

Программная инженерия



Пр **7**
ИН **2021**
Том 12

Рисунки к статье Д. Д. Руховича
 «ПРОЕКЦИЯ ИЗОБРАЖЕНИЙ В ПРОСТРАНСТВО
 ДЛЯ МОНОКУЛЯРНОЙ И МУЛЬТИРАКУРСНОЙ
 ТРЕХМЕРНОЙ ДЕТЕКЦИИ ОБЪЕКТОВ В УЛИЧНЫХ СЦЕНАХ»

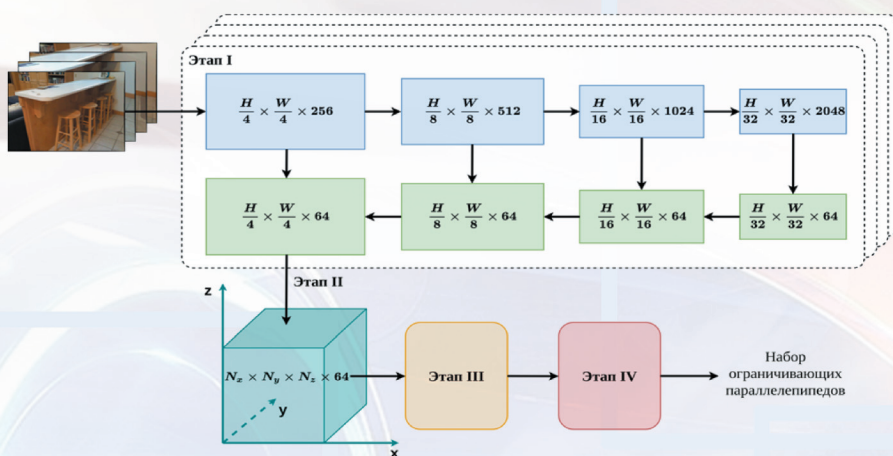


Рис. 1. Схема предложенного метода детекции трехмерных объектов:
 H и W – высота и ширина входного изображения в пикселях;
 N_x , N_y и N_z – число вокселей по осям x , y , z

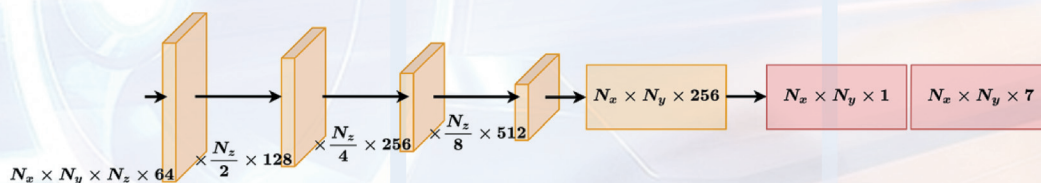


Рис. 2. Нейросетевая архитектура, применяемая на этапах III и IV
 для детекции объектов в уличных сценах



Рис. 3. Объекты, обнаруженные с помощью предложенного метода
 по изображениям из валидационной части набора данных KITTI

Программная инженерия

Том 12
№ 7
2021
Пр
ИН

Учредитель: Издательство "НОВЫЕ ТЕХНОЛОГИИ"

Издается с сентября 2010 г.

DOI 10.17587/issn.2220-3397

ISSN 2220-3397

Редакционный совет

Садовничий В.А., акад. РАН
(председатель)
Бетелин В.Б., акад. РАН
Васильев В.Н., чл.-корр. РАН
Жижченко А.Б., акад. РАН
Макаров В.Л., акад. РАН
Панченко В.Я., акад. РАН
Стемпковский А.Л., акад. РАН
Ухлинов Л.М., д.т.н.
Федоров И.Б., акад. РАН
Четверушкин Б.Н., акад. РАН

Главный редактор

Васенин В.А., д.ф.-м.н., проф.

