминимально-фазового типа для обеспечения безопасного управления требуется использование специальной автоматической системы регулирования режимов вентиляции участка.

Синтез и моделирование безопасного управления проветриванием участков неминимально-фазового типа

Для формирования желаемого управления концентрацией метана на участках неминимально-фазового типа используем закон управления:

$$G = -(K_N + \frac{1}{T_U p})(S_Z - S) = (K_N + \frac{1}{T_U p})(S - S_Z), \quad (11)$$

где K_N , T_U – параметры регулятора расхода воздуха; $K = (1 + (K_N \cdot T_U) p)$; S_Z , S – заданное и текущее значение концентрации метана на участке, соответственно; p – оператор дифференцирования.

Тогда, согласно (7) и (11), переходные процессы в замкнутой системе управления вентиляцией участка будут описываться передаточной функцией:

$$S = \frac{K}{T_U p} (S - S_Z) \frac{p-1}{Tp+1}. \quad (12)$$

В силу этого получим:

$$T \cdot T_U \dot{S} + T_U \dot{S} = K (S - S_Z)(\tau p - 1) = K \tau (\dot{S} - \dot{S}_Z) - K (S - S_Z) = K \tau \dot{S} - K \tau \dot{S}_Z + KS - KS_Z - KS,$$

$$\begin{aligned} \text{где } K \tau \dot{S} &= (1 + T_U K_N p) \tau \dot{S} = \tau \dot{S} + T_U K_N \tau \dot{S}; \\ -K \tau \dot{S}_Z &= -(1 + T_U K_N p) \tau \dot{S}_Z = -\tau \dot{S}_Z - T_U K_N \tau \dot{S}_Z; \\ -KS &= -(1 + T_U K_N p) S = -S - T_U K_N \dot{S}; \\ KS_Z &= (1 + T_U K_N p) S_Z = -S_Z - T_U K_N \dot{S}_Z. \end{aligned}$$

Откуда следует:

$$T \cdot T_U \ddot{S} + T_U \dot{S} = \tau \dot{S} + T_U K_N \tau \ddot{S} - \tau \dot{S}_Z - T_U K_N \tau \dot{S}_Z - S - T_U K_N \dot{S} + S_Z + T_U K_N \dot{S}_Z.$$

После преобразований имеем:

$$TT_U \ddot{S} \left(1 - K_N \frac{\tau}{T}\right) + T_U \dot{S} \left(1 + K_N - \frac{\tau}{T_U}\right) + S = S_Z + T_U \dot{S}_Z \left(K_N - \frac{\tau}{T_U}\right) - T_U K_N \tau \dot{S}_Z. \quad (13)$$

Для обеспечения устойчивости системы управления (12), согласно (13), примем [3]:

$$T_U \gg \tau, \quad \frac{\tau}{T_U} \ll K_N < 1. \quad (14)$$

При ограничениях (14) уравнение (13) вырождается в соотношение:

$$TT_U \ddot{S} + T_U \dot{S}(1 + K_N) + S = S_Z + T_U \dot{S}_Z K_N - T_U K_N \tau \dot{S}_Z. \quad (15)$$

Уравнение (15) равносильно соотношению

$$\frac{d}{dt}(TT_U \dot{S} + T_U S(1 + K_N) - T_U S_Z K_N + T_U K_N \tau \dot{S}_Z) = S_Z - S. \quad (16)$$

Для исследования динамических свойств системы управления (15)-(16) рассмотрим режим работы схемы при следующих условиях:

$$S_Z(t \geq 0) = const, \quad S_Z(t < 0) = 0, \quad \dot{S}(t < 0) = 0, \quad \dot{S}_Z(t < 0) = 0.$$

Тогда, согласно (16), получим:

$$S(t = 0) = \frac{K_N}{1 + K_N} S_Z(t = 0). \quad (17)$$

Таким образом, согласно (15) и (17), имеем:

$$sign S(t=0) = sign S(t \rightarrow \infty), \quad \text{где } S(t \rightarrow \infty) = S_Z.$$

Таким образом, при скачкообразном изменении задающего воздействия $S_Z(t=0)$ начальный всплеск концентрации метана на участке $S(t=0)$ и установившееся значение концентрации метана $S(t \rightarrow \infty) = S_Z$ имеют одинаковую полярность, что создает предпосылки безопасного управления участком при отработке заданного режима вентиляции участка в системе управления.

Траектория движения системы (15) при $t > 0$, $S_Z(t > 0) = const$, $\dot{S}_Z(t > 0) = 0$, $\dot{S}_Z(t > 0) = 0$ определяется, согласно (17), дифференциальным уравнением:

$$TT_U \ddot{S} + T_U \dot{S}(1 + K_N) + S = S_Z, \quad (18)$$

где $S(t = 0) = \frac{K_N}{1 + K_N} S_Z$ – начальное условие.

Следовательно, при оптимальных настройках регулятора расхода воздуха (14) характеристическое уравнение системы (18) имеет вид [3]:

$$TT_U r^2 + T_U a_N r + 1 = 0, \quad (19)$$

где $a_N = 1 + K_N$.

В силу этого корни характеристического уравнения системы (19) описываются выражением:

$$r_{1,2} = \frac{-T_U a_N \pm \sqrt{a_N^2 T_U^2 - 4TT_U}}{2TT_U} = \frac{T_U a_N (1 \pm \sqrt{1 - 4T/(a_N^2 T_U)})}{2TT_U} = \frac{-1 \pm \sqrt{d}}{2T/a_N}, \quad (20)$$

где $d = 1 - 4T/(a_N^2 T_U)$.

Следовательно, общее и частное решение уравнения (18), согласно (20), имеет вид:

$$S(t) = A_1 \exp(-t/T_1) + A_2 \exp(-t/T_2) + S_Z, \quad (21)$$

где A_1 и A_2 – постоянные интегрирования;
 $T_{1,2} = \frac{2T/(1+K_N)}{-1 \pm \sqrt{d}}$.

Значение постоянных интегрирования A_1 и A_2 определяются решением системы уравнений в соответствии с (17):

$$S(t = 0) = A_1 + A_2 + S_Z = \frac{K_N}{1 + K_N};$$

$$\dot{S}(t = 0) = -A_1/T_1 - A_2/T_2 = 0; S_Z(t \geq 0) = const,$$

где

$$\dot{S}(t) = -A_1/T_1 \cdot \exp(-t/T_1) - A_2/T_2 \cdot \exp(-t/T_2).$$

Откуда получаем:

$$A_1 = -\frac{T_1}{(T_1 - T_2)(1 + K_N)} S_Z, \quad A_2 = \frac{T_2}{(T_1 - T_2)(1 + K_N)} S_Z. \quad (22)$$

Следовательно, согласно (21)-(22), траектория движения системы управления описывается выражением:

$$S(t) = \left(\frac{1}{(T_1 - T_2)(1 + K_N)} (T_2 \exp(-t/T_2) - T_1 \exp(-t/T_1)) + 1\right) S_Z, \quad (23)$$

и

$$\dot{S}(t) = \frac{S_Z}{(T_1 - T_2)(1 + K_N)} (\exp(-t/T_1) - \exp(-t/T_2)).$$

Таким образом, поскольку согласно (21) $T_1 > T_2$, согласно (23), имеем $sign \dot{S}(t > 0) = sign S_Z$. Поэтому в системе управления (11) обеспечивается безопасное по концентрации метана управление участком неминимально-фазового типа в течение всего переходного процесса отработки заданной уставки $S_Z(t \geq 0) = const$. Результаты моделирования системы управления (11) участком

неминимально-фазового типа (7) показаны на рис.1.

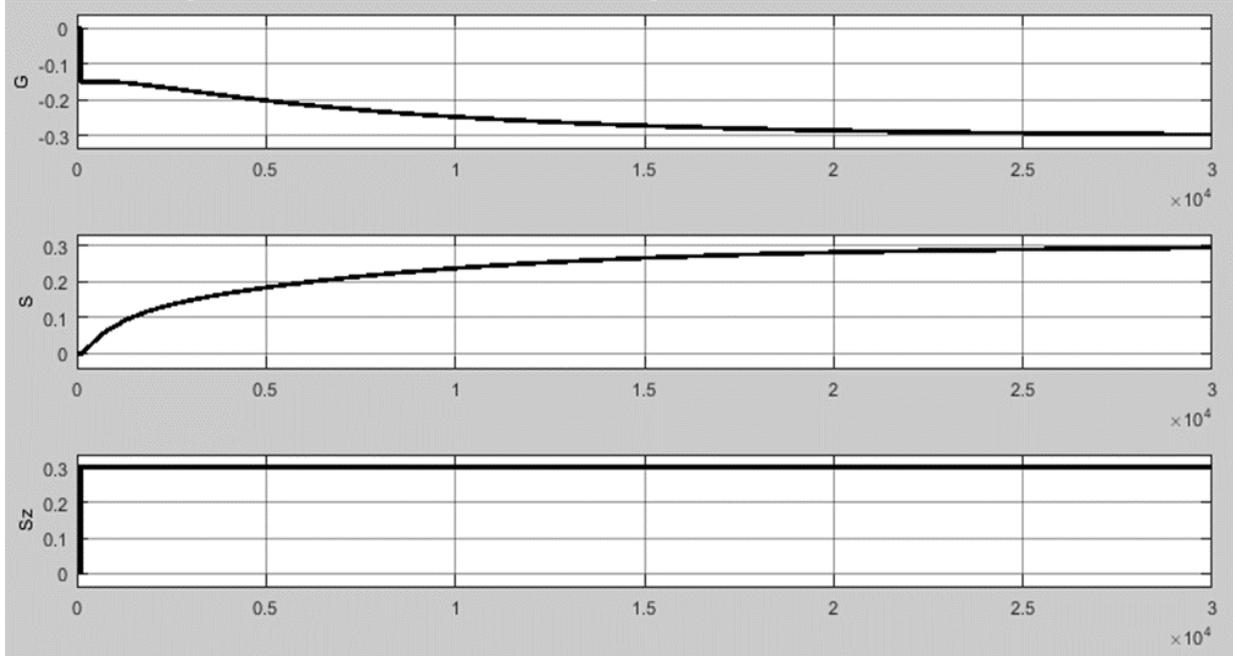


Рисунок 1 – Результаты моделирования процессов управления на участке неминимально-фазового типа при увеличении задающего значения концентрации метана

При этом были использованы следующие параметры участка и модели:
 $\tau = 300 \text{ сек.}, T = 1200 \text{ сек.},$
 $T_U = 20 \tau = 6000 \text{ сек.}, K_N = 0,5.$

Графики переходных процессов достаточно адекватно коррелируют с данными проведенных аналитических исследований, что подтверждает корректность выполненных математических исследований.

Синтез и моделирование системы безопасного управления участком минимально-фазового типа

Для безопасного управления участками минимально-фазового типа используем закон управления (11). Тогда, согласно (1), передаточная функция системы управления участками минимально-фазового типа будет иметь вид:

$$S = -\frac{K}{T_U p} (S - S_Z) \frac{\tau p + 1}{T p + 1}. \quad (24)$$

Преобразовав (24), получим:

$$T \cdot T_U \ddot{S} + T_U \dot{S} = -KS (\tau p + 1) + K S_Z (\tau p + 1),$$

$$\text{где } KS_Z (\tau p + 1) = (1 + T_U K_N p) (\tau p + 1) S_Z = S_Z + (\tau + T_U K_N) \dot{S}_Z + T_U K_N \tau \ddot{S}_Z;$$

$$-KS (\tau p + 1) = -(1 + T_U K_N p) \tau \dot{S} - (1 + T_U K_N p) S = -S - (T_U K_N + \tau) \dot{S} - T_U K_N \tau \ddot{S};$$

Откуда следует:

$$T T_U \ddot{S} \left(1 + K_N \frac{\tau}{T}\right) + T_U \dot{S} \left(1 + K_N + \frac{\tau}{T_U}\right) + S = S_Z + T_U \dot{S}_Z \left(K_N + \frac{\tau}{T_U}\right) + T_U K_N \tau \ddot{S}_Z. \quad (25)$$

При ограничениях (14) согласно (25), имеем:

$$T T_U \ddot{S} + T_U \dot{S} (1 + K_N) + S = S_Z + T_U \dot{S}_Z K_N + T_U K_N \tau \ddot{S}_Z. \quad (26)$$

Уравнение (26) равносильно соотношению:

$$\frac{d}{dt} (T T_U \dot{S} + T_U S (1 + K_N) - T_U S_Z K_N - T_U K_N \tau \dot{S}_Z) = S_Z - S. \quad (27)$$

При $S_Z(t \geq 0) = const, S_Z(t < 0) = 0, S(t < 0) = 0, \dot{S}(t < 0) = 0, \dot{S}_Z(t < 0) = 0$ согласно (17) и (27), получим:

$$S(t = 0) = \frac{K_N}{1 + K_N} S_Z(t \geq 0).$$

Следовательно, согласно (26) и (28), при скачкообразном изменении уставки $S_Z(t \geq 0) = const$ временные графики системы управления (24) описываются дифференциальным уравнением (18).

В силу этого даже при форсированном изменении задающего воздействия процессы управления в системе управления участком

минимально-фазового типа (24) будут, согласно (23), устойчивыми и безопасными по концентрации метана. Результаты

моделирования управления участком минимально-фазового типа при изменении режимов проветривания показаны на рис.2.

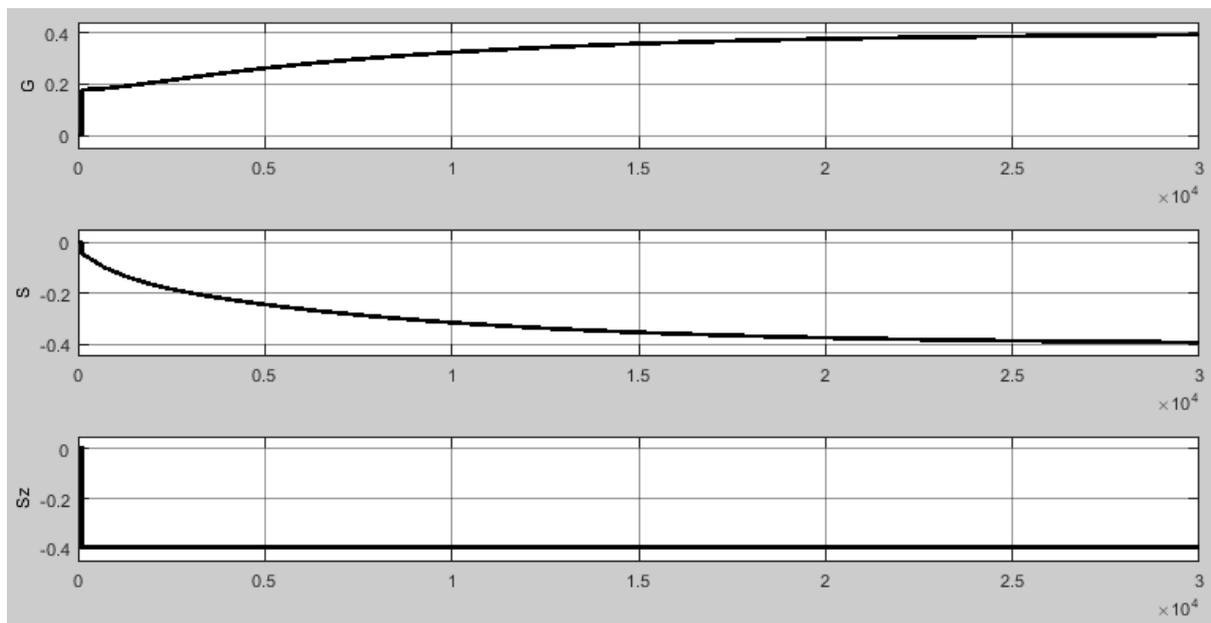


Рисунок 2 – Результаты моделирования управления участком минимально-фазового типа при уменьшении задающего воздействия

Переходные процессы в модели системы управления (24) соответствуют аналитическому решению (23), что подтверждает корректность проведенных аналитических исследований.

Заключение

В работе использованы совместно аналитические исследования и методы математического моделирования [7-10].

Основные результаты работы сводятся к следующему:

- для всей области определения газодинамических параметров участков шахтной вентиляционной сети предложен универсальный алгоритм управления вентиляцией участка;
- синтезирована математическая модель предложенной системы управления участками;
- обоснованы оптимальные значения параметров регулятора расхода воздуха системы управления участками;
- аналитическими исследованиями и методами математического моделирования совместно и согласованно показано, что предложенный закон управления обеспечивает корректную отработку задающего воздействия при отсутствии загазования участков в переходном режиме.