Редколлегия

Антонов Б.И.
Афонин С.А., к.ф.-м.н.
Бурдонов И.Б., д.ф.-м.н., проф.
Борзовс Ю., проф. (Латвия)
Гаврилов А.В., к.т.н.
Галатенко А.В., к.ф.-м.н.
Корнеев В.В., д.т.н., проф.
Костюхин К.А., к.ф.-м.н.
Махортов С.Д., д.ф.-м.н., доц.
Манцивода А.В., д.ф.-м.н., доц.
Назирова Р.Р., д.т.н., проф.
Нечаев В.В., д.т.н., проф.
Новиков Б.А., д.ф.-м.н., проф.
Павлов В.Л. (США)
Пальчунов Д.Е., д.ф.-м.н., доц.
Петренко А.К., д.ф.-м.н., проф.
Позднеев Б.М., д.т.н., проф.
Позин Б.А., д.т.н., проф.
Серебряков В.А., д.ф.-м.н., проф.
Сорокин А.В., к.т.н., доц.
Терехов А.Н., д.ф.-м.н., проф.
Филимонов Н.Б., д.т.н., проф.
Шапченко К.А., к.ф.-м.н.
Шундеев А.С., к.ф.-м.н.
Щур Л.Н., д.ф.-м.н., проф.
Язов Ю.К., д.т.н., проф.
Якобсон И., проф. (Швейцария)

Редакция

Лысенко А.В., Чугунова А.В.

Журнал издается при поддержке Отделения математических наук РАН, Отделения нанотехнологий и информационных технологий РАН, МГУ имени М.В. Ломоносова, МГТУ имени Н.Э. Баумана

СОДЕРЖАНИЕ

- Кодубец А. А., Артемьева И. Л.** Обзор исследований в области инженерии требований для программных систем 339
- Шевелев С. С.** Устройство выполнения логических и арифметических операций 350
- Орлова Е. В.** Предиктивная аналитика кредитных рисков на основе данных цифровых следов заемщиков и методов статистического машинного обучения 358
- Рухович Д. Д.** Проекция изображений в пространство для монокулярной и мультиракурсной трехмерной детекции объектов в уличных сценах ... 373
- Торгашев М. А., Страшнов Е. В.** Определение коллизий аппроксимирующих цилиндров в системах виртуального окружения 385

Журнал зарегистрирован
в Федеральной службе
по надзору в сфере связи,
информационных технологий
и массовых коммуникаций.

Свидетельство о регистрации

ПИ № ФС77-38590 от 24 декабря 2009 г.

Журнал распространяется по подписке, которую можно оформить в любом почтовом отделении (индекс по Объединенному каталогу "Пресса России" — 22765) или непосредственно в редакции.

Тел.: (499) 269-53-97. Факс: (499) 269-55-10.

Http://novtex.ru/prin/rus E-mail: prin@novtex.ru

Журнал включен в систему Российского индекса научного цитирования и базу данных RSCI на платформе Web of Science.

Журнал входит в Перечень научных журналов, в которых по рекомендации ВАК РФ должны быть опубликованы научные результаты диссертаций на соискание ученой степени доктора и кандидата наук.

© Издательство "Новые технологии", "Программная инженерия", 2021

SOFTWARE ENGINEERING

PROGRAMMNAYA INGENERIA

Vol. 12

N 7

2021

Published since September 2010

DOI 10.17587/issn.2220-3397

ISSN 2220-3397

Editorial Council:

SADOVNICHY V. A., Dr. Sci. (Phys.-Math.),
Acad. RAS (*Head*)
BETELIN V. B., Dr. Sci. (Phys.-Math.), Acad. RAS
VASIL'EV V. N., Dr. Sci. (Tech.), Cor.-Mem. RAS
ZHIZHCENKO A. B., Dr. Sci. (Phys.-Math.),
Acad. RAS
MAKAROV V. L., Dr. Sci. (Phys.-Math.), Acad.
RAS
PANCHENKO V. YA., Dr. Sci. (Phys.-Math.),
Acad. RAS
STEMPKOVSKY A. L., Dr. Sci. (Tech.), Acad. RAS
UKHLINOV L. M., Dr. Sci. (Tech.)
FEDOROV I. B., Dr. Sci. (Tech.), Acad. RAS
CHETVERTUSHKIN B. N., Dr. Sci. (Phys.-Math.),
Acad. RAS

Editor-in-Chief:

VASENIN V. A., Dr. Sci. (Phys.-Math.)

Editorial Board:

ANTONOV B.I.
AFONIN S.A., Cand. Sci. (Phys.-Math)
BURDONOV I.B., Dr. Sci. (Phys.-Math)
BORZOV JURIS, Dr. Sci. (Comp. Sci), Latvia
GALATENKO A.V., Cand. Sci. (Phys.-Math)
GAVRILOV A.V., Cand. Sci. (Tech)
JACOBSON IVAR, Dr. Sci. (Philos., Comp. Sci.),
Switzerland
KORNEEV V.V., Dr. Sci. (Tech)
KOSTYUKHIN K.A., Cand. Sci. (Phys.-Math)
MAKHORTOV S.D., Dr. Sci. (Phys.-Math)
MANCIVODA A.V., Dr. Sci. (Phys.-Math)
NAZIROV R.R. , Dr. Sci. (Tech)
NECHAEV V.V., Cand. Sci. (Tech)
NOVIKOV B.A., Dr. Sci. (Phys.-Math)
PAVLOV V.L., USA
PAL'CHUNOV D.E., Dr. Sci. (Phys.-Math)
PETRENKO A.K., Dr. Sci. (Phys.-Math)
POZDNEEV B.M., Dr. Sci. (Tech)
POZIN B.A., Dr. Sci. (Tech)
SEREBRJAKOV V.A., Dr. Sci. (Phys.-Math)
SOROKIN A.V., Cand. Sci. (Tech)
TEREKHOV A.N., Dr. Sci. (Phys.-Math)
FILIMONOV N.B., Dr. Sci. (Tech)
SHAPCHENKO K.A., Cand. Sci. (Phys.-Math)
SHUNDEEV A.S., Cand. Sci. (Phys.-Math)
SHCHUR L.N., Dr. Sci. (Phys.-Math)
YAZOV Yu. K., Dr. Sci. (Tech)

Editors: LYSENKO A.V., CHUGUNOVA A.V.

CONTENTS

Kodubets A. A., Artemieva I. L. Requirements Engineering for Software Systems: A Systematic Literature Review	339
Shevelev S. S. Device for Calculating Logical and Arithmetic Operations	350
Orlova E. V. Predictive Credit Risk Analytics Using Borrowers' Digital Footprint and Methods of Statistical Machine Learning	358
Rukhovich D. D. 2D-to-3D Projection for Monocular and Multi-View 3D Object Detection in Outdoor Scenes	373
Torgashev M. A., Strashnov E. V. Collision Detection of Bounding Cylinders in Virtual Environment Systems	385

А. А. Кодубец, аспирант, alexey@kodubets.ru,
И. Л. Артемьева, д-р техн. наук, проф., зав. кафедрой, iartemeva@mail.ru,
Дальневосточный федеральный университет, Владивосток

Обзор исследований в области инженерии требований для программных систем

Представлен краткий обзор состояния исследований за последние пять лет в области инженерии требований для сложноорганизованных программных систем. Сформулирована проблема исследования в области инженерии требований для промышленно создаваемых программных систем. Проведена классификация направлений исследований. Проанализированы результаты по каждому направлению исследований в контексте их вклада в разрешение сформулированной проблемы.

Ключевые слова: инженерия требований, спецификация требований программного обеспечения, промышленно создаваемые программные системы

Введение

При разработке средних и крупных промышленно создаваемых программных систем требования значительно влияют на успех проекта [1, с. 4]. Требования определяют функции, поведение и атрибуты системы, используются как формальный механизм коммуникации между заказчиком и исполнителем. На основе требований к программной системе создается проект, архитектура, определяется дизайн ее компонентов, ведется разработка кода и выполняется тестирование. Требования являются основой для формальной приемки программного продукта заказчиком по окончании проекта. Они важны при создании не только программных, но и программно-аппаратных систем.

Систематизацией методов и подходов в области выявления, анализа, разработки, документирования требований для программных систем, а также поддержанием требований на протяжении жизненного цикла программного обеспечения (ПО) занимается дисциплина, называемая инженерией требований (*Requirements Engineering*). Эта инженерная дисциплина рассматривается как самостоятельный компонент программной и системной инженерии [2, с. 143]. В дальнейшем по тексту (для сокращения) термины "требования" и "инженерия требований" используются в смысловых значениях требований к программным системам.

Существуют международные стандарты, в которых определяются процессы управления требованиями (СММi[3], ISO/IEC/IEEE 12207:2017 [4]). Работа над требованиями выполняется всеми организациями в индустрии, однако многие организации продолжают заниматься этой деятельностью недостаточно системно, без должного использования нормативных процессов (согласно монографии Д. Леффингуэлл и Д. Уидриг "Принципы работы с требованиями к программному обеспечению") [5, с. 43], принятых в этой области.

Проблема и цель

Для разработки требований необходимо взаимодействие заказчика и исполнителя с момента принятия решения о необходимости создания программной системы и постановки целей. Необходимость этого взаимодействия обусловлена перечисленными далее задачами.