Литература

1. Автоматизированное управление вентиляцией шахт и рудников. Проблемы,

современный опыт, направления совершенствования / М. А. Семин, Е. Л. Гришин, Л. Ю. Левин, А. В. Зайцев // Записки Горного института, 2020. - Т. 246. - С. 623-632.

2. Управление вентиляционными потоками в горных выработках подземных рудников на основе математического моделирования аэродинамических процессов / С. А. Козырев, А. В. Осинцева, П. В. Амосов. — Апатиты: КНЦ РАН, 2019. — 114 с.: ил.

3. Бесекерский, В. А. Теория систем автоматического управления / В. А. Бесекерский, Е. П. Попов. – Санкт-Петербург: Профессия, 2003. – 752 с.

4. Чередникова, О. Ю. Исследование переходных процессов в системе управления проветриванием угольных шахт в выемочных участках минимально-фазового типа / О. Ю. Чередникова, Ю. С. Достлев // Информатика и кибернетика. – Донецк: ДонНТУ, 2021. - № 1-2 (23-24). – С. 108-114.

5. Чередникова, О. Ю. Исследование переходных процессов в системе управления проветриванием угольных шахт в выемочных участках неминимально-фазового типа / О. Ю. Чередникова, Ю. С. Достлев // Информатика и кибернетика. – Донецк: ДонНТУ, 2022. - №1 (27). – С. 37-43.

6. Бойко, В. А. Основы теории расчета вентиляции шахт / В. А. Бойко, Н. Ф. Кременчуцкий. – М.: Недра, 1978. - 280 с.

7. Ушаков, В. А. Математическое модели-

рование надежности и эффективности шахтных вентиляционных систем / В. А. Ушаков. – Москва: Высшее горное образование, 2003. – 181 с.

8. Пучков, Л. А. Методы и алгоритмы автоматического управления проветриванием угольных шахт / Л. А. Пучков, Л. А., Бахвалов. – 1992. – 399 с.

9. Каледина, Н. О. Вентиляция производ-

ственных объектов / Н. О. Каледина. – М.: Горное образование, 2008. – 193 с.

10. Чередникова, О. Ю. Синтез и исследование методами математического моделирования систем управления проветриванием выемочных участков угольных шахт / О. Ю. Чередникова, Ю. С. Достлев // Информатика и кибернетика. – Донецк: ДонНТУ, 2019. – № 1(15). - С. 65-74.

Чередникова О. Ю., Достлев Ю. С. Синтез и моделирование управления нестационарными участками шахтной вентиляционной сети. Предложен универсальный закон управления участками шахтной вентиляционной сети, газодинамические свойства которых описываются минимально- и неминимально-фазовыми дифференциальными уравнениями. Синтезирована математическая модель системы управления участками. Определены оптимальные значения параметров настройки предложенного регулятора расхода воздуха системы управления. Аналитическими исследованиями и методами математического моделирования совместно и согласовано показано, что предложенный закон управления обеспечивает корректную отработку заданной уставки при отсутствии газозавоза участков в переходном процессе.

Ключевые слова: неминимально-фазовый объект, минимально-фазовый объект, моделирование, система управления, концентрация метана, математическая модель

Cherednikova O. Yu., Dostlev Yu. S. Synthesis and modeling of control of non-stationary sections of a mine ventilation network. A universal control law for sections of a mine ventilation network is proposed, the gas-dynamic properties of which are described by minimal- and non-minimal-phase differential equations. A mathematical model of the site management system has been synthesized. The optimal values of the setting parameters of the proposed air flow controller of the control system are determined. Analytical studies and methods of mathematical modeling jointly and in agreement show that the proposed control law ensures the correct working out of the given setpoint in the absence of gassing of the sections in the transient process.

Key words: non-minimum phase object, minimal phase object, modeling, control system, methane concentration, mathematical model

Статья поступила в редакцию 25.12.2022
Рекомендуется к публикации профессором Павлышом В. Н.

Обзор методов трехмерной реконструкции объектов по изображениям

Ю. А. Якубовская

Донецкий национальный технический университет, г.Донецк

iulia.yakubovskaja@yandex.ru

Аннотация

В статье рассматриваются и анализируются методы трехмерной реконструкции объектов по их изображениям. Приводится их классификация по типу и количеству входящей информации. Произведено описание классифицированных методов, приведены их отличительные черты, выделены достоинства и недостатки, рассмотрены их современные реализации. Приводится сравнительный анализ описанных методов и выбирается оптимальный метод для создания программного решения для реконструкции объектов в условиях неполноты исходных данных.

Введение

В последние годы активно развивается компьютерное зрение, в связи с этим возрастает потребность в трехмерных моделях местности. Они позволяют создавать подробные изображения земной поверхности, включая рельеф, растительность, здания и сооружения, дороги и другие объекты. 3D модели местности широко используются в различных областях, таких как геодезия, картография, геоинформатика, градостроительство. Также такие модели используются историками и архитекторами в решении задач, связанных с реконструкцией исторических и археологических объектов. 3D модели зданий широко используются при реставрации, для определения первоначального облика и размеров объекта.

Еще одним преимуществом 3D моделей является возможность создания виртуальных экскурсий и экспозиций. С помощью таких моделей можно создать реалистичные изображения исторических объектов и показать их посетителям в новом свете с использованием систем виртуальной или дополненной реальности. Это может способствовать более глубокому пониманию истории и культуры, а также повысить интерес к изучению прошлого.

В этой области актуальной и наукоемкой задачей является построение трехмерных моделей таких объектов на основе их изображений.

Следует учитывать, что фотографий утраченных строений может сохраниться малое количество, возможно даже единственный экземпляр.

При этом предъявляются достаточно высокие требования к качеству цифровой модели. Понятие «качество» в данном случае включает в себя две составляющие: метрическую

точность предлагаемых моделей архитектурных сооружений и фотографические свойства их текстур.

Методы трехмерной реконструкции

Один из наиболее широко применимых методов реконструкции геометрических моделей – на основе использования изображений объекта. Фотографические изображения могут быть использованы различных типов: спутниковая съемка, снимки с беспилотников, наземные обычные или панорамные и др. Такие методы относят к фотограмметрии – науке об определении форм, размеров и положении объектов по их фотографиям.

Фотограмметрия достаточно широко используется в различных областях науки, техники, для решения прикладных задач и имеет ряд преимуществ перед другими методами трехмерного моделирования:

- высокая точность: используются алгоритмы, которые по фотографиям позволяют получить точные измерения;
- быстрота: алгоритмы работают быстро и позволяют получать трехмерные модели за достаточно короткое время;
- удобство: не требует специального дорогостоящего оборудования;
- универсальность: может быть применена к широкому спектру объектов;
- безопасность: измерения производятся бесконтактным способом.

Методы создания 3D сцен по их изображениям можно поделить на три группы: по одному изображению; по нескольким изображениям; по видеоряду.

На рисунке 1 приведена классификация методов трехмерной реконструкции объектов по изображениям.

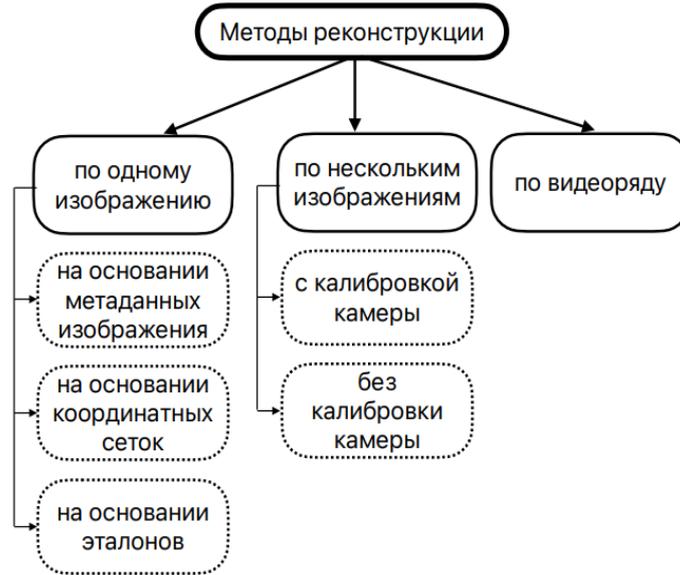


Рисунок 1 – Классификация методов трехмерной реконструкции объектов

Методы реконструкции по одному изображению

Методы реконструкции по одному изображению можно поделить на три класса: на основании фотографии и ее метаданных, на основании координатных сеток, на основании эталонов.

Методы на основании метаданных изображения

Методы, которые используют один снимок и его метаданные, требуют большого количества дополнительной информации о снимке, такие как угол фотографирования, параметры освещенности и тени. Также большинство подобных методов используются для определенных объектов или типов изображений, некоторые из них предъявляют определенные условия к фотосъемке. Таким образом, применение подобных методов затруднительно при отсутствии необходимых сведений.

Сушецкий Д.В. предлагает использовать лидарные изображения высокого разрешения, где в качестве основного источника информации используются интенсивности излучения отраженного света от соответствующих точек земной поверхности [1]. Для получения таких изображений требуется специальное дорогостоящее оборудование, а трехмерные модели получаются довольно низкого качества, малого уровня детализации (рис 2).

Тужилкин А.Ю. в своей работе предлагает модель реконструкции на основе спутниковых изображений и априорной информации [2].

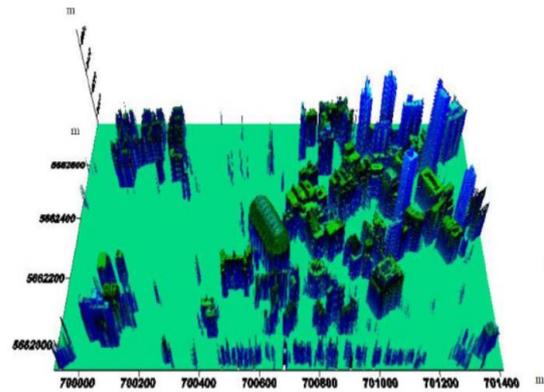


Рисунок 2 – Реконструкция трехмерных объектов на основании комплексных данных лидарных и RGB изображений

Для обнаружения зданий на фото предлагается группировать регионы, которые определяются цветовыми характеристиками, значениями высоты, областью затенения, растительности и т.д.

При реконструкции трехмерных моделей используют геометрические гистограммы (рис.3).

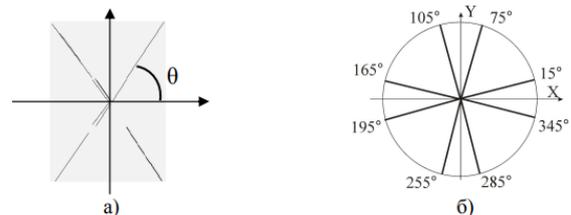


Рисунок 3 – Построение геометрических гистограмм: а) определение угла между отрезком и осью абсцисс в системе координат полигона, б) интервалы разбиения числовой окружности

Для них выбирается система координат, в которой ось ординат совмещена с центральной осью ограничивающего прямоугольника, параллельной большей его стороне (рис. 3а), а также используются неравные интервалы углов (рис. 3б). На основе расстояния Бхаттачария производится сравнение геометрических гистограмм.

Методы на основании координатных сеток

При использовании методов с использованием координатных сеток за основу берется единственное изображение без атрибутивной информации, но существует информация о форме этого объекта и способ соотношения пикселей изображения с местоположением ряда точек этой формы. Для решения задачи на изображение накладывается сетка линий с заданной структурой. Сетки могут быть линейными или нелинейными, плоскими или пространственными, прямыми или наклонными, ортотрансформированными или перспективными.

Шахраманьян М.А предлагает методику генерации трехмерных объектов «правильной» формы, выбирая на изображении опорные точки, так чтоб они образовывали базис пространственной системы координат различного типа в зависимости от формы объекта [3]. На основании этих точек находятся масштабирующие коэффициенты, рассчитывается изображение координатной сетки, которая наносится на изображение. На сетке размечаются характерные точки, которые определяют положения точек в пространстве. Основные этапы построения трехмерной модели в соответствии с данной методикой представлены на рисунке 4.

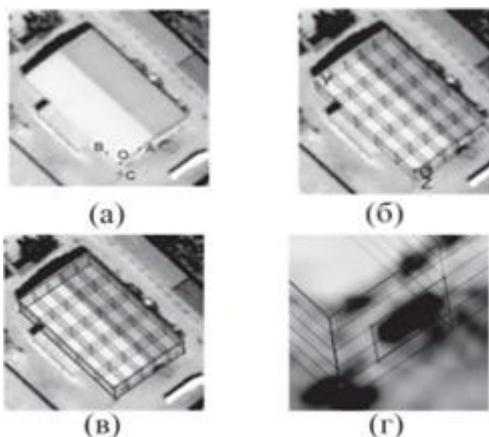


Рисунок 4 – Этапы построения трехмерной модели на основании координатных сеток:
а) выбор опорных точек; б) расчет и построение координатной сетки; в) разметка точек и линий на сетке; г) аффинные преобразования для улучшения точности разметки (увеличение)

Руденко М.П. предлагает алгоритм с использованием метода перспективных масштабов, который позволяет определить относительные натуральные величины элементов объекта. В разработанной системе эксперту предлагается определить количество точек схода на изображении. После этого на изображение наносятся опорные точки, на основании которых строится перспективная сетка и производится построение трехмерной модели (рис. 5) [4].

Для построения элементов, отсутствующих на изображении, но уместных в модели, требуется участие эксперта.

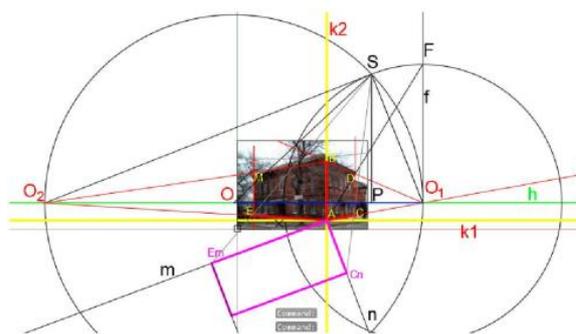


Рисунок 5 – Отыскание относительных натуральных величин основного блока архитектурного сооружения

Методы на основании эталонов

Используя методы на основании эталонов на исходном изображении необходимо задать эталонные объекты, по их известным размерам угловым или линейным. В основе методов лежит расчёт масштабирующих коэффициентов в зависимости от направляющих осей и пространственного разрешения.

Рихтер А.А. предлагает методику построения по объекту-эталоно, для которого известно значение высоты, соответствующей определенному отрезку на изображении, таким образом получив масштабирующий коэффициент. Также предложен альтернативный вариант расчета «по тени» в условиях отсутствия видимости боковых частей расчетного объекта [5].

Также необходимо отметить, что существуют подходы к трехмерной реконструкции объектов, которые объединяют несколько методов из вышеперечисленных, а также интегрируют их с нейросетевыми технологиями. Так, например, в [6] описывается метод построения трехмерных объектов по одному спутниковому изображению, который основан на сочетании традиционных и нейросетевых методов. В работе на основании эталонных объектов (опорных железнодорожных столбов и рельс) оцениваются масштабирующие коэффициенты для оценки высоты и формы

основания строения. Метод основан на применении двух сверточных нейронных сетей, первая из которых выполняет сегментацию и выделение объектов заданных классов, вторая – на сегментированных изображениях извлекает индивидуальные объекты, определяет параметры и строит модель по растровым областям. Данный метод предполагает, что исходные изображения были предварительно обработаны.

Методы реконструкции по нескольким изображениям

Такие методы можно поделить на два класса: методы с калибровкой камер и методы без калибровки. Калибровка камер – это изучение взаимного расположения камер в пространстве, а также их внутренних параметров.

Методы реконструкции с калибровкой камер

Калибровка камер широко используется при реконструкции по стереопаре или при малом количестве снимков. В процессе калибровки необходимо установить точные параметры камер, такие как фокусное расстояние, оптическая ось, геометрическое искажение, углы обзора и т.д. Для калибровки камер по стереопаре необходимо определить положение камер в пространстве, на основании относительных координат сопоставленных точек на изображениях, а также данных полученных с камеры. Калибровка камер при трехмерных реконструкциях является трудоемким процессом и зависит от множества факторов, таких как количество камер, их расположение, условия съемки, наличие «шумов», наличие достаточного количества точек, настройка параметров камеры, размер сцены и т.д. В [7] описывается реконструкция с использованием калиброванных камер по двум изображениям. Подробно рассмотрено вычисление трехмерных координат точки по стереопаре. Сделан вывод, что для получения трехмерных координат необходимо решить ряд следующих задач: задача калибровки, взаимного ориентирования, поиска сопряженных точек.