- Задачи заказчика:
 - предоставление целей и ограничений для создаваемой системы;
 - подготовка ответов на вопросы исполнителя;
 - утверждение формально задокументированных требований.
- Задачи исполнителя:
 - анализ проблемы;
 - выявление и документирование требований;
 - проведение внутреннего ревью требований с релевантными экспертами;
 - проверка того факта, что требования соответствуют необходимым критериям качества;
 - согласование формально задокументированных требований с заказчиком.

Представленный список задач может дополняться в зависимости от области приложения решаемой задачи, структуры ответственности и других факторов. Например, если заказчик является исполнителем части компонентов программной системы, он будет являться одновременно еще и экспертом соответствующей области, а также будет предоставлять ограничения компонентов и отвечать за техническое исполнение соответствующих требований. Другой пример: если исполнитель является поставщиком уже готового компонента (продукта), на основе которого разрабатывается необходимая программная система, то соответствующие требования, цели и ограничения компонента будут определяться в значительной части самим исполнителем.

В проектах, ориентированных на создание системы в относительно простых для реализации предметных областях, когда представитель заказчика один, эксперт один и система имеет небольшое число требований, процесс взаимодействия, координации и контроля разработки требований не нуждается в больших трудозатратах. Работа сводится к анализу и документированию требований.

По мере роста сложности программной системы число требований увеличивается. Например, промышленно создаваемые программные системы содержат тысячи требований ([1, с. 179; 5, с. 126]; Примеры таких проектов: Siemens [6, с. 2], QT Company [7, с. 5]). Рост сложности программной системы вызывается (но не ограничивается) следующими характеристиками:

- увеличением числа элементов (компонентов, подсистем) внешних и внутренних, а также увеличением числа связей между ними, усложнением самих элементов;
- сложностью предметной области (междисциплинарность, многоуровневой структурой предметной области);
- сложностью окружения целевой системы;
- масштабом решаемой прикладной задачи.

В силу такого большого масштаба подлежащей решению задачи, ее междисциплинарного характера и ограниченных сроков ее решения, необходимо архитектурное разбиение целевой системы на подсистемы, компоненты и структурирование работ через специализацию и разделение труда. В таких условиях один человек уже не способен полностью осознать и проанализировать всю систему, включая ее окружение и поведение, необходимые для ее функционирования компоненты, структуру связей между ними. Структура разделения труда (ответственности) может быть достаточно сложной как вертикально — заказчик, подрядчики, внутренние или внешние субподрядчики, — так и горизонтально, когда инфраструктурные компоненты пронизывают различные уровни системы, затрагивая многих участников проекта (здесь и далее под участниками проекта понимаются как индивидуальные участники, так и представители вовлеченных команд, а также представители внутренних или внешних организаций). Например, структура разделения труда поставщика автомобильных компонентов имеет 35 команд по разработке программного обеспечения, и чтобы получить информацию даже от части этих команд по простым вопросам необходимо несколько дней [8, с. 28]. Другой пример структуры разделения труда: компания Siemens имеет 50 доменов (инженерных ролей на проекте) в области железнодорожной автоматизации [6, с. 2]. При этом отдельные участники могут иметь собственное видение цели, плана ее реализации, мотивов, которые на начальном этапе не всегда согласуются друг с другом, и требуется координация для выработки единого виденья проекта. Например, чтобы показать, что система в целом технически реализуема, необходимо доказать, что отдельные компоненты системы технически реализуемы в режиме их изоляции друг от друга и что не-

обходимые комбинации взаимодействия компонентов и подсистем тоже технически реализуемы. При этом необходимо учитывать, что упрощения в одном компоненте могут создавать сложности в реализации другого компонента и наоборот.

Все представленные выше доводы свидетельствуют о том, что вместе с увеличением сложности системы и числа требований, увеличивается и число представителей заказчика, и число вовлекаемых внутренних экспертов исполнителя, привлекаются внешние и внутренние субподрядчики. При таком значительном увеличении числа участников, разработка требований начинает в основном зависеть от управления коллективом участников процесса и их эффективного взаимодействия [9, с. 145]. Трудозатраты на документирование и разработку требований возрастают при этом линейно. Собрать всех участников вместе по каждому вопросу неэффективно. Собрания и взаимодействие структурируются через специализацию и разделение труда, с учетом структуры ответственности и решаемых вопросов. Число вариантов взаимодействия участников процесса в упрощенной модели имеет квадратичную зависимость (число сочетаний

$$C_n^2 = \frac{n^2}{2} - \frac{n}{2}, \text{ где } n \text{ — число участников процесса),}$$