Методы реконструкции без калибровки камер

В методах, работающих без калибровки камер часто накладываются определенные ограничения на внутренние параметры камер или используются методы, основанные на алгоритмах триангуляции, факторизации матриц, а нахождение координат точек осуществляется с помощью эпиполярной геометрии. Такие алгоритмы, как правило, не используют непосредственно изображение, им необходимы

координаты характерных точек на ряде изображений. Координаты точек в данном случае имеют абстрактные величины, и оценивается их взаимное относительное расположение.

Беляков П. В. предлагает использовать вариационный метод на основе вычисления фундаментальной матрицы и оптического потока [8]. Вместе с этим определяется пространственное положение точек сцены и требование гладкости поверхности. В качестве входных данных используется последовательность изображений, которые получены путем движения камеры по непрерывной траектории, также предполагается, что внутренние параметры камеры известны.

В [9] предложен метод реконструкции без калибровки камер по двум изображениям, который также основывается на вычислении фундаментальной матрицы. В работе используется полиномиальный алгоритм триангуляции для нахождения трехмерных координат, который характеризуется достаточно хорошим быстродействием, а также достаточно высокой точностью. В качестве входной информации используется стереопара, без информации о метрических характеристиках камеры.

Методы реконструкции по видеоряду

Третий тип методов реконструкции – по видео, эти методы предполагают построение геометрических моделей объектов, которые фиксирует движущаяся камера. Такие методы основываются на использовании силуэтов, освещенности и движения. В некоторых работах, для реконструкции по видео из видеоряда просто извлекаются стереопары, в других используется дополнительно специализированное оборудование: дальномеры, системы освещенности, высокая частота съемки, множество камер и т.д. Подобные методы требуют больших материальных затрат и предполагают высокую вычислительную трудоемкость.

Бобков В. А. предлагает метод трехмерной реконструкции в видео-сценах, восстановление структуры и движения. Модель динамического объекта строится путем объединения видов из различных позиций камеры [10]. В рамках данной работы решаются следующие задачи: построение трехмерного точечного представления сцены и выделение из нее объектов; выделение статических и движущихся объектов, вычисление траекторий движения; генерация различных точечных видов движущегося объекта; построение трехмерной поверхностной модели по полученным видам.

В [11] предлагается из видеоряда выбрать два соседних кадра, далее применяются методы эпиполярной геометрии, без калибровки камер описанные выше.

Сравнительный анализ методов реконструкции по изображениям

Сравнительный анализ рассмотренных методов приведен в таблице 1. Основными параметрами для сравнения выбраны: количество входящей информации; необходимость дополнительной или атрибутивной информации об изображении или объекте; необходимость особых условий фотосъемки; необходимость предварительной обработки фото; экспертный контроль. Для реконструкции трехмерных объектов зданий для

исторических и археологических исследований предпочтительны методы, которым необходим минимум входящей информации, так как от утраченного памятника архитектуры может остаться единственное изображение или чертеж. Также, для объектов такого рода практически невозможно предъявлять особые требования к условиям съемки или запрашивать атрибутивную информацию. Поэтому оптимальными методами для трехмерной реконструкции зданий и сооружений памятников архитектуры являются методы на основе одного изображения с использованием координатных сеток.

Таблица 1 – Сравнение методов реконструкции

	Методы реконструкции					
	По одному изображению			По нескольким изображениям		По видео
	Используют метаданные	Используют координатные сетки	Используют эталоны	С калибровкой камеры	Без калибровки камеры	
Количество входящей информации	1 фото	1 фото	1 фото	≥ 2 фото	≥ 2 фото	видео
Необходимость дополнительной или атрибутивной информации	+	-	+	+	-	+
Особые требования к условиям получения изображений	+	-	-	+	-	+
Необходимость предварительной обработки фото	-	-	-	+	+	+
Необходимость экспертного контроля	+	+	+	+	-	+

Выводы

В работе рассмотрены и классифицированы методы реконструкции трехмерных моделей по изображениям. Предложена классификация по входящей информации: методы на основании одного изображения – с использованием метаданных, с использованием эталонов, с использованием координатных сеток; на основании нескольких изображений – с калибровкой камеры, без калибровки; на основании видеоряда.

Оптимальной группой методов для реконструкции при наличии недостаточного количества изображений и неизвестных условиях съемки являются методы на основании одного изображения с использованием координатных сеток. Перспективным направлением разработки является объединение этих методов с нейросетевыми технологиями для повышения уровня автоматизации реконструкции.

Литература

1. Суцевский, Д. В. Выявление и 3-D визуализация искусственных объектов земной поверхности с использованием лидарных и оптических спутниковых изображений // Вестник Херсонского национального технического университета, 2014. - №3 (50).
2. Тужилкин, А. Ю. Модели и алгоритмы реконструкции трехмерных сцен на основе спутниковых изображений и априорной информации: автореф. дис. ... кан. техн. наук. - Нижний Новгород, 2016. - 19 с.
3. Шахраманьян, М. А. Методика построения 3D-модели объектов правильной формы по одному изображению и ее применение в задаче космического мониторинга объектов захоронения отходов / М. А. Шахраманьян, А. А. Рихтер, А. Б. Мурынин, М. Л. Казарян // Информация и космос, 2018. - № 2 (8). - С. 76-81.

4. Руденко, М. П. Усовершенствованный алгоритм синтеза моделей трехмерных объектов по их изображению / М. П. Руденко, А. А. Бабакина, В. В. Карабчевский // Проблемы искусственного интеллекта, 2020. - №1 (16).

5. Рихтер, А.А. Методика построения 3D-моделей ригидных объектов и её применение в моделировании антропогенных территорий по космическим изображениям / А. А. Рихтер, А. Б. Мурынин // Материалы 15-й Всерос. откр. ежегод. конф.: Современные проблемы дистанционного зондирования Земли из космоса. Методы и алгоритмы обработки спутниковых данных (ИКИ РАН-2017, 13-17 ноября). - М.: ИКИ РАН, 2017. - С. 50.

6. Гвоздев, О. Г. Построение трехмерных моделей ригидных объектов по спутниковым изображениям высокого пространственного разрешения с использованием сверточных нейронных сетей / О. Г. Гвоздев, В. А. Козуб, Н. В. Кошелева [и др.] // Исследование Земли из космоса, 2020. – № 5. – С. 78-96

7. Меженин, А. В. Реконструкция трехмерных моделей по растровым изображениям / А. В. Меженин, В. Т. Тозик // Научно-технический вестник информационных технологий, механики и оптики, 2007. - №45.

8. Беляков, П. В. Вариационный метод трехмерной реконструкции по изображениям / П. В. Беляков, М. Б. Никифоров, А. И. Новиков // XXI век: итоги прошлого и проблемы настоящего плюс, 2019. – Т. 8, № 3(47). – С. 107-112.

9. Шепилова, К. М. Метод трехмерной реконструкции сцены в относительных координатах по двум изображениям с неоткалиброванных видеокамер / К. М. Шепилова, А. В. Сотников, А. В. Шипатов, Ю. В. Савченко // Известия высших учебных заведений. Электроника, 2020. – Т. 25. - № 3. – С. 265-276.

10. Бобков, В. А. Пространственная реконструкция объектов в динамической сцене по видеопотоку / В. А. Бобков, А. П. Кудряшов // Подводные исследования и робототехника, 2020. – № 3(33). – С. 11-17.

11. Захаров, А. А. Алгоритм автоматической реконструкции трехмерных сцен по видеопоследовательности / А. А. Захаров, М. И. Ткачук // Информатика, телекоммуникации и управление, 2011. - №3 (126).

Якубовская Ю. А. Обзор методов трехмерной реконструкции. В статье рассматриваются и анализируются методы трехмерной реконструкции объектов по их изображениям. Приводится их классификация по типу и количеству входящей информации. Произведено описание классифицированных методов, приведены их отличительные черты, выделены достоинства и недостатки, рассмотрены их современные реализации. Приводится сравнительный анализ описанных методов и выбирается оптимальный метод для создания программного решения для реконструкции объектов в условиях неполноты исходных данных.

Ключевые слова: фотограмметрия, 3D модель, изображение, трехмерная реконструкция, метод реконструкции.

Yakubovskaya Y. Review of three-dimensional reconstruction methods. The article considers and analyzes methods of three-dimensional reconstruction of objects based on their images. Their classification by type and quantity of incoming information is given. The classified methods are described, their distinctive features are given, advantages and disadvantages are highlighted, and their modern implementations are considered. A comparative analysis of the described methods is given and the optimal method is chosen to create a software solution for the reconstruction of objects in conditions of incompleteness of the initial data.

Keywords: photogrammetry, 3D model, image, 3D reconstruction, reconstruction method.

Статья поступила в редакцию 25.02.2023
Рекомендована к публикации профессором Карабчевским В.В.

УДК 004.92

Высокопроизводительный рендеринг трехмерных изображений для систем объемной пространственной визуализации

С.А. Зори

профессор кафедры «Программная инженерия» им. Л.П. Фельдмана,
Донецкий национальный технический университет

ik.ivt.rec@mail.ru

Аннотация

В статье рассмотрена предложенная автором общая концепция представления трехмерной дискретной объемной сцены и обобщенный метод синтеза объемного изображения воксельной сцены для систем объемной пространственной визуализации. Предложена архитектура вычислительной системы объемной пространственной визуализации на основе разработанных средств. Намечены направления дальнейших исследований методов и алгоритмов эффективной генерации воксельных представлений, а также разработок новых решений на основе ускоряющих техник и упрощений.

Введение

Организация бурно развивающихся в последнее время компьютерных систем объемной пространственной 3D-визуализации предусматривает принципиально новую организацию вычислительного процесса, отличную от «классического» 3D графического конвейера, и применения принципиально другого представления трехмерной дискретной объемной сцены, сложных методов синтеза и визуализации (объемная трассировка лучей, объемный рендеринг и пр.). Среди этих систем наибольшее распространение получили системы воксельного представления и визуализации [1 - 5]. Недостаточный уровень практической реализации таких систем – их сложность, дороговизна, отсутствие стандартизированной аппаратной поддержки, слабая реалистичность создаваемых образов и большое время синтеза (за счет применения в них усложненных трудоемких вычислительных процедур модельной 3D-генерации и 3D-пространственной обработки) обуславливает необходимость некоей «стандартизации» общей концепции представления трехмерной дискретной объемной сцены и обобщенного метода синтеза объемного изображения воксельной сцены, с целью разработки «под них» эффективных методов и алгоритмов синтеза изображений объемных графических примитивов [1 - 5].

В работе рассмотрены предложенная автором общая концепция представления трехмерной дискретной объемной сцены и обобщенный метод синтеза объемного изображения воксельной сцены для систем объемной пространственной визуализации (3D-объемной СОИ). Предложена архитектура вычислительной системы объемной пространственной визуализации на основе раз-

работанных средств, освещены дальнейшие направления исследований и разработок.

Общая концепция представления трехмерной дискретной объемной сцены и синтеза изображений в системе объемной пространственной визуализации

Предлагается следующая обобщенная концепция представления дискретной объемной трехмерной сцены (объемного объектного пространства, «мира») и синтеза 3D-объемных изображений [1] (рис. 1). Далее будем использовать прямоугольную (декартову) правостороннюю трехмерную систему координат.

Пусть заданы:

1) **Воксель** – минимальный трехмерный пространственный элемент дискретизированного с некоторым шагом некоторого трехмерного пространства, в общем случае трехмерный прямоугольный параллелепипед. В простейшем случае – трехмерный куб с единичными сторонами.

2) **Вокселизованный дискретный мир** (сцена) Wd как объемное дискретно-пространственное представление геометрии (Gd) и энергетических характеристик (Ed) «реального» мира (сцены) Wr , заданное объемными воксельными моделями (аппроксимациями) объектов и окружающего их 3D-пространства мира $Wr \xrightarrow{Gd, Ed} Wd$. Для Wd заданы:

2.1) **Параметры пространственной дискретизации мира** (пространства сцены и объектов) – разрешение Ds, Do .

2.2) **Мировая система координат (МСК) - OXYZ**.

В МСК определено воксельное представление пространства сцены S как массив S вокселей $S_{l,m,n}$, где l, m, n – индексы вокселя в S , соответствующие координатам XYZ в МСК.

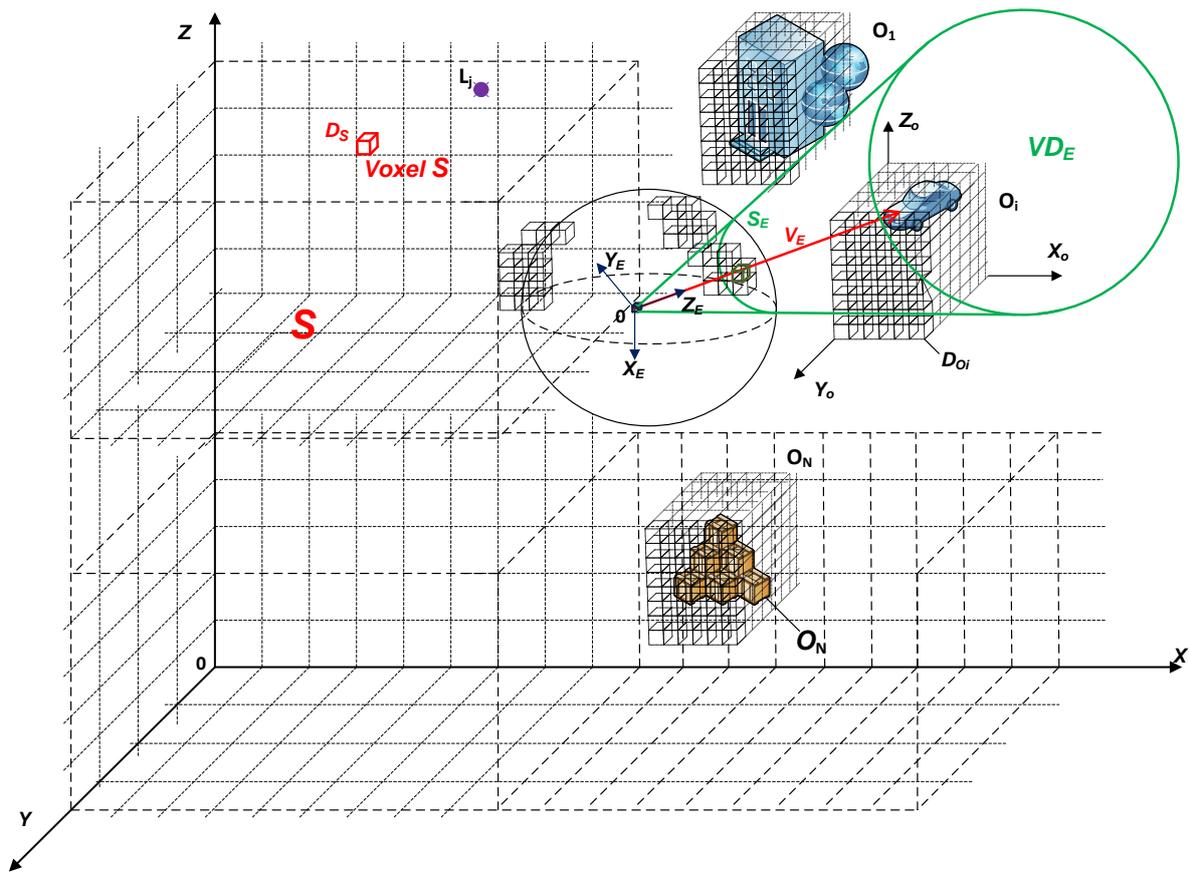


Рисунок 1 - Концепция представления дискретной трехмерной объемной сцены и синтеза изображений в 3D-объемной СОИ

Значение вокселя $S_{l,m,n}$ задает некоторую характеристику «воздуха» пространства сцены в определенном данным вокселем объеме (например, коэффициент поглощения, и т.п.)

2.3) **Воксельные представления объектов сцены O_i** , определенные в системе координат объекта (СКО) $Ox_i Oy_i Oz_i$ или в МСК.

Под воксельным представлением объекта O_i будем понимать множество (массив) Φ_i вокселей $O_{l,m,n}$, таких, что $O_{l,m,n} \in O_i$, l,m,n – индексы вокселя в Φ_i , соответствующие координатам $X_{oi} Y_{oi} Z_{oi}$ в СКО.

Для объекта O_i собственная (связанная) система координат (СКО_i), задается:

- положением СКО_i относительно МСК $(X_{0oi}, Y_{0oi}, Z_{0oi})$;

- углами поворота СКО_i относительно МСК $(\psi_{oi}, \theta_{oi}, \gamma_{oi})$.