если взаимодействуют любые два участника. В худшем случае, при большом числе участников, взаимодействие имеет экспоненциальную зависимость

$$\left(\sum_{k=2}^n C_n^k = \sum_{k=0}^n C_n^k - C_n^0 - C_n^1 = 2^n - n - 1 \right).$$

Эти оценочные формулы сделаны по аналогии с теорией Грайкунаса [10, с. 186], где рассчитывается максимальное число вариантов связей между сотрудниками одного руководителя. Из квадратичной и экспоненциальной зависимостей следует, что все варианты взаимодействия от двух и более участников значительно увеличивают трудозатраты на коммуникацию. Таким образом, по мере роста сложности системы критический путь процесса разработки требований начинает зависеть в основном от взаимодействия, коммуникации, координации и контроля участников процесса разработки требований. Будем считать, что в этом случае имеет место *проблема коммуникации, координации и контроля* процессов разработки требований для промышленно создаваемых программных систем. Иногда эту проблему называют так: *"организационные и коммуникационные проблемы инженерии требований"* [9]. Далее для краткости будем использовать сочетание *"проблема коммуникации, координации и контроля"*.

Многие исследования в области инженерии требований *прямо* или *опосредовано* связаны с уменьшением влияния от описанной выше проблемы. Например, поиск орфографических ошибок в требованиях несомненно влияет на эффективность работы с требованиями. Однако в данном примере вклад в разрешение проблемы *опосредованный*, поскольку речь идет о качестве требований, а не об эффективности коммуникации с экспертами. Другой пример:

допустим, требования технически изложены качественно, но требуется подтверждение от большого числа экспертов по различным группам требований. У каждого эксперта могут быть свои вопросы, зависимости или даже противоречащая другим экспертам точка зрения. В данном примере вклад в разрешение проблемы коммуникации, координации и контроля *прямой*, поскольку речь идет именно о координации и коммуникации с экспертами.

Цель обзора — рассмотрение направлений исследований в области инженерии требований, а также анализ того, насколько обозначенная выше *проблема* представлена в этих исследованиях и насколько исследования дают *прямой* вклад в уменьшение или разрешение этой проблемы.

Структура обзора

На самом верхнем уровне процессы инженерии требований делятся на перечисленные далее два больших класса согласно модели СММИ [3, с. 325—348].

Класс 1. Разработка требований (*Requirements Development*), это класс, содержащий: процессы выявления требований (*Requirements Elicitation*); анализ требований (*Requirements Analysis*), включая, при необходимости, моделирование, эскизное проектирование архитектуры, проверку технической реализуемости; непосредственно документирование требований уровня пользователя, заказчика, продукта и его компонентов; процессы валидации требований; коммуникации с заказчиком и заинтересованными сторонами.

Класс 2. Управление требованиями (*Requirements Management*), это класс, содержащий: процессы управления требованиями продукта и его компонентов, включая изменение требований и анализ потенциального влияния изменений на проект — трассировку требований (*Requirements Traceability*); приоритизацию требований (*Requirement Prioritization*) в целях согласования требований, проектных планов и рабочих продуктов.

Существуют системы управления требованиями (*Requirements Management Tool*, IBM Doors [11], Jama [12]), которые позволяют повысить эффективность работы инженера по требованиям на простых рутинных операциях за счет следующих компонентов: специализированного редактора текстов (специализированный аналог MS Word); визуального редактора моделей (специализированный аналог MS Visio); подсистемы трассировки требований (специализированный аналог MS Excel).

В промышленно создаваемых программных системах сложность инженерии требований вызывается сложностью предметной области и сложностью создаваемой системы, ее окружением, компонентами, а также наличием ограничений технического, экономического, управленческого и социального характеров. Задачи анализа и взаимодействия с представителями заказчика и доменными экспертами таких систем слабо поддаются автоматизации. Такие задачи продолжают выполняться инженерами по требованиям ПО в ручном режиме даже при наличии существующих систем управления требованиями.

Современные исследования в области инженерии требований сфокусированы: на разработке новых методов, моделей, подходов и процессов упрощения работы с требованиями; на уменьшении зависимости от человеческого фактора, и, как следствие, на повышении эффективности работы инженера по требованиям.

Кроме процессов инженерии требований, исследователи уделяют внимание специфике, присущей отдельным видам требований, а именно — бизнес- и пользовательским требованиям; нефункциональным требованиям (к надежности и безопасности); пользовательскому интерфейсу.

Также уделяется внимание в исследованиях известным характеристикам качества требований: проверяемости, выполнимости, атомарности, консистентности, модифицируемости, недвусмысленности, завершенности, обязательности и актуальности.