При этом значение вокселя $O_{l,m,n}$ суть некоторая характеристика объекта в этом «элементарном» объеме (цвет, прозрачность и т.п.)

2.4) **Источники освещения L_j** – заданы в МСК своим местоположением (X_{Lj}, Y_{Lj}, Z_{Lj}) и энергетическими характеристиками (цвет, интенсивность и пр.), точечные источники света.

Отметим, что источники света могут быть и не точечными, тогда для них задается геометрическое описание поверхности источника и энергетические характеристики (аналогично O_i).

3) **Наблюдатель - E:**

3.1) **Система координат наблюдателя (СКН) - $Ox_E Oy_E Oz_E$.**

СКН относительно МСК задается аналогично СК объектов – своим положением (X_{0E}, Y_{0E}, Z_{0E}) и ориентацией - углами поворота системы координат наблюдателя относительно мировой системы координат $(\psi_E, \theta_E, \gamma_E)$.

3.2) **Направление взгляда наблюдателя (луч визирования) – V_E .** Обычно направление визирования совпадает с осью Z_E .

3.3) **Сектор пространственного обзора наблюдателя - S_E .**

Сектор обзора может задаваться различными способами – пространственными угловыми характеристиками, ограничивающими лучами и пр. в зависимости от используемой модели наблюдателя и формы дисплея.

4) **Объемный экран (дисплей)**

Под **объемным экраном (дисплеем) VD_E** будем понимать видимую наблюдателем E часть

объемного пространства мира - VS_E (дисплей отображает видимое пространство мира).

Видимый объем VS_E (видовое пространство) – видимая наблюдателем E пространственная геометрическая часть объемного пространства S мира (сцены), определяемая физической моделью наблюдения, $VS_E \in S$.

Видимый объем VS_E задается как массив V вокселей $V_{l,m,n}$, полностью покрывающий (заполняющий) VS_E .

В зависимости от физической модели наблюдения (с использованием специфики зрения человека - модели глаза, или без ее использования), а также других принятых упрощений и ограничений, в [1, 4] определены примеры разновидностей форм видового пространственного объема объемных дисплеев.

Объемное изображение – видимая наблюдателю объемная часть вокселизованного дискретного мира, попавшей в видовое пространство объемного дисплея.

5) **Динамические характеристики мира:**

5.1) Параметры, характеризующие поведение объектов во времени:

- изменения в пространственно-дискретизированном представлении геометрии каждого объекта сцены, собственно изменение \mathcal{O}_i ;

- изменения визуальных характеристик объектов (изменения значений \mathcal{O}_i);

- изменения расположения и ориентации объектов (перемещение, вращение объектов и пр.) в МСК.

5.2) Параметры динамики источников освещения:

- пространственного расположения источников освещения;

- изменения в энергетических характеристиках источников освещения.

5.3) Параметры динамики наблюдателя:

- пространственного расположения и ориентации наблюдателя;

- изменения направления визирования и сектора обзора.

5.4) Параметры динамики изменения визуальных свойств окружающего внутреннего пространства сцены – изменения значений $S_{l,m,n}$.

Задачей объемной визуализации является построение объемного пространственно-дискретизированного образа мира – объемного изображения \mathbb{P} , видимого наблюдателю с его позиции в направлении визирования в пространстве объемного дисплея.

То есть, для каждого момента времени найти такое **множество вокселей** $\mathbb{P}_{l,m,n}$, что (рис. 2):

$$\mathbb{P}_{l,m,n} \in \mathbb{P}, \mathbb{P} = V \cap S \cap_{i=1}^N \mathcal{O}_i. \quad (1)$$

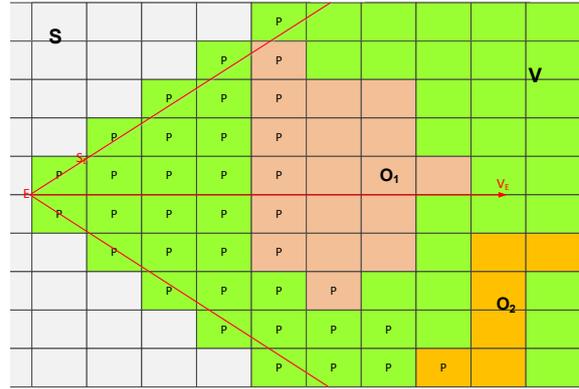


Рисунок 2 - Объемное изображение

При этом каждый воксель объемного изображения $\mathbb{P}_{l,m,n}$ должен получить **значение (характеристику)** $C_{\mathbb{P}_{l,m,n}}$, с учетом характеристик вокселей объектов $C_{\mathcal{O}_{l,m,n}}$, попавших в видовой объем VS_E объемного дисплея VD_E , источников освещения C_{L_j} и характеристик вокселей пространства сцены:

$$C_{S_{l,m,n}}: C_{\mathbb{P}_{l,m,n}} = C_{\mathcal{O}_{l,m,n}} \cup C_{L_j} \cup C_{S_{l,m,n}}. \quad (2)$$

Здесь необходимо отметить **главную особенность такого подхода** к объемной пространственной визуализации – фактически, при наличии готовых пространственно-дискретизированных (воксельных) моделей объектов мира (и других характеристик дискретного мира) построение изображения можно свести к определению видимых наблюдателю вокселей моделей объектов, попавших в видовой пространственной объем объемного дисплея, с дальнейшим учетом в их характеристиках параметров освещения и других визуальных свойств внутреннего окружающего пространства мира.

Таким образом, в рамках предложенной концепции, при **решении статической задачи синтеза объемного изображения** вокселизованного мира с заданными моделями объектов в МСК по сути достаточно:

1. Выполнить построение вокселизированной модели мира – воксельных представлений (моделей) окружающего пространства и объектов, их пространственного размещения в сцене.

Данный этап - сценарный процессинг - для статической задачи синтеза объемного изображения (если геометрия объектов и мира не изменяется) выполняется один раз и может быть вынесен за рамки основного процесса (препроцессирование).

2. Выполнять процедуру трассировки лучей визирования с заданным шагом пространственной дискретизации по видимому наблюдателю объему пространства сцены до первого пересечения с вокселем некоторого объекта, который и

будет непосредственно виден наблюдателю (лучу) в данной точке.

3. Для применения к этому вокселю характеристик освещения достаточно выполнить от него трассировку светового (теневого) луча до источника(-ов) света и модифицировать исходную энергетическую характеристику материала вокселя (цвет и др.) с учетом характеристик источника света и окружающей среды (спец. эффекты и пр.).

4. Для найденного воксельного элемента объекта можно также вычислять и более слож-

ные характеристики освещения - фотореалистичные эффекты преломлений и отражений, используя аналогичные, применяемые в традиционной трехмерной графике, техники методов трассировки лучей [1, 3, 4, 5, 6].

Обобщенный метод синтеза объемного изображения дискретного воксельного мира проиллюстрирован на рис. 3.

В псевдокоде приведено упрощенное определение отражений и преломлений (единичные эффекты, аналогично базовому алгоритму Уиттеда трассировки лучей [7]).

Метод синтеза объемного изображения воксельной сцены

```

Begin
Сформировать  $S = \{S_{l,m,n}; C_{S_{l,m,n}}\}$ 
Do (для всех объектов  $O_i$  объемного мира  $S$ ):
Сформировать  $\mathbb{O}_i = \{O_{l,m,n}; C_{O_{l,m,n}}\}$ ;
Преобразовать  $O_i$  в МСК И Разместить в  $S$ :
 $S \cup \mathbb{O}_i$ .
End_Do
Do (для всех вокселей  $V$  объемного экрана  $VD_E$ ):
 $V_{l,m,n} = S_{l,m,n}$ ;
 $C_{V_{l,m,n}} = C_{S_{l,m,n}}$ ;
End_Do
Do (для каждого луча визирования видового пространства  $VS_E$  (первичного луча) из  $E$  с заданными  $D_S$  и  $S_E$ ):
Выполнить пошаговую ( $k$ ) дискретную трассировку луча через воксели  $VS_E$  и найти первое пересечение луча с вокселем некоторого объекта  $\mathbb{O}_i$  мира;
 $\mathfrak{P}_{l,m,n}(k) = V_{l,m,n}(k)$ 
 $C_{\mathfrak{P}_{l,m,n}}(k) = C_{V_{l,m,n}}(k)$ ;
If (есть пересечение)
Then
 $\mathfrak{P}_{l,m,n}(k) = O_{l,m,n}(k)$ 
Do (для каждого источника света  $L_j$ ):
Выполнить дискретную трассировку теневого луча из вокселя пересечения первичного луча с объектом к источнику света  $L_j$ ;
If (есть пересечение)
If (пересечение с некоторым объектом  $\mathbb{O}_i'$  мира)
Then
прекратить обработку этого  $L_j$  (он не виден), воксель - в тени;
Else
учесть освещенность от этого источника  $L_j$ :
 $C_{\mathfrak{P}_{l,m,n}}^j = C_{O_{l,m,n}}(k) \cup C_{L_j}$ ;

```

```

End_If
End_If
 $C_{\mathfrak{P}_{l,m,n}}(k) = C_{V_{l,m,n}}(k) + C_{\mathfrak{P}_{l,m,n}}^j$ ;
End_Do
If (материал вокселя объекта  $\mathbb{O}_i$  отражающий)
Then
Выполнить дискретную трассировку отраженного луча и найти первое пересечение луча с вокселем некоторого объекта  $\mathbb{O}_i''$  мира;
If (есть пересечение с некоторым объектом  $\mathbb{O}_i''$ )
Then
учесть характеристику вокселя объекта  $\mathbb{O}_i''$  с коэффициентом отражения  $R_i$  материала  $\mathbb{O}_i$ :
 $C_{\mathfrak{P}_{l,m,n}}(k) = C_{\mathfrak{P}_{l,m,n}}(k) \cup (C_{\sigma''_{l,m,n}} * R_i)$ ;
End_If
End_If
If (материал вокселя объекта  $\mathbb{O}_i$  преломляющий)
Then
Выполнить дискретную трассировку преломленного луча и найти первое пересечение луча с вокселем некоторого объекта  $\mathbb{O}_i'''$  мира;
If (есть пересечение с некоторым объектом  $\mathbb{O}_i'''$ )
Then
учесть значение вокселя объекта  $\mathbb{O}_i'''$  с коэффициентом преломления  $F_i$  материала  $\mathbb{O}_i$ :
 $C_{\mathfrak{P}_{l,m,n}}(k) = C_{\mathfrak{P}_{l,m,n}}(k) \cup (C_{\sigma'''_{l,m,n}} * F_i)$ ;
End_If
End_If
End_Do
End

```

Рисунок 3 - Обобщенный метод решения статической задачи синтеза объемного изображения воксельной дискретной сцены в объемной 3D- СОИ

Алгоритм дискретной трассировки (траверса) луча при этом может быть наглядно представлен следующим образом (рис. 4) [1, 4]. Здесь показана проекция воксельной сетки в плоскости XY. Для корректного траверса воксельной сетки луч должен последовательно посетить воксели 1, 2, 3, 4, 5, 6, 7, 8. Уравнение луча при этом - $u \vec{v} + t \vec{v}$ для $t \geq 0$ и алгоритм распространяет луч в интервале t , каждый t покрывает 1 воксель.

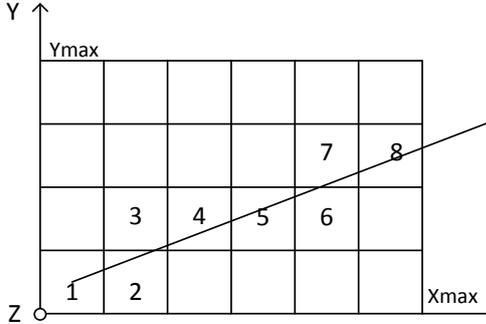


Рисунок 4 - Траверс трассирующего луча

Предлагается следующий инкрементный алгоритм траверса дискретного луча (рис. 5).

```

Begin
List:= NULL;
Определение  $tX, tY, tZ, X, Y, Z$  по  $t(\vec{u})$ ;
Do {
If ( $tX < tY$ ) Then
If ( $tX < tZ$ ) Then
 $X := X + dX$ ;
If ( $X \geq Xmax$ ) Then /* вышли за сетку по X */
return (NULL); End_If
 $tX := tX + tdX$ ;
Else {
 $Z := Z + dZ$ ;
If ( $Z \geq Zmax$ ) Then /* вышли за сетку по Z */
return (NULL); End_If
 $tZ := tZ + tdZ$ ; } End_If
Else {
If ( $tY < tZ$ ) Then
 $Y := Y + dY$ ;
If ( $Y \geq Ymax$ ) Then /* вышли за сетку по Y */
return (NULL); End_If
 $tY := tY + tdY$ ;
Else {
 $Z := Z + dZ$ ;
If ( $Z \geq Zmax$ ) Then /* вышли за сетку по Z */
return (NULL); End_If
 $tZ := tZ + tdZ$ ; } End_If
} End_If
List:= List_of_Object ([X][Y][Z]);
} while (List == NULL);
return (List);
End

```

Рисунок 5 - Алгоритм траверса пространственно- дискретного луча

Алгоритм начинается с инициализации - определения начального вокселя \vec{u} . Если начало луча находится вне сетки, то нужно найти воксель, через который луч проникает в сетку (сцену). Переменные X и Y инициализируются координатами исходного вокселя. Переменные dX и dY – шаги и представляются либо +1, либо -1, указывая на увеличение или уменьшение X и Y на каждом шаге пересечения луча вокселя (определяется знаком X и Y координат в \vec{v}).

Далее определяется значение t , при котором луч пересекает вертикальную границу первого вокселя, и сохраняется в переменной tX . Аналогично производятся вычисления для Y (tY). Минимум этих двух величин будет показывать, как долго мы можем распространять луч в текущем вокселе. Далее вычисляется tdX и tdY , которые показывают, как далеко вдоль луча мы можем двигаться (в единицах t) для горизонтальной составляющей (равной ширине вокселя) или вертикальной (равной высоте вокселя). Аналогичная техника используется для Z - координаты.

Базовый цикл продолжается до тех пор, как мы не найдем воксель с непустым списком объектов ($List$) в нем (видимый воксель объекта в этом трассирующем луче), или не покинем пространство (сетку) сцены.

Совершенно понятно, что таким же образом можно представить и траверсы вторичных лучей.

Таким образом, предложенный подход сочетания пространственно-дискретного представления мира и пространственно- дискретизированной трассировки лучей позволяет достаточно просто алгоритмически выполнить построение объемного изображения, фактически одновременно решая сразу множество задач-стадий графического 3D-конвейера – сценарного процессинга, геометрических модельных преобразований (включая отсечение, видовые и аффинные преобразования), рендеринга (включая решение задачи удаления невидимых поверхностей и освещения), и собственно визуализации.

Обобщенная архитектура синтеза объемного изображения, иллюстрирующая техники синтеза в воксельной 3D- объемной СОИ с использованием предложенных средств, представлена на рис. 6.

Предлагается [1] следующее качественно-адекватное представление значений (характеристик - параметров) вокселей объектов мира $C_{O_{l,m,n}}$, позволяющее производить расчет освещенности вокселей $C_{\Phi_{l,m,n}}$ объемного изображения $\Phi_{l,m,n}$ с учетом характеристик источников освещения C_{L_j} и характеристик вокселей пространства $C_{S_{l,m,n}}$ (2), включающее следующие признаки и значения (рис. 7).