Инженерии требований посвящены более 50 монографий иностранных авторов. В отечественной литературе тема инженерии требований представлена в основном в качестве отдельных глав книг по программной инженерии в целом. На русском языке доступны несколько переводов иностранной литературы [1, 5].

При подготовке обзора проведен анализ иностранных и отечественных публикаций в области инженерии требований за последние 5 лет. На первом этапе для поиска использовались ключевые слова "требования" и "requirements" в названиях публикаций, а также дополнительно фильтр по computer science и техническим наукам по следующим базам данных: 1) РИНЦ; 2) Scopus; 3) Springer (для поиска журналов); 4) сайт ВАК (для поиска диссертации, а из них — поиска релевантных публикаций). Публикации из журналов и конференций, специализирующихся только на инженерии требований (подобно [7, 9]), анализировались полностью независимо от наличия ключевых слов в названии. Результат первого этапа — это более 600 публикаций. На втором этапе анализировалась аннотация, а также для неоднозначных случаев анализировались введение и заключение. Результат второго этапа — оставлены более 300 публикаций, которые непосредственно имеют отношение к инженерии требований. На третьем этапе происходила выработка классификации направлений исследований посредством полного чтения и анализа найденных материалов (более 300 публикаций) с предыдущего этапа. После создания классификации внутри направлений близкие работы группировались (например, если один и тот же автор). Таким образом, получилось более 40 групп. Далее, из каждой группы была взята одна наиболее показательная публикация, и на их основе был подготовлен данный обзор.

Итак, на основе анализа выявлены следующие направления исследований, связанных с инженерией требований:

- автоматический анализ и обработка требований как текста на естественном языке (*Natural Language Processing for Requirements Engineering — NLP4RE*);
- приоритизация требований (*Requirement Prioritization*);
- трассировка требований (*Requirements Traceability*);

- качество спецификаций требований (*Quality of Software Requirements*);
- нефункциональные требования (*Non-functional Requirements — NFR*);
- сбор и выявление требований (*Requirements Elicitation*).

Автоматический анализ и обработка требований как текста на естественном языке

На всех этапах жизненного цикла ПО есть сущности (элементы), зафиксированные частично или полностью текстом на естественном языке, например, модели, документы, проектные решения и т. п. По этой причине направление обработки текста на естественном языке широко представлено в исследованиях различных стадий жизненного цикла ПО.

В обзоре [13] представлены результаты 81 исследования в области обработки текста на естественном языке в целях повышения качества ПО. Было выявлено, что по статистике исследования в области обработки текста требований на естественном языке занимают второе место после обработки текста описания программных дефектов.

Каждое отдельное требование имеет уникальный идентификатор и представляет собой утверждение на естественном языке. Исследователи проводят автоматический анализ спецификаций требований для различных целей, также как это делает инженер. Сама автоматизация выполняется с использованием последних достижений науки и техники, включая алгоритмы поиска, машинное обучение, искусственные нейронные сети и т. п.

Первая подгруппа задач, решаемых в исследованиях, связана с качеством единичных требований [14]:

- автоматический анализ корректности формулировок требований, например, обнаружение сложных предложений в тексте, использование отрицания, пророчечий и т. п.;
 - определение степени полноты требований через автоматический анализ детализации текста, например, односложное требование может не иметь достаточной детализации;
 - проверка совместимости формулировок требований с практиками из индустрии, например, таких как INCOSA [15], которая позволяет избегать расплывчатых изложений "много", "мало" [15, с. 57].
- Вторая подгруппа решаемых задач связана со структурой, версиями и семантикой требований:
- поиск дубликатов на основе построения "дистанции похожести" между утверждениями в требованиях [16] с использованием методов ранжирования поисковиков (BM25);
 - поиск зависимостей — когда требования чрезмерно часто ссылаются друг на друга, образуя циклические зависимости [7];
 - анализ консистентности — когда в одних требованиях описываются одни параметры с одним значением, а в других требованиях те же параметры имеют противоположные значения [17];

- поиск различий между версиями требований [18];
- определение синонимов в тексте требований [19].

Третья подгруппа задач связана с извлечением информации, структурированием и классификацией требований для поддержки деятельности инженера:

- автоматическое определение домена (ответственной команды) методами машинного обучения из текста требований [6];
- извлечение значений параметров создаваемой системы из текста требований [20];
- извлечение ограничений системы из текста (*requirements constrains mining*) [21];
- поиск информации в требованиях запросом на естественном языке путем построения онтологии домена и обработки текста [22] — исследование проблемы сценария, когда ответ на вопрос не находится в требованиях простым поиском по ключевым словам, однако информация в документе присутствует, если перечитать весь документ;
- автоматическая классификация типов требований (функциональные, нефункциональные) на основе машинного обучения [23].

Четвертая группа задач связана со сменой представления требований. Если задать ограничения для изложения текста на естественном языке, то предоставляется возможность приблизить формулировки к декларативному языку со строгой спецификацией. В этом случае возможна конвертация требований в другие представления для задач моделирования и тестирования [24].

Исследования в области обработки требований как текста на естественном языке носят целенаправленный характер. Подтверждение этому находится в ежегодных специализированных конференциях, которые фокусируются на этом направлении исследований, например, NLP4RE (*Natural Language Processing for Requirements Engineering*) [7].

Многие исследования в этой области содержат описание готовых прототипов инструментальных средств на основе предложенных методов и подходов, а в некоторых публикациях результаты исследований оформлены в виде готовых коммерческих систем [14].

Таким образом, исследования в области обработки текста требований на естественном языке являются важным направлением, позволяющим повысить эффективность работы инженеров по требованиям. Как следствие, можно прогнозировать появление большего числа инструментальных средств для анализа и обработки текста требований на естественном языке. Многие исследователи сфокусированы на анализе требований, задокументированных на английском и немецком языках. Исследователям предстоит еще провести адаптацию этих результатов для других языков.

Приоритизация требований

Инженерия требований оказывает значительное влияние на деятельность по управлению проектами. От объема требований, их полноты и корректности представления зависит бюджет проекта в целом. От постановки требований зависит, насколько сложно

технически реализовать (*Feasibility study*) запрашиваемые функциональные возможности целевой системы. Нередко одни функциональные требования реализуемы, а другие нет, хотя с точки зрения конечного пользователя и те и другие функциональные возможности могут быть эквивалентны. Инженер по требованиям выравнивает ожидания между заказчиком программной системы и командой разработки. Например, нужно, чтобы система работала 1 час или работала в режиме 24/7. Что в этом случае приоритетней для ускорения запуска или выполнения проекта? Трудозатраты на разработку программной системы обслуживания в режиме 24/7 значительно отличаются от трудозатрат на разработку систем, которые должны работать в течение одного часа. При этом сами функциональные требования к системе могут быть одинаковыми. Инженер по требованиям занимается приоритизацией требований — иногда это называют *Requirements Triage* [25] — сортировка требований, чтобы максимизировать или минимизировать значение целевой функции проекта, вытекающей из целей заказчика, исполнителей и ограничений. В зависимости от фазы проекта (предпроектная работа или проектная работа) в процесс приоритизации требований может вовлекаться вся команда проекта и представители заказчика.

Бизнес-цели могут быть противоречивы. Например, получить максимум функциональных возможностей целевой системы при минимальных затратах на разработку или минимальной стоимости изделия (минимизировать требования к параметрам оборудования, на котором будет работать ПО). В случае таких противоречивых целей от инженера по требованиям необходим высокий уровень творчества и реализации инженерного подхода, в том числе — использование математических методов линейного программирования для поиска баланса и понимания, какие функциональные возможности главные, а какие носят второстепенный характер.

Еще одна задача, решаемая приоритизацией, это сортировка и определение ключевых пожеланий заказчика, когда объем вводных данных от заказчика большой до начала работ над требованиями. Вводные данные могут иметь размер сотни страниц неструктурированного текста и вопрос, что главное, а что второстепенное, напрямую влияет на эффективность разработки требований. Либо обратная ситуация, когда объем вводных данных мал и необходима детализация требований, чтобы был предмет приоритизации.

Вопросам приоритизации требований посвящены исследования на следующих направлениях:

1) методы приоритизации требований на основе пользовательской обратной связи, когда вследствие огромного числа пользователей нельзя провести опрос классическими методами, что актуально для современных социальных сетей и мобильных приложений, в которых число пользователей измеряется миллионами [26];

2) подходы приоритизации требований в Open Source, когда нет единого заказчика, а в качестве заказчика выступает социум разработчиков и пользователей [27];

3) приоритизация требований в проектах с большим числом лиц, принимающих решения [28];

4) геймификация в управлении требованиями для различных целей: обучение выявлению и приоритизации требований ([29] — обзор 13 исследований в области геймификации в инженерии требований).