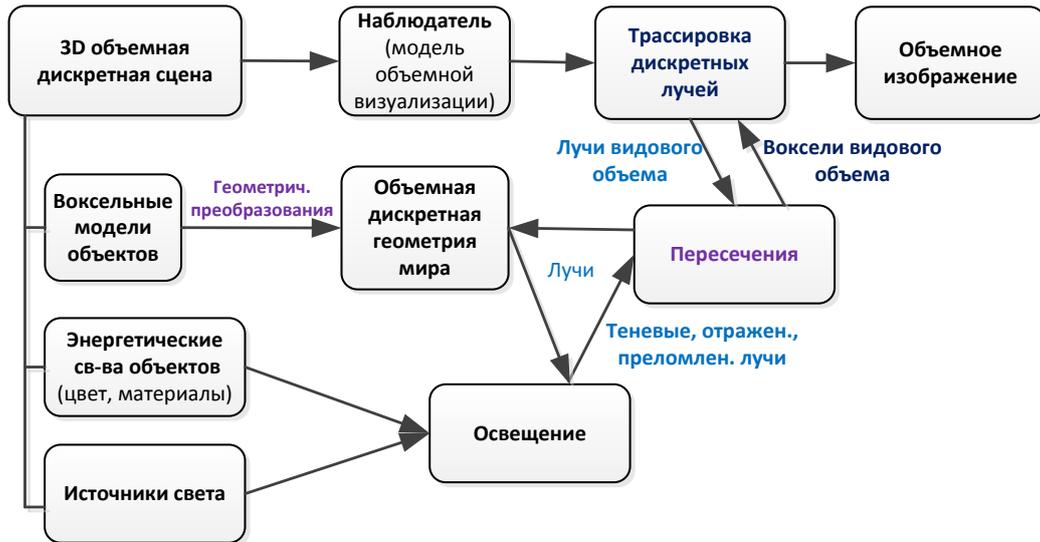


Рисунок 6 - Обобщенная архитектура синтеза 3D-объемного изображения дискретного воксельного мира

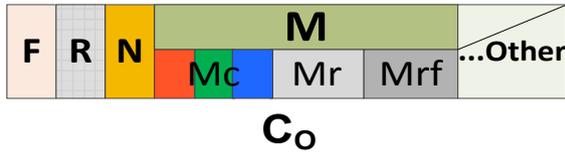


Рисунок 7 - Предлагаемое представление характеристики $C_{O_{l,m,n}}$ вокселей объектов $O_{l,m,n}$

- признак **F** (face) вокселя (воксель внешней образующей поверхности объекта (F=1) / воксель внутреннего объема (F=0));
- доля **R** (gate) единичного объема в этом вокселе (R=1, если F=0 и R = доли объемной площади поверхности объекта в этом вокселе, если F=1);
- нормаль **N** (normal) к поверхности объекта;
- энергетические характеристики материала **M** (material) вокселя – цвет материала **Mc** (например, в RGB), отражающая **Mr** и преломляющая **Mrf** характеристики материала или их упрощения;
- другие необходимые характеристики.

По аналогии и с использованием упрощений можно подойти и к заданию значений вокселей внутреннего пространства сцены $C_{S_{l,m,n}}$ и источников света C_{L_j} . Например, в модели $C_{S_{l,m,n}}$ можно исключить признаки F, R, N, а вместо Mr и Mrf использовать значение прозрачности Mt (transparency) материала. Для задания C_{L_j} достаточно определить цвет источника света Mc и вектор освещения N.

Характеристики вокселей **C** определяются (заполняются) на этапе генерации моделей мира (рис. 3). Учет характеристик вокселей **C** мира при синтезе объемных изображений методами 3D- воксельной трассировки лучей выполняется

в соответствии с основными этапами 3-4 метода решения задачи синтеза объемного изображения вокселизованного дискретного мира (рис. 3). При этом используются аналогичные, применяемые в традиционной трехмерной графике, техники определения освещенности методов трассировки лучей в зависимости от используемого прототипа метода рейтрейсинга [1, 3, 4].

Заметим, что для организации ускорения синтеза динамических объемных изображений мира (динамика объектов, наблюдателя, освещения), можно использовать известные в компьютерной графике подходы – распараллеливание, временную и пространственную декомпозицию процесса обработки [1, 2, 4, 5, 6, 8, 9].

Например, организацию объемного дисплея с двумя буферами кадра для соблюдения заданных временных критериев синтеза объемного изображения, применение конвейерной обработки для уменьшения времени 3D- пространственной визуализации, закрепление вычислительных компонент 3D СОИ за объектами мира (параллельная перестройка (генерация) пространственных воксельных моделей объектов, геометрические аффинные преобразования (вращения, перемещения и пр.) и т.д.).

Архитектура вычислительной системы аппаратной поддержки 3D-объемной визуализации воксельного мира на основе предложенных средств

Предлагаемая архитектура вычислительной системы аппаратной поддержки 3D-объемной визуализации пространственно-дискретизированного воксельного мира с учетом предложенной концепции и обобщенного метода синтеза (рис. 3 - 7) показана на рис. 8 [2].

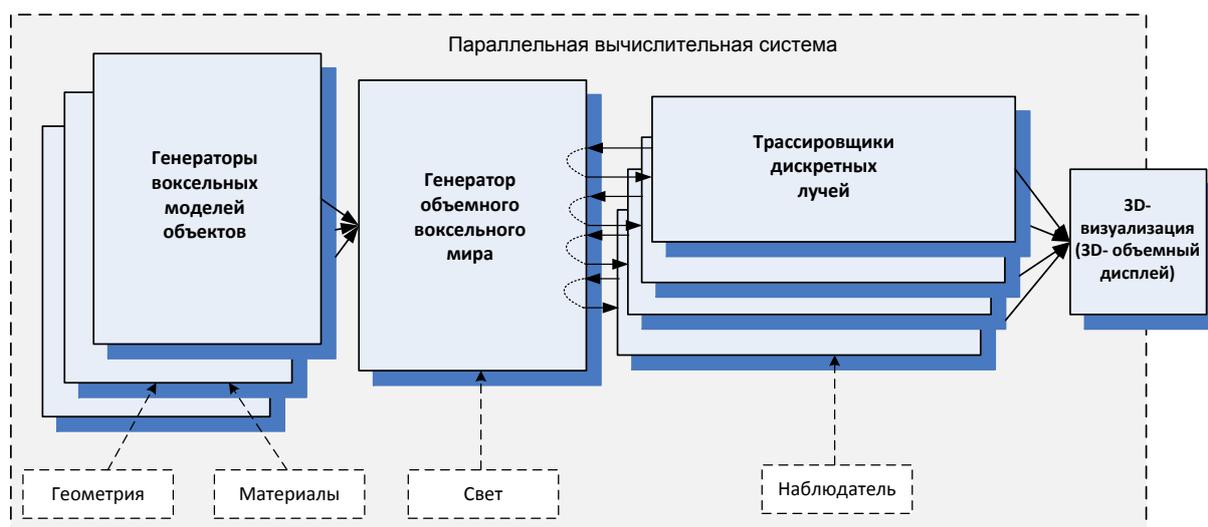


Рисунок 8 - Архитектура вычислительной системы аппаратной поддержки 3D- объемной визуализации воксельного мира)

Основные направления ускорения синтеза в системах такого класса связаны, согласно предложенной концепции генерации 3D- объемных изображений дискретного воксельного мира, с ускорением решения двух основных задач:

- эффективной генерации воксельных представлений (моделей) объектов пространственно-дискретного мира;
- эффективного и быстрого выполнения дискретной трассировки лучей.

Таким образом, предлагаются к дальнейшим исследованиям такие направления ускорения синтеза 3D- изображений в системах 3D-объемной пространственной визуализации:

- ускорение алгоритмической базы синтеза на основе упрощений в модели дискретной пространственной трассировки лучей (одного отраженного/преломленного луча (нет рекурсии)) и быстрого траверса луча;
- ускорение алгоритмической базы синтеза на основе модификации ускоряющей техники трассировки BVH - AABB;
- ускорение алгоритмической базы синтеза на основе разработки быстрых алгоритмов 3D-дискретной (воксельной) аппроксимации пространственных геометрических форм объектов сцены и генерации пространственных 3D- графических примитивов (включая быструю генерацию дискретного 3D- трассировочного луча вокселизованного пространства сцены);

- ускорение 3D- объемного синтеза изображений на основе предложенной алгоритмической базы путем отображения и эффективной реализации параллельной вычислительной системы его аппаратной поддержки (рис. 8) на архитектуру GPU-CUDA для организации синтеза в реальном времени [10].

Выводы

В статье рассмотрена предложенная автором общая концепция представления трехмерной дискретной объемной сцены и обобщенный метод синтеза объемного изображения воксельной сцены для систем объемной пространственной визуализации.

Предложена архитектура вычислительной системы объемной пространственной визуализации на основе разработанных средств. Освещены дальнейшие направления исследований и разработок, главными из которых являются:

- методы и алгоритмы эффективной генерации воксельных представлений (3D моделей) объектов пространственно-дискретного мира (на сегодняшний день нет стандартов и общепринятых концепций);
- алгоритмы эффективного и быстрого выполнения дискретной трассировки лучей на основе ускоряющих техник и упрощений;
- способы отображения предложенной архитектуры вычислительной системы на GPU для организации работы 3D СОИ в реальном времени.

Литература

1. Zori, S. A. Spatial Visualization via Real Time 3D Volumetric Display Technologies / S. A. Zori, E. A. Bashkov, Anas-M. Al Oraiqat // Lambert Academic Publishing (LAP). - 2018-01-25 | book. ISBN: 978-620-2-09398-9
2. Specialized Computer Systems for Environment Visualization // Springer 3D Research. – 2018. - Vol. 9, No 2. - 13 P. | DOI: 10.1007/s13319-018-0168-9
3. Башков, Е. А. Реалистичная пространственная визуализация с использованием технологий объемного отображения: Монография /

Е.А. Башков, С.А. Зори. – Донецк: ГВУЗ "ДонНТУ", 2014. – 150 с.

4. Зори, С. А. Высокопроизводительные средства организации реалистичной объемной визуализации в системах трехмерной компьютерной графики: Монография / Е. А. Башков, С. А. Зори. – Покровск: ГВУЗ "ДонНТУ", 2017. - 215 с.: ил. ISBN 978-966-377-215-8

5. Зори, С. А., Аноприенко, А. Я. Особенности организации объемной визуализации в системах трехмерной компьютерной графики // Программная инженерия: методы и технологии разработки информационно- вычислительных систем (ПИИВС-2018): сборник научных трудов II Международной научно-практической конференции, 14-18 ноября 2018 г. – Донецк, ГОУВПО «Донецкий национальный технический университет», 2018. – Том. 1. - С. 302 – 308.

6. Bashkov, E. A. Realistic stereo visualization system architecture using ray tracing / Evgeniy A. Bashkov; Sergii A. Zori; Anas M. Al-Oraiqat; Akmaral Tleshova; Piotr Kisala // Proc. SPIE 10808 - Photonics Applications in Astronomy, Communi-

cations, Industry and High-Energy Physics Experiments, 2018. - Wilga, Poland. - Vol. 108080X (1 October 2018). – 8 p. DOI: 10.1117/12.2501601

7. Whitted Turner. An improved illumination model for shaded display / Whitted T. // Commun. ACM, 1989. – Vol. 23, No. 6. – PP. 343 – 349.

8. Зори, С. А. Использование средств аппаратной поддержки для повышения производительности систем 3D- пространственной визуализации / С. А. Зори, А. Я. Аноприенко, Р. В. Мальчева, О. А. Авксентьева // Информатика и кибернетика. – Донецк: ДонНТУ, 2019. – № 1 (15). - С. 5 – 12.

9. Malcheva, R. An Acceleration of FPGA-based Ray Tracer / R. Malcheva, M. Yunis // European Scientific Journal, 2014. - Vol.10, N7. – PP. 186-190.

10. 3D Primitives GPGPU Generation for Volume Visualization in 3D Graphics Systems Anas M. Al-Oraiqat, Sergii A. Zori //Journal of KING ABDULAZIZ UNIVERSITY: Computing and Information Technology Sciences, 2020. - Vol. 9, No.2. – PP. 85-97. (ISSN 1658-6336)

Зори С.А. Высокопроизводительный рендеринг трехмерных изображений для систем объемной пространственной визуализации. В статье рассмотрена предложенная автором общая концепция представления трехмерной дискретной объемной сцены и обобщенный метод синтеза объемного изображения воксельной сцены для систем объемной пространственной визуализации. Предложена архитектура вычислительной системы объемной пространственной визуализации на основе разработанных средств. Намечены направления дальнейших исследований методов и алгоритмов эффективной генерации воксельных представлений, а также разработок новых решений на основе ускоряющих техник и упрощений.

Ключевые слова: объемная пространственная визуализация, синтез 3D изображений, воксель, 3D объемная сцена, рендеринг, повышение производительности, архитектура

Zori S.A. High-performance rendering of 3D images for volumetric spatial visualization systems. The article discusses the general concept for representing a 3D discrete volumetric scene and a generalized method for synthesizing a volumetric image of a voxel scene for volumetric spatial visualization systems. The architecture of a computer system for volumetric spatial visualization based on the developed tools is proposed. The directions of further research of methods and algorithms for efficient generation of voxel representations, as well as the development of new solutions based on accelerating techniques and simplifications are offered.

Keywords: volumetric spatial visualization, 3D image synthesis, voxel, volumetric 3D scene, rendering, performance improvement, architecture

Статья поступила в редакцию 02.03.2023
Рекомендована к публикации профессором Мальчевой Р.В.

Анализ эффективности модифицированного алгоритма измерения физических величин для сенсорных систем

А. А. Койбаш

Донецкий национальный технический университет
mr.koibash@gmail.com

Аннотация

Рассмотрены детали реализации модифицированного алгоритма измерения физических величин для сенсорных систем, в основе которого лежит постбинарное кодирование информации. С целью проверки работоспособности алгоритма выполнен анализ его эффективности с использованием реального прототипа сенсорной системы. На основании полученных в опытах данных сформированы интервалы и выполнено их постбинарное кодирование с использованием различных форматов. Определены преимущества и недостатки использования каждого формата. На основании результатов экспериментального анализа предложен гибридный вариант постбинарного кодирования информации.

Введение

На сегодняшний день наблюдается увеличение степени интеграции информационно-компьютерной инфраструктуры во многие сферы человеческого общества. Активно развивается область цифровых услуг и удобств, а промышленное производство характеризуется всё большей степенью автоматизации рутинной работы. Эти процессы невозможно представить без повсеместного увеличения количества используемых датчиков, поскольку именно они и являются «глазами и ушами» для компьютерной техники.

Такую совокупность датчиков можно очертить в рамках единой сенсорной составляющей информационно-компьютерной инфраструктуры. Улучшение её характеристик положительным образом скажется на работе большинства цифровых систем.

Для достижения этой цели в статье [1] рассмотрены различные подходы к модернизации сенсорных устройств. Одним из вариантов является взаимная интеграция большого количества сенсоров в единое периферийное устройство (суперсенсор) [2]. Спектр их применения широк: от сбора статистических параметров окружающей среды до возможности предоставления полной информации обо всех муниципальных аспектах умных городов с автоматическим принятием системой соответствующих решений [3].

В условиях стремительного развития компьютерных систем в мировом масштабе [4] необходимо учитывать качественные характеристики отдельных элементов сенсорной составляющей такой инфраструктуры. В частности, на основе аналитических зависимостей, полученных в статье [5] по

нескольким различным видам датчиков, обнаружены тенденции к снижению размеров, веса, тока питания, а также улучшению шумовых характеристик. Данные параметры позволяют дать благоприятный прогноз для существенного повышения степени взаимной интеграции разнородных датчиков в единую систему.

В рамках поисков путей модернизации сенсорной составляющей в статье [6] разработан модифицированный алгоритм измерения физических величин для сенсорных систем. Блок-схема алгоритма с небольшими доработками представлена на рис. 1. В его основе находятся 2 процедуры:

– формирование и расширение интервала относительно среднего арифметического значения;

– постбинарное кодирование интервала.

Таким образом, в рамках первого этапа происходит рационализация способа хранения информации. Значение, получаемое с датчика, как и полезная информация в виде диапазона погрешности сохраняются в виде единого интервала.

Второй этап связан с использованием современных исследований в области дальнейшей эволюции вычислений. В книге [7] авторами определены закономерности развития компьютеринга и рассмотрен переход к постбинарному представлению информации. Основой такого подхода является логика четырёх состояний – тетралогика, в которой помимо булевых 0 (ложь) и 1 (истина) появились ещё два состояния: А (неопределённость: «ни истина, ни ложь») и М (множественность: «и истина, и ложь»). Эта концепция в определённой степени соответствует квазиквантовому и квазигенетическому компьютерингу.

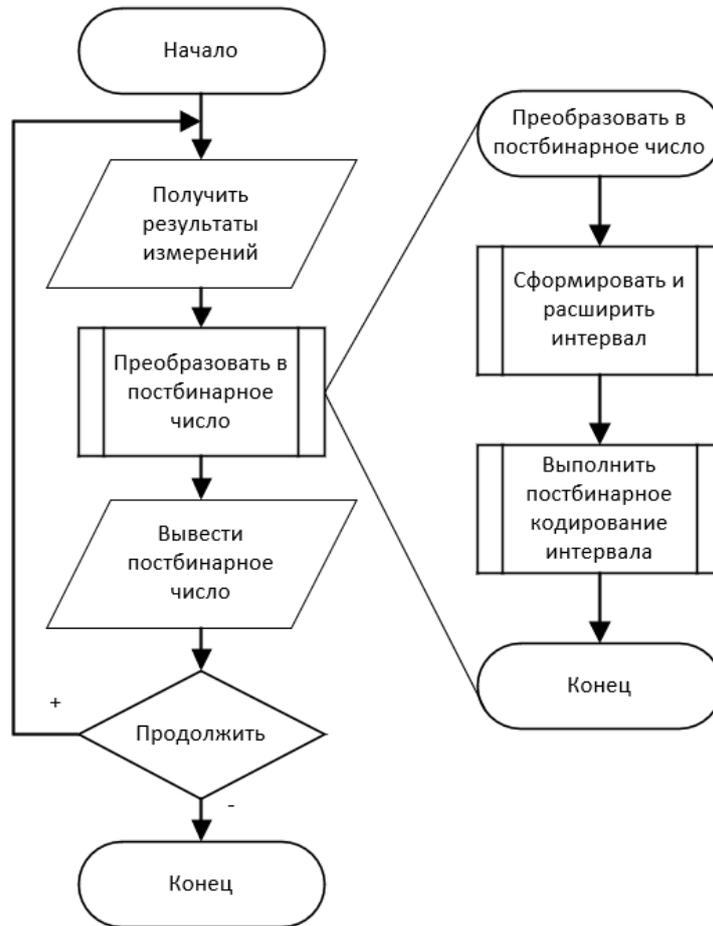


Рисунок 1 – Блок-схема модифицированного алгоритма измерения физических величин для сенсорной системы [6]

Несмотря на то, что развитие квантовых и ДНК-компьютеров находится лишь на начальной стадии, постбинарный компьютеринг уже сейчас позволяет реализовать некоторые особенности таких вычислений [8-9].

В рамках предложенного алгоритма постбинарное кодирование предоставляет множество инструментов эффективной работы с информацией. С целью моделирования работы алгоритма в реальных задачах проведён эксперимент на основе прототипа сенсорной системы.

Проведение эксперимента

Для анализа эффективности алгоритма необходимы реальные значения температуры. Следовательно, нужно расширить набор полученных с сенсора данных, продолжив начатые в статье [6] серии измерений. Схема эксперимента остаётся прежней: за короткий промежуток времени микроконтроллер на плате STM32F407G Discovery получает ряд показаний с подключенного к плате терморезистора. Сам датчик погружён в большой сосуд с теплоёмкой жидкостью, что позволяет минимизировать погрешности при нагревании или остывании до

температуры окружающей среды. Полученный сигнал дискретизируется посредством встроенного 12-разрядного аналогово-цифрового преобразователя (в значениях от 0 до 4095). Одновременно с использованием сконструированной сенсорной системы в жидкость погружён аквариумный термометр с заводской калибровкой, что позволяет достоверно определить реальную температуру жидкости.

Количество измерений за малый промежуток времени было выбрано равным 16. При выборе числа [6] учитывалась не только вероятность попадания результата в доверительный интервал, но и меньшие в сравнении с компьютерами аппаратные ресурсы микроконтроллеров, которые в свою очередь накладывают определённые ограничения на вычисления. Таким образом, при 16 измерениях результат будет расположен:

– с вероятностью 99.993% в доверительном интервале $[-\sigma_x; +\sigma_x]$;

– с вероятностью 95% в доверительном интервале $[-0.5\sigma_x; +0.5\sigma_x]$.

Предыдущие опыты уже подтверждали, что такое количество измерений не будет оказывать существенного влияния на

производительность всей сенсорной системы. В дополнение к проведённым экспериментам сделаны 7 серий по 16 измерений при различных температурах в диапазоне от $\approx 18\text{ }^{\circ}\text{C}$ до $\approx 49\text{ }^{\circ}\text{C}$. Для каждой серии измерений посчитано среднее арифметическое значение аналогово-цифрового преобразователя. В дополнение ко всему вышеперечисленному представлена температура, полученная с аквариумного электронного

термометра. Результаты измерений представлены в таблице 1.

Из таблицы видно, что присутствуют колебания получаемых с АЦП значений. Тем не менее, среднее арифметическое измерений должно представлять собой достаточно точный результат. На следующем шаге необходимо получить значения в градусах Цельсия ($^{\circ}\text{C}$).

Таблица 1 – Результаты серий измерений датчика температуры

№	1	2	3	4	5	6	7	8	9
АЦП	1415	1426	1608	1727	1868	2043	2166	2192	2575
	1382	1412	1635	1723	1877	2053	2147	2196	2559
	1391	1421	1610	1753	1868	2039	2156	2189	2596
	1409	1424	1628	1746	1876	2028	2145	2206	2581
	1415	1421	1629	1723	1875	2042	2159	2196	2581
	1399	1425	1613	1723	1886	2050	2164	2206	2588
	1404	1422	1630	1720	1869	2030	2156	2196	2596
	1390	1420	1629	1729	1862	2052	2152	2205	2576
	1406	1428	1615	1728	1864	2042	2162	2180	2598
	1415	1430	1636	1725	1866	2054	2173	2187	2590
	1395	1423	1611	1721	1877	2035	2158	2198	2575
	1401	1428	1625	1725	1873	2033	2165	2185	2571
	1409	1416	1628	1730	1865	2047	2149	2199	2579
	1397	1425	1633	1748	1869	2051	2151	2196	2588
	1403	1419	1618	1731	1870	2043	2170	2201	2595
1411	1427	1625	1723	1873	2051	2155	2193	2596	
АЦП _{ср}	1402.6	1422.9	1623.3	1729.7	1871.1	2043.3	2158	2195.3	2584
t _{акв} , $^{\circ}\text{C}$	17.8	18.6	23.3	26.1	29.7	33.3	37.5	38.9	49.0

При анализе алгоритма прецизионность датчика температуры (повторяемость, или, другими словами, соответствие показаний датчика реальной температуре) не является целевой задачей, поскольку работа алгоритма нацелена именно на точность (математическое ожидание). Поэтому в данном эксперименте прецизионностью можно пренебречь и выполнить «грубую» калибровку датчика температуры.

График на рис. 2 оси абсцисс соответствуют серии значений среднего арифметического числа, полученного после дискретизации сигнала с терморезистора. Ось ординат соответствует температуре в градусах Цельсия ($^{\circ}\text{C}$), полученной с аквариумного термометра. По результатам сформирована

линия тренда и получена линейная зависимость температуры от полученного с АЦП дискретного значения.

Вследствие этого получена формула:

$$y = 0,0262x - 19,95. \quad (1)$$

Данные из таблицы 1 пересчитаны с использованием формулы (1): определены минимальные и максимальные значения, полученные с АЦП (АЦП_{min} и АЦП_{max}), а также вычислено среднее арифметическое (АЦП_μ).

С использованием формулы эти числа преобразованы в градусы Цельсия ($^{\circ}\text{C}$). Исходные данные и результаты расчётов представлены в таблице 2

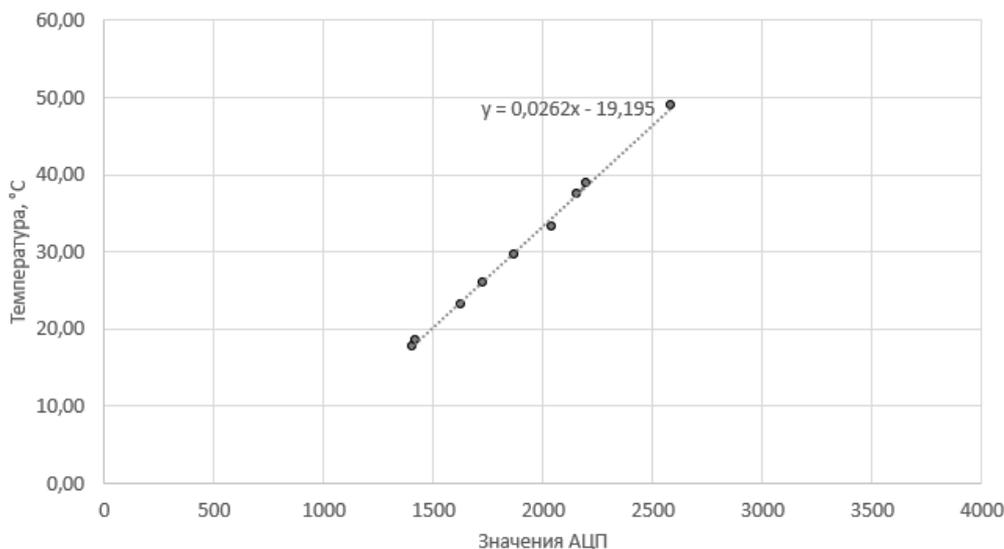


Рисунок 2 – Градуировка терморезистора

Таблица 2 – Данные пересчёта дискретных значений температуры в градусы Цельсия ($^\circ\text{C}$)

№	АЦП _{min}	АЦП _{max}	$\overline{\text{АЦП}}$	$t_{\min}, ^\circ\text{C}$	$t_{\max}, ^\circ\text{C}$	$\bar{t}, ^\circ\text{C}$
1	1382	1415	1402,6	17,0134	17,8780	17,55312
2	1412	1430	1422,9	17,7994	18,2710	18,08498
3	1608	1636	1623,3	22,9346	23,6682	23,33546
4	1720	1753	1729,7	25,8690	26,7336	26,12314
5	1862	1886	1871,1	29,5894	30,2182	29,82782
6	2028	2054	2043,3	33,9386	34,6198	34,33946
7	2145	2173	2158	37,0040	37,7376	37,34460
8	2180	2206	2195,3	37,9210	38,6022	38,32186
9	2559	2598	2584	47,8508	48,8726	48,50580

Из таблицы 2 можно сделать вывод, что погрешность в данном случае достаточно велика и разница между минимальным и максимальным значениями достигает величины вплоть до одного градуса. Тем не менее, благодаря заранее определённому количеству измерений результат в виде среднего арифметического с высокой вероятностью имеет необходимую точность.

Выравнивание интервала

Поскольку входные данные для первой процедуры алгоритма готовы, в рамках эксперимента каждый интервал прошёл процедуру выравнивания. Согласно статье [6] данный этап выполняется в три шага:

– вычисление расстояний до границ интервала (формулы 1, 2):

$$t_{d_1} = |\bar{t} - t_{\min}| \quad (1)$$

$$t_{d_2} = |\bar{t} - t_{\max}| \quad (2)$$

– поиск максимального расстояния из предыдущего шага (формула 3):

$$t_{d_{\max}} = \max \{t_{d_1}, t_{d_2}\}. \quad (3)$$

– расширение границ интервала (формула 4):

$$t \in [\bar{t} - t_{d_{\max}}, \bar{t} + t_{d_{\max}}]. \quad (4)$$

Эти шаги и формирование нового интервала для всех серий измерений отражены в таблице 3. Итоговый расширенный интервал отражён в столбцах t_{\min} и t_{\max} ($t \in [t_{\min}; t_{\max}]$).

Таблица 3 – Данные процесса выравнивания интервала

№	$t_{\min}, ^\circ\text{C}$	$t_{\max}, ^\circ\text{C}$	$\bar{t}, ^\circ\text{C}$	$t_{d1}, ^\circ\text{C}$	$t_{d2}, ^\circ\text{C}$	$t_{d\max}, ^\circ\text{C}$	$t_{n\min}, ^\circ\text{C}$	$t_{n\max}, ^\circ\text{C}$
1	17.0134	17.8780	17.5531	0.5397	0.3249	0.5397	17.0134	18.0928
2	17.7994	18.2710	18.0850	0.2856	0.1860	0.2856	17.7994	18.3706
3	22.9346	23.6682	23.3355	0.4009	0.3327	0.4009	22.9346	23.7364
4	25.8690	26.7336	26.1231	0.2541	0.6105	0.6105	25.5126	26.7336
5	29.5894	30.2182	29.8278	0.2384	0.3904	0.3904	29.4374	30.2182
6	33.9386	34.6198	34.3395	0.4009	0.2803	0.4809	33.9386	34.7404
7	37.0040	37.7376	37.3446	0.3406	0.3930	0.3930	36.9516	37.7376
8	37.9210	38.6022	38.3219	0.4009	0.2803	0.4009	37.921	38.7228
9	47.8508	48.8726	48.5058	0.6550	0.3668	0.6550	47.8508	49.1608

Постбинарное кодирование информации

Вторая процедура алгоритма предполагает постбинарное кодирование интервала. Для интервала, границы которого представлены вещественными числами в двоичном формате float, наибольшим образом подходят следующие два формата [10]:

- PostBinary 64/32 p, позволяющий хранить одно вещественное число одинарной точности в виде тетракода;
- PostBinary 128/32 ip, предназначенный для хранения двух вещественных чисел одинарной точности в виде тетракода.

В статье [6] дан обзор первого формата и разобран пример кодирования/декодирования на одном из диапазонов (без выравнивания интервала). Данный формат характеризуется меньшими затратами по памяти, чем второй, а также хорошо показывает себя на интервалах, между границами которых разница не слишком велика.

Таблица 4 – Результаты кодирования и декодирования интервала (формат PostBinary 64/32 p)

№	$[t_{n\min}; t_{n\max}], ^\circ\text{C}$	$\bar{t}, ^\circ\text{C}$	$t_{\text{postbinary64/32p}}$	$[t_{\text{decmin}}; t_{\text{decmax}}], ^\circ\text{C}$	$\bar{t}_{\text{dec}}, ^\circ\text{C}$
1	[17.0134; 18.0928]	17,5531	0 10000011 00ММАААААААААААААААА М0	[17; 19]	18
2	[17.7994; 18.3706]	18,0850	0 10000011 00МММААААААААААААААА М0	[16.5; 19.5]	18
3	[22.9346; 23.7364]	23,3355	0 10000011 011МММАААААААААААААА М0	[22.25; 23.75]	23
4	[25.5126; 26.7336]	26,1231	0 10000011 10ММААААААААААААААААА М0	[25; 27]	26
5	[29.4374; 30.2182]	29,8278	0 10000011 11ММААААААААААААААААА М0	[29; 31]	30
6	[33.9386; 34.7404]	34,3395	0 10000100 000ММААААААААААААААААА М0	[33; 35]	34
7	[36.9516; 37.7376]	37,3446	0 10000100 0010МММААААААААААААААА М0	[36.25; 37.75]	37
8	[37.9210; 38.7228]	38,3219	0 10000100 001ММААААААААААААААААА М0	[37; 39]	38
9	[47.8508; 49.1608]	48,5058	0 10000100 ММММААААААААААААААААААА М0	[34; 62]	48

Тем не менее, в [10] автором указано, что при конвертировании чисел в постбинарный формат не должен быть затронут порядок вещественного числа, что может иметь место в реальных ситуациях.

Для того, чтобы постбинарное число полностью охватывало интервал, при этом не слишком его расширило, введено дополнительное условие для битов, переходящих в процессе кодирования в серию множественности (M):

- меньшее число интервала должно содержать по крайней мере один бит со значением 1 (истина);
- большее число интервала должно содержать по крайней мере один бит со значением 0 (ложь).

Постбинарное кодирование интервала, а также обратное декодирование с дальнейшим расчётом среднего арифметического отображено в таблице 4.

– потеря точности границ интервала и конечного результата при значениях множественности в старших разрядах мантиссы;
– необходимость отслеживания некорректных значения множественности в порядке вещественного числа.

Аналогично и для PostBinary 128/32 ip:

положительные стороны:

– высокая точность границ интервала;
– высокая точность результата вычисления;

отрицательные стороны:

– большой расход памяти.

Из сравнения можно сделать вывод, что целесообразнее использовать второй формат. Однако, несмотря на более высокую точность хранения информации, в контексте огромного количества вычислений всей сенсорной составляющей информационно-компьютерной инфраструктуры повышенный объём занимаемой памяти приведёт не только к необходимости задействовать множество дополнительных хранилищ, но и увеличит нагрузку на сетевые системы.

В связи с этим предлагается использование гибридного варианта использования форматов:

- постбинарный формат PostBinary 64/32 р рационально использовать в тех случаях, когда множественность не затрагивает порядок и старшие разряды мантиссы;

- в случаях, когда под угрозой точность хранения информации, целесообразно пожертвовать дополнительной памятью и использовать формат PostBinary 128/32 ip.

Выводы

В статье рассмотрены отдельные детали реализации алгоритма измерения физических величин. С целью анализа его эффективности проведены новые серии измерений. На основании полученных данных сформированы интервалы, которые в дальнейшем прошли через процедуру постбинарного кодирования.

После декодирования интервала и сравнения с изначальными данными выявлено, что в ряде случаев формат PostBinary 64/32 р может снижать точность результата. Предложено использовать формат PostBinary 128/32 ip в случаях, когда формат PostBinary 64/32 р может приводить к ухудшению качества представления информации.

Литература

1. Койбаш, А. А. Пути повышения эффективности устройств сенсорной составляющей информационно-компьютерной инфраструктуры / А. А. Койбаш // Информатика

и кибернетика. – Донецк: ДонНТУ, 2019. – № 2(16). – С. 51-57.

2. Аноприенко, А. Я. Разработка прототипа суперсенсорного компьютера: особенности реализации и визуализации результатов измерений / А. Я. Аноприенко, Р. Л. Варзар // Материалы пятой международной научно-технической конференции «Моделирование и компьютерная графика», 24-27 сентября 2013 г., Донецк. – Донецк: ДонНТУ, 2013. – С. 218-229.

3. Аноприенко, А. Я. Анализ эффективности устройств сенсорной составляющей образовательной системы умного города / А. Я. Аноприенко, А. А. Койбаш, Е. И. Приходченко // Информатика и кибернетика. – Донецк: ДонНТУ, 2020. – № 1(19). – С. 5-11.

4. Аноприенко, А. Я. Системный анализ закономерностей мирового развития компьютерных систем / А. Я. Аноприенко, К. А. Сидоров, Н. С. Максименко, А. А. Койбаш // Информатика и кибернетика. – Донецк: ДонНТУ, 2019. – № 4(18). – С. 41-48.

5. Аноприенко, А. Я. Закономерности развития инерциальных датчиков информационно-компьютерной инфраструктуры / А. Я. Аноприенко, А. А. Койбаш, Н. С. Максименко, К. А. Сидоров // Информатика и кибернетика. – Донецк: ДонНТУ, 2021. – № 1 - 2 (23 - 24). – С. 14-21.

6. Койбаш, А. А. Разработка модифицированного алгоритма измерения физических величин для сенсорных систем / А. А. Койбаш // Информатика и кибернетика. – Донецк: ДонНТУ, 2021. – № 4(26). – С. 63-69.

7. Аноприенко, А. Я. Введение в постбинарный компьютеринг. Арифметико-логические основы и программно-аппаратная реализация / А. Я. Аноприенко, С. В. Иваница. — Донецк: ДонНТУ- УНИТЕХ, 2017. — 308 с.

8. Аноприенко, А. Я. Развитие многомерной логики / А. Я. Аноприенко, С. Г. Джура, С. В. Иваница // Дельфис, 2016. — № 4(88). – С. 85-91.

9. Аноприенко, А. Я. Пример Румпа в контексте традиционных, интервальных и постбинарных вычислений / А. Я. Аноприенко, В. А. Гранковский, С. В. Иваница // Научные труды Донецкого национального технического университета. Серия «Проблемы моделирования и автоматизации проектирования» (МАП-2011). – Донецк: ДонНТУ, 2011. – Вып. 9 (179). – 356 с.

10. Иваница, С. В. Обоснование закономерностей, арифметико-логических алгоритмов и структур систем компьютерной обработки информации / С. В. Иваница // Диссертация на соискание ученой степени кандидата технических наук. – Донецк: ДонНТУ, 2019. – 226 с.

Койбаш А. А. Анализ эффективности модифицированного алгоритма измерения физических величин для сенсорных систем. Рассмотрены детали реализации модифицированного алгоритма измерения физических величин для сенсорных систем, в основе которого лежит постбинарное кодирование информации. С целью проверки работоспособности алгоритма выполнен анализ его эффективности с использованием реального прототипа сенсорной системы. На основании полученных в опытах данных сформированы интервалы и выполнено их постбинарное кодирование с использованием различных форматов. Определены преимущества и недостатки использования каждого формата. На основании результатов экспериментального анализа предложен гибридный вариант постбинарного кодирования информации.

Ключевые слова: сенсоры, сенсорные системы, алгоритм измерения физических величин, микроконтроллер, постбинарный компьютеринг.

Koibash A. Analysis of the effectiveness of the modified algorithm for measuring physical quantities for sensor systems. The details of implementation are considered for the modified algorithm for measuring physical quantities for sensor systems based on post-binary coding of information. Analysis of effectiveness of the algorithm was carried out using a real prototype of the sensor system in order to check its performance. Intervals were formed based on the data obtained in the experiments and their post-binary coding was performed with the usage of various formats. Advantages and disadvantages of using each format are identified. Based on the results of the experimental analysis, a hybrid version of post-binary information coding is proposed.

Keywords: sensors, sensor systems, algorithm for measuring physical quantities, microcontroller, post-binary computing.

Статья поступила в редакцию 09.03.2023
Рекомендована к публикации профессором Мальчевой Р.В.

Особенности преподавания программирования в высшем учебном заведении

М. А. Стрижко, Е. И. Приходченко
Донецкий национальный технический университет, г. Донецк
strizhko.mihail@yandex.ru

Аннотация

В статье рассмотрены самые распространенные особенности преподавания дисциплин, связанных с алгоритмизацией и программированием. Выявлены проблемы, с которыми сталкиваются преподаватели и студенты высшей школы при их освоении. Предложены пути решения таких проблем, основанные на выборе наиболее подходящих методов обучения с учетом специфики предмета. Описанные способы повышения успеваемости студентов сочетают в себе использование соответствующих программных средств и педагогических подходов в образовательном процессе.

Введение

Программирование является одним из самых востребованных направлений профессиональной деятельности в современном мире. С каждым днем все больше и больше компаний и организаций нуждаются в специалистах данной области. Это создает множество возможностей для карьерного роста и развития выпускников высших учебных заведений. Однако, обучение программированию может быть трудным для многих студентов, особенно для тех, кто не имеет опыта в области информационных технологий. Правильное применение различных педагогических подходов и методов играет важную роль в успешном и эффективном обучении программированию, поэтому вопрос подбора методики преподавания с учетом особенностей дисциплины актуален.

В настоящее время существует ряд исследований, посвященных разработке подходов к обучению программированию, среди которых работы И. О. Одинцова [1], Е. А. Ракитиной [2], Е. В. Касьяновой [3], К. И. Буамане [4], В. Ш. Кауфмана [5], И. Н. Скопина [6].

Образовательные стандарты высшего образования по информатике предусматривают, что студенты должны изучать программирование с объектно-ориентированным подходом, что практически невозможно без предварительного базового образования студента. Поэтому, как правило, необходимо начинать с основ. Однако, большинство студентов уже на начальном этапе освоения считают «Алгоритмизацию и программирование» одним из самых сложных предметов и сталкиваются с определенными трудностями. Данное явление можно объяснить тем, что программирование – это вид человеческой деятельности, для успешного освоения которого требуется не только применение знаний и навыков, полученных в

процессе обучения, но и определенный стиль мышления, изначально абстрактный, хоть и связанный с решением конкретных задач. Поэтому программирование следует рассматривать прежде всего как средство развития мышления. Задача преподавателя – не научить студентов писать алгоритмы на языках программирования, а научить их самостоятельно конструировать функции для решения конкретных задач. Знание языка в данном случае – это всего лишь инструмент, и само по себе ничего не дает – гораздо важнее умение им пользоваться. Правильно составленные учебные задачи требуют понимания построения и функционирования информационных систем различных типов и интеграции принципов управления процессами в природе, технике и обществе. Это означает, что необходимо научить студентов мыслить как программистов и самостоятельно обучаться в дальнейшем. Одна из важнейших задач преподавателя – научить студентов воспринимать программирование как искусство, творческую деятельность.

Основное содержание статьи

Пакеты прикладных программ Microsoft Office могут существенно помочь преподавателям информатики в обучении программированию студентов. Например, программный пакет Microsoft Excel с его языком программирования VBA (Visual Basic for Application) – подходящая среда для начального обучения программированию в университетах: при манипуляциях с формулами в таблице Excel у пользователя может сложиться впечатление, что он овладел секретами программы, но для сложной автоматизации необходимо знакомство с более глубокими ресурсами. И главным практическим ресурсом является язык VBA. Среда программирования VBA, относящаяся к категории объектно-ориентированных языков,

является наиболее «прозрачной» среди других известных сред объектно-ориентированного программирования в плане формирования абстрактного мышления в программировании. Освоение VBA можно принять первым шагом, на основе которого показать студентам направление дальнейшего обучения.

Практические занятия со студентами показали, что освоение принципов программирования на основе VBA вполне по силам даже тем студентам, которые ранее мало что знали о нём. И, самое главное, есть абсолютная уверенность в том, что навыки, полученные при работе с Microsoft Excel с использованием VBA пригодятся специалистам в будущем независимо от того, какое направление деятельности они выберут в будущем [1].

Далее важна форма организации образовательной деятельности и методы обучения. Здесь открываются широкие возможности для стратегических образовательных траекторий саморазвития и внедрения в образовательный процесс эвристических (нестандартных) методов. Другими словами, можно отказаться от традиционных структур обучения, которые предоставляют обучаемым большой объем готовых знаний и ожидают, что сильные студенты смогут самостоятельно выстроить общие закономерности и принципы обучения. Как показывает практика, таких студентов совсем не много. Для эффективного обучения преподавателю следует организовать самостоятельную познавательную деятельность студентов на занятии и заставить их думать «о природе вещей» при малом количестве предоставляемой базовой информации. Такая работа гораздо важнее заучивания, и в долгосрочной перспективе дает лучшие результаты. Какие же навыки необходимо развивать при обучении программированию? Можно разделить их на три основные категории:

- 1) умение находить ответы на вопросы;
- 2) умение применять знания (и выбирать правильный инструмент для работы);
- 3) умение решать проблемы (писать программы), находя ответы самостоятельно.

Классификации методов обучения различаются в зависимости от критериев, на которых они основаны. Далее будут рассмотрены основные классификации методов обучения с точки зрения их применения к обучению программированию. По способу передачи информации от преподавателя к обучаемому различают вербальные, наглядные и практические методы обучения [2]. При преподавании курса или раздела программирования используются вербальные (лекционный материал) и практические (лабораторные работы) методы, с акцентом на

усовершенствование навыков, в процессе применения которых студенты не только приобретают новые знания, но и развивают технику исполнения. Преподаватель осуществляет руководство, определяет цель работы, направляет и контролирует ход работы. Деятельность студентов в основном представляет собой практические задания (реальные и абстрактные задачи), в которых особую роль играют самостоятельные мыслительные процессы, позволяющие на примерах находить необходимые данные и решать проблемы. В зависимости от основных типов учебных задач, встречающихся на занятиях, можно выделить методы приобретения знаний, формирования умений, творческой деятельности и методы проверки знаний, умений и навыков. Следует отметить, что все эти методы можно использовать при обучении программированию. Важно также и то, что методология учебной деятельности часто является итеративным и прогрессивным процессом. Речь идёт об итеративных методах обучения. Если повторение рассматривать как пошаговый подход к достижению определенной цели, то итеративные методы обучения можно применять при изложении лекционного материала и проведении лабораторных работ, имеющих связь с информатикой [3]. Это связано с тем, что определение задач, предназначенных для лабораторного практикума, соответствует постепенному итерационному процессу, выражающемуся в построении ряда алгоритмов и программ для решения задач, в то время как следующий алгоритм является усовершенствованием или расширением предыдущего. Построение конечной программы представляет собой итерационный процесс, в котором на каждом этапе вносятся некоторые изменения, поэтому в данной дисциплине может быть применен метод итерационного обучения. Исходя из приведённых выше положений, порядок изложения лекционного материала зависит от порядка проведения практических и лабораторных работ.

Учитывая его повторяющийся характер, логично, что изложение лекционного материала должно быть основано на итерационном методе. В этом случае обучение представляет собой не последовательное изучение новых структур и операторов, а постепенный итерационный процесс выявления и расширения возможностей изучаемой системы.

Для обеспечения преемственности между школьным и университетским курсами информатики, возможно, не стоит давать ученикам школы расширенный курс программирования, а перенести акцент на изучение основ программирования, развивая абстрактный способ мышления [4].

Выводы

Несмотря на то, что навыки программирования становятся всё более востребованными в самых различных сферах деятельности современного человека и с каждым днём всё чаще входят в круг интересов молодых людей, существуют некоторые проблемы при изучении дисциплины, связанные с неправильным подходом в её преподавании. Решение данных проблем возможно путём применения педагогических методов, учитывающих специфику предмета. Алгоритмизацию и программирование предлагается рассматривать как особый способ абстрактного мышления, и овладение этим способом мышления становится более важным, чем изучение конкретного языка.

Литература

1. Одинцов, И. О. Профессиональное программирование. Системный подход. – СПб.: БХВПетербург, 2004. – 624 с.
2. Ракитина, Е. А. Построение методической системы обучения информатике на деятельностной основе: дис. ...д-ра. пед. наук. – М., 2002. – 485 с.
3. Касьянова, Е. В. Адаптивные методы и средства поддержки дистанционного обучения программированию: дис. ...канд. физ.-мат. наук. – Новосибирск, 2006. – 181 с.
4. Баумане, К. И. Методика обучения семиотике языков программирования будущих учителей информатики: дис. ...канд. пед. наук: СПб., 2004. – 289 с.
5. Скопин, И. Н. Проблемные задачи при изучении общих методов информатики и программирования // Вестник Российского университета дружбы народов? – 2011. – №4. – С. 21–33.
6. Кауфман, В. Ш. Языки программирования: концепции и принципы. – М.: ДМК-пресс, 2010. – 464 с.
7. Кашаев, С. М. Программирование в Microsoft Excel на примерах. – СПб.: БХВ-Петербург, 2007. – 320 с.
8. Кузенкова, Г. В. Обучение программированию в средней и высшей школе: проблемы и пути решения / Г. В. Кузенкова, Д. В. Шабазян, А. А. Штанюк // Современные проблемы науки и образования. – М., 2021. – № 1. – С. 13–21.
9. Жужжалов, В. Е. Интеграционные методы изучения программирования в вузовском курсе информатики // Вестник МГПУ. Информатика и информатизация образования. – М., 2003. – № 1 (1). – С. 58–63.
10. Кочеткова, О. А., Пудовкина Ю. Н. Обучение учащихся программированию на языке Python в рамках элективного курса по информатике // Современные проблемы науки и образования. – М., 2019. – № 2. – С. 43–45.

Стрижко М. А., Приходченко Е. И. Особенности преподавания программирования в высшем учебном заведении. В статье рассмотрены самые распространенные особенности преподавания дисциплин, связанных с алгоритмизацией и программированием. Выявлены проблемы, с которыми сталкиваются преподаватели и студенты высшей школы при их освоении. Предложены пути решения таких проблем, основанные на выборе наиболее подходящих методов обучения с учетом специфики предмета. Описанные способы повышения успеваемости студентов сочетают в себе использование соответствующих программных средств и педагогических подходов в образовательном процессе.

Ключевые слова: программирование, педагогические методы, итерационный метод, абстрактное мышление, практические занятия.

Strizhko M. A., Prikhodchenko K. I. Features of teaching programming at a higher educational institution. The article discusses the most common features of teaching disciplines related to algorithmization and programming. The problems faced by teachers and students of higher education in their learning are revealed. The ways of solving such problems are proposed. They are based on the choice of the most appropriate teaching methods, taking into account the specifics of the subject. The described methods of improving students' academic performance combine the use of appropriate software and pedagogical approaches in the educational process.

Keywords: programming, pedagogical methods, iterative method, abstract thinking, practical exercises.

Статья поступила в редакцию 12.03.2023
Рекомендована к публикации профессором Мальчевой Р.В.

Об авторах

Баляба Яна Витальевна – магистрант кафедры программной инженерии им. Л.П. Фельдмана факультета интеллектуальных систем и программирования ГОУ ВПО «Донецкий национальный технический университет».

Васяева Татьяна Александровна - кандидат технических наук, доцент, доцент кафедры автоматизированных систем управления факультета информационных систем и технологий ГОУ ВПО «Донецкий национальный технический университет».

Григорьев Александр Владимирович - кандидат технических наук, доцент, доцент кафедры программной инженерии им. Л. П. Фельдмана факультета интеллектуальных систем и программирования ГОУ ВПО «Донецкий национальный технический университет».

Достлев Юрий Сергеевич – старший преподаватель кафедры компьютерной инженерии факультета интеллектуальных систем и программирования ГОУ ВПО «Донецкий национальный технический университет».

Золушкин Юрий Алексеевич - аспирант кафедры автоматизированных систем управления факультета информационных систем и технологий ГОУ ВПО «Донецкий национальный технический университет».

Зори Сергей Анатольевич - доктор технических наук, доцент, заведующий кафедрой программной инженерии им. Л. П. Фельдмана факультета интеллектуальных систем и программирования ГОУ ВПО «Донецкий национальный технический университет».

Койбаш Александр Андреевич – старший преподаватель кафедры компьютерной инженерии факультета интеллектуальных систем и программирования ГОУ ВПО «Донецкий национальный технический университет».

Мальчева Раиса Викторовна – кандидат технических наук, доцент, доцент кафедры компьютерной инженерии факультета интеллектуальных систем и программирования ГОУ ВПО «Донецкий национальный технический университет».

Мартыненко Татьяна Владимировна - кандидат технических наук, доцент, доцент кафедры автоматизированных систем управления факультета информационных систем и технологий ГОУ ВПО «Донецкий национальный технический университет».

Приходченко Екатерина Ильинична – доктор педагогических наук, профессор, заведующий кафедрой инженерной педагогики и лингвистики ГОУ ВПО «Донецкий национальный технический университет».

Рычка Ольга Валентиновна - кандидат технических наук, доцент кафедры программной инженерии им. Л. П. Фельдмана факультета интеллектуальных систем и программирования ГОУ ВПО «Донецкий национальный технический университет».

Савкова Елена Осиповна - кандидат технических наук, доцент, доцент кафедры автоматизированных систем управления факультета информационных систем и технологий ГОУ ВПО «Донецкий национальный технический университет».

Стрижко Михаил Александрович - аспирант, ассистент кафедры автоматизации и телекоммуникаций факультета компьютерных информационных технологий и автоматизации ГОУ ВПО «Донецкий национальный технический университет».

Терещенко Кирилл Александрович – аспирант кафедры компьютерной инженерии факультета интеллектуальных систем и программирования ГОУ ВПО «Донецкий национальный технический университет».

Чередникова Ольга Юрьевна – кандидат технических наук, доцент, доцент кафедры компьютерной инженерии факультета интеллектуальных систем и программирования ГОУ ВПО «Донецкий национальный технический университет».

Шуватова Екатерина Александровна - ассистент кафедры автоматизированных систем управления факультета информационных систем и технологий ГОУ ВПО «Донецкий национальный технический университет».

Якубовская Юлия Алексеевна – ассистент кафедры компьютерного моделирования и дизайна факультета информационных систем и технологий ГОУ ВПО «Донецкий национальный технический университет».

**Требования к статьям,
направляемым в редакцию научного журнала
«Информатика и кибернетика»**

Редколлегией принимаются к рассмотрению статьи, в которых рассматриваются важные вопросы в области информатики и кибернетики. Научный журнал издаётся с 2015 года, периодичность издания – 4 раза в год.

В журнале предусмотрены следующие рубрики:

- информатика и вычислительная техника;
- компьютерные и информационные науки;
- инженерное образование.

В соответствии с номенклатурой специальностей научных работников МОН ДНР первые две рубрики соответствуют следующим укрупненным группам специальностей научных работников:

05.01 – «Инженерная геометрия и компьютерная графика»,

05.13 – «Информатика, вычислительная техника и управление».

С 01.02.2019 Научный журнал включён в Перечень рецензируемых научных изданий, в которых должны быть опубликованы основные научные результаты диссертаций на соискание учёной степени кандидата наук, на соискание учёной степени доктора наук (приказ МОН ДНР № 135) по группам специальностей 05.01.00 и 05.13.00.

Рубрика «Инженерное образование» предназначена опубликования сотрудниками научно-методических статей.

Журнал также включён в базу данных РИНЦ (Российский индекс научного цитирования) (лицензионный договор № 425-07/2016 от 14.07.2016).

Статьи, представляемые в данный сборник, должны отвечать следующим требованиям. **Содержание статьи** должно быть посвящено актуальным научным проблемам и включать следующие необходимые элементы:

- постановку проблемы в общем виде, её связь с важными научными и практическими задачами;
- анализ последних исследований и публикаций, в которых решается данная задача и на которые опирается автор, выделение нерешенных ранее частей общей проблемы, которым посвящается статья;
- формулировка цели статьи и постановка задач, решаемых в ней;
- изложение основного материала с полным обоснованием полученных научных результатов;
- выводы и перспективы последующих исследований в данном направлении.

Каждый элемент должен быть выделен соответствующим названием раздела, например, «введение», «постановка задачи», «цель и задачи работы», «цель статьи», «цель исследования», «цель разработки», «анализ ... », «сравнительная оценка ... », «разработка ... », «проектирование ... », «программная реализация», «тестирование ... », «полученные результаты», «выводы», «литература». Разделы «введение», «выводы», «литература» являются **обязательными**. Включать в названия разделов нумерацию не разрешается.

В основном тексте статьи формулируются и обосновываются полученные авторами утверждения и результаты. Выводы должны полностью соответствовать содержанию основного текста. Языки публикаций: русский, английский.

Объём статьи, формат страницы

Для оформления статьи следует использовать листы формата А4 (210x297 мм) с полями по 2,5 см со всех сторон. Нумерацию страниц выполнять не нужно.

Рекомендуемый объём статьи – 6-12 страниц. Рукописи меньшего объёма могут быть рекомендованы к публикации в качестве коротких сообщений.

Последняя страница текста статьи должна быть заполнена не менее чем на две трети, но содержать не менее трёх пустых строк в конце.

Форматирование текста

Подготовка статьи осуществляется в текстовом редакторе Microsoft Office Word.

Весь текст статьи оформляется шрифтом Times New Roman 10 пт с одинарным междустрочным интервалом, если ниже в требованиях не сказано иного. Абзацный интервал «перед» – 0 пт, «после» – 0 пт.

На первой строке с выравниванием по левому краю располагается УДК.

Заголовок (название) статьи оформляется шрифтом Times New Roman 14 пт, полужирное начертание, с выравниванием по центру (без абзацных отступов). Заголовок статьи следует печатать с прописной буквы без точки в конце, переносы слов не допускаются. Абзацный интервал «перед» – 12 пт, «после» – 12 пт.

После названия статьи следует информация об авторах, которая выравнивается по центру (без абзацных отступов). На одной строке указываются инициалы и фамилии всех авторов через запятую. Между двумя инициалами ставится пробел. С новой строки указывается название вуза (организации) и город (для каждого автора, если не совпадают). На следующей строке указываются адреса электронной почты (один адрес либо каждого автора – по желанию). Адрес электронной почты оформляется в виде гиперссылки.

К тексту аннотации применяется курсивное начертание, с выравниванием по ширине, отступы слева и справа по 1 см. Заголовок «Аннотация» выделяется полужирным начертанием. Объём аннотации – 450-550 символов (без пробелов). Абзацный интервал «перед» – 12 пт, «после» – 12 пт.

Основной текст статьи разбивается на две колонки шириной по 7,5 см (промежуток между столбцами – 0,99 см), выравнивается по ширине. Абзацный отступ первой строки – 1 см. Автоматический перенос слов не применяется.

Заголовки разделов выполняются шрифтом Arial 10 пт, полужирное курсивное начертание. Абзацный отступ отсутствует, интервал перед абзацем – 12 пт, после абзаца – 6 пт. Для заголовка «Введение» установить интервал «перед» – 0 пт, «после» – 6 пт.

Таблицы в тексте статьи

Название следует помещать над таблицей с абзацного отступа (1 см) в формате: слово «Таблица», пробел, номер таблицы, пробел, тире, пробел, название таблицы. Название таблицы записывают с прописной буквы без точки в конце строки и выравнивают по ширине. В ячейках таблицы устанавливается выравнивание текста по центру по вертикали. По горизонтали текст выравнивается по центру либо по левому краю. Границы ячеек таблицы должны быть только чёрного цвета, толщина линии – 1 пт. На все таблицы должны быть приведены ссылки в тексте статьи, при ссылке следует писать слово «табл.» с указанием её номера, например, «... данные приведены в табл. 5». Таблицы нумеруются в пределах статьи. Таблица располагается сразу после ссылки на неё, если это возможно (например, после окончания абзаца). Если же таблица не помещается на текущей странице, то она должна быть расположена в начале следующей страницы (или колонки). При необходимости допускается включение в статью таблицы, ширина которой превышает ширину колонки. В этом случае таблица и её название размещаются по центру страницы. Таблица не должна выступать за границы полей страницы. Таблица и её название отделяются от основного текста статьи одной пустой строкой до и после.

Рисунки в статье

Ссылки на иллюстрации по тексту статьи обязательны и оформляются в виде «... на рис. 2» и т. п. Рисунок и его подпись выравниваются по центру колонки (без абзацных отступов), положение рисунка – «в тексте». Размещается рисунок после его первого упоминания в тексте, если это возможно (например, после окончания абзаца). Если же иллюстрация не помещается на текущей странице, то она должна быть расположена в начале следующей страницы (или колонки). При необходимости допускается включение в статью рисунка, ширина которого превышает ширину колонки. В этом случае рисунок и его подпись выравниваются по центру страницы. Иллюстрация не должна выступать за границы полей страницы. Подпись рисунка оформляется в формате: слово «Рисунок», пробел, номер иллюстрации, пробел, тире, пробел, название рисунка. Название рисунка записывают с прописной буквы без точки в конце строки. Для подписи иллюстрации применяют курсивное

начертание. Иллюстрация и её подпись отделяются от основного текста статьи одной пустой строкой до и после. Не допускается выполнять рисунки с помощью встроенного графического редактора Microsoft Office Word. Если на иллюстрации имеется текст, размер шрифта должен быть не менее чем аналогичный текст, набранный шрифтом Times New Roman 10-го размера. Иллюстрация не должна содержать много незаполненного пространства.

Формулы

Формулы и уравнения рекомендуется набирать с использованием MathType (предпочтительно) или MS Equation. Формулы и математические символы не должны существенно отличаться по размеру от основного текста. Обязательной является нумерация формул, на которые имеется ссылка в тексте статьи. Ссылки в тексте на порядковые номера формул дают в скобках, например, «... согласно формуле (2)». Формулы размещаются по центру колонки, а их номера – по правому краю. Как для строки с формулой, так и для первой строки пояснений (при наличии), абзацный отступ убирается. Первая строка пояснения начинается со слова «где», после которого следует поставить табуляцию на 1 см, затем само пояснение в формате: символ, подлежащий объяснению, пробел, тире, пробел, поясняющий текст, запятая, обозначение единицы измерения физической величины. Пояснения перечисляются через точку с запятой, выравниваются по ширине. Вторая и последующие строки пояснений начинаются с абзацного отступа (1 см). Весь блок текста, связанный с формулой (только формула, несколько формул подряд или формула с пояснениями), отделяется от основного текста одной пустой строкой до и после. Переносить формулы на следующую строку допускается только на знаках выполняемых операций, причем знак в начале следующей строки повторяют. При переносе формулы на знаке умножения применяют знак «×». Формулы и математические уравнения могут быть записаны в тексте документа, если их высота не превышает высоту строки. При этом следует учитывать, что знаки математических операций отделяются от чисел или символов пробелами с обеих сторон. Например, «Если учесть, что $y < 0$ и $2x + y = 1$, то из формулы (3) можно выразить $x...$ ». К символам, которые приведены в формуле, при дальнейшем их употреблении (в том числе в пояснениях к формуле) должно применяться курсивное начертание. При этом к любым числам (верхние и нижние индексы, содержащие цифры и т.п.), а также к математическим знакам курсивное начертание не применяется. Не допускается вставлять формулы, выполненные в виде рисунков.

Перечисления: оформление списков

Основной текст статьи может содержать перечисления, оформленные в виде маркированного списка. В качестве маркера элемента списка разрешается использовать только короткое тире «–». Каждый элемент перечисления записывается с новой строки с абзацного отступа, равного 1 см. После символа короткого тире текст располагается с отступом в 1,5 см от левой границы строки, выравнивается по ширине, при переносе на новые строки располагается без отступов. Нумерованные и многоуровневые списки включать в статью не разрешается.

Литература

В тексте статьи обязательны ссылки на все литературные источники, номер источника указывается в квадратных скобках. Ссылки на неопубликованные работы не допускаются. Рекомендуемое количество источников, на которые ссылается автор, не менее 10. Перечень источников приводится в порядке их упоминания в статье. Библиографическое описание каждого литературного источника оформляется в соответствии с ГОСТ Р 7.0.100–2018. Перечень литературных источников оформляется в виде нумерованного списка. В качестве маркеров элементов списка используют порядковые арабские цифры с точкой. Каждый источник представляет собой отдельный элемент перечисления, записывается с новой строки с абзацного отступа, равного 1 см. После порядкового номера с точкой текст располагается с отступом в 1,5 см от левой границы строки, выравнивается по ширине, при переносе на новые строки располагается без отступов.

В конце статьи обязательно приводятся аннотации на русском и английском языках, каждая заканчивается перечнем 5-6 ключевых слов.

К тексту аннотации применяется курсивное начертание, с выравниванием по ширине, отступы слева и справа по 1 см. Слово «Аннотация» опускается. Текст аннотации начинается с ФИО авторов и названия статьи, выделяемых полужирным начертанием. Аннотация на русском языке совпадает с аннотацией, приведенной в начале статьи. В тексте аннотации на английском языке после фамилии автора указывается только первая буква имени с точкой. Абзацный интервал «перед» – 12 пт, «после» – 12 пт. Ключевые слова оформляются с новой строки аналогично тексту аннотации. Заголовок «Ключевые слова:» (англ. «Keywords:») выделяется полужирным начертанием. Ключевые слова перечисляются через запятую.

Порядок представления статьи и сопроводительные документы

В редакцию необходимо представить:

- файл с текстом статьи;
- файл, содержащий фамилию, имя и отчество авторов полностью; ученую степень, ученое звание; место работы с полным указанием должности, подразделения и наименования организации, города (страны); номера телефонов и e-mail для связи;
- экспертное заключение о возможности публикации статьи, подписанное руководителем и заверенное печатью организации, в которой работает автор статьи;
- выписка из заседания кафедры или письмо организации с просьбой об опубликовании и указанием, что изложенные в статье результаты ранее не публиковались.

Статьи и сопроводительные документы следует высылать на электронный адрес infcyb.donntu@yandex.ru.

К сведению авторов

Если статья оформлена с нарушением указанных выше требований и правил, редакция после предварительного рассмотрения может отклонить статью.

На рецензирование статьи направляются членам редакционной коллегии журнала. Все статьи публикуются при наличии положительной рецензии.

В статью могут быть внесены изменения редакционного характера без согласования с автором. Ответственность за содержание статьи и качество перевода аннотаций несут авторы.

Публикация статей в научном журнале «Информатика и кибернетика» осуществляется на некоммерческой основе.

Все номера Научного журнала размещаются на сайте <http://infcyb.donntu.ru/>.

CONTENT

Informatics and computer engineering

Development of a speech recognition application <i>Balyaba Ya., Rychka O.V.</i>	5
Neural network model of automated translation <i>Vasyaeva T.A., Zolushkin Yu.A., Martynenko T.V., Shuvatova E.A.</i>	12
Tree adjunct grammar inference as a tool for automatic generalization of algorithms <i>Vorobyov L. O., Grigoriev A. V.</i>	17
Designing the architecture of the electronic money system <i>Malcheva R. V., Tereshchenko K. A.</i>	23
Synthesis and modeling of control of non-stationary sections of a mine ventilation network <i>Cherednikova O. Yu., Dostlev Yu. S.</i>	30
Review of three-dimensional reconstruction methods <i>Yakubovskaya Y.</i>	36
High-performance rendering of 3D images for volumetric spatial visualization systems <i>Zori S.A.</i>	42
Analysis of the effectiveness of the modified algorithm for measuring physical quantities for sensor systems <i>Koibash A.</i>	50

Engineering education

Features of teaching programming at a higher educational institution <i>Strizhko M. A., Prikhodchenko K. I.</i>	59
<u>About Authors</u>	62
<u>Requirements to articles which are sent to the editors' office of the scientific journal "Informatics and Cybernetics"</u>	64

Электронное периодическое издание

Научный журнал

ИНФОРМАТИКА И КИБЕРНЕТИКА

(на русском, английском языках)

№ 1 (31) - 2023

Ответственный за выпуск Р. В. Мальчева

Технический редактор Р. В. Мальчева

Компьютерная верстка Р. В. Мальчева

Подписано к выпуску 15.03.2023. Усл. печ. лист. 7,97. Уч.-изд. лист.5,1.
Адрес редакции: ДНР, 283001, г. Донецк, ул. Артема, 58, ГОУ ВПО «ДонНТУ»,
4-й учебный корпус, к. 36, ул. Кобозева, 17.
Тел.: +7 (856) 301-07-35, +7 (949) 334-89-11
E-mail: infcyb.donntu@yandex.ru, URL: <http://infcyb.donntu.ru>