Упомянутая выше *проблема коммуникации, координации и контроля* проявляется в том числе в процессе приоритизации требований, когда участников приоритизации становится слишком много, и требуются медиация и групповая выработка решений [28].

Трассировка требований

Таблицы трассировки (*Traceability Matrix*) широко используются в программной инженерии. Они позволяют убедиться, что создаваемые элементы (сущности) проекта соответствуют элементам, созданным на предыдущих этапах жизненного цикла ПО. Это один из методов обеспечения качества.

Наиболее часто применяемые таблицы трассировки требований (*Requirements Traceability Matrix* — RTM) — это трассировка на дизайн и на тесты. Такие таблицы связывают сущности разных дисциплин программной инженерии в проекте. Есть RTM-таблицы, которые находятся только внутри инженерии требований, например:

- системные требования на требования к компонентам — используются для проверки, что все функции системы реализуются компонентными требованиями;

- системные требования на бизнес-требования — используются для проверки, что ранние бизнес-требования и пожелания заказчика учтены в системных требованиях;

- анализ требований проекта или конкретных функций и трассировка результатов анализа на системные требования — используются для проверки того, что все требования проекта или функции целевой системы проанализированы.

Как было отмечено ранее, промышленно создаваемые программные системы содержат от тысячи и более единичных требований. Чтобы уменьшить вероятность человеческих ошибок при работе с таким объемом информации, необходимы средства для формальной проверки соответствия проектных элементов (проектных решений, сущностей и т. п.) спецификациям требований. Трассировка требований является эффективным средством такой формальной проверки. В то же время при слабой структуре требований и большом их числе поддержание RTM-таблиц в актуальном состоянии становится трудозатратным процессом [29]. В связи с этим обстоятельством существует большое число инструментальных средств для автоматизации этого процесса. Однако и такая автоматизация требует ручного вмешательства, чтобы подтвердить актуальность RTM-таблиц.

Вопросам трассировки требований посвящены следующие исследования:

1) разработка методов построения RTM-таблиц — трассировка на соответствие дизайну, на исходный код программы и на тесты [30];

1.1) определение связанности компонентов программной системы;

1.2) отслеживание статуса разработки;

1.3) отслеживание влияния изменений в требованиях на компоненты системы;

2) способы применения обязательных стандартов ISO (например, для автомобильной индустрии) в области трассировки [31];

3) методы трассировки требований на архитектурные уровни (подсистемы, компоненты) [32];

4) трассировка между тремя и более сущностями поддисциплин программной инженерии одновременно (такое-то требование реализовано в таком-то компоненте и проверяется в таком-то тесте "implemented in and verified by") [33].

Исследователи в области трассировки требований в основном направляют усилия на этапы жизненного цикла ПО, когда системные требования разработаны и согласованы, а *проблема коммуникации, координации и контроля* проявляется в меньшей степени. Как правило, на этих этапах жизненного цикла происходят небольшие изменения в требованиях, что связано с минимизацией влияния изменений на сроки выполнения проекта. Трассировка требований эффективно помогает отслеживать влияние таких небольших изменений на разрабатываемую систему и соответствующие элементы проекта (проектные решения, тесты и т. п.). Такой подход делает коммуникацию со всеми участниками проекта более формальной. Однако если для развития или модернизации существующей системы необходимо крупное изменение [18] в согласованных требованиях, то такое изменение будет иметь проблему, аналогичную *проблеме коммуникации, координации и контроля*, т. е. в таких случаях полезность существующих RTM-таблиц (требования на тесты, требования на дизайн) минимальна.

Качество спецификаций требований

На ранних этапах жизненного цикла ПО от заказчика к поставщику приходят слабоструктурированные требования, иногда их называют "пожеланиями". Как было ранее сказано, заказчик определяет проблему, цель, ограничения и другие вводные данные для создаваемой программной системы, но заказчик не обязан иметь все необходимые компетенции для технической разработки детальных требований к создаваемой программной системе. Заказчик не обязан являться доменным экспертом так, что сразу заботиться о том, чтобы пожелания (ранние требования) удовлетворяли всем характеристикам качества требований, в том числе технической реализуемости, и т. п. В самом сложном случае, когда заказчик не имеет необходимой экспертизы, он может предоставить только проблему и цель, а все ограничения, функциональные и нефункциональные требования необходимо выявить поставщику самостоятельно. Поэтому техническая проработанная спецификация требований на основе пожеланий заказчика и других источников требований (стандартов, аналитики подобных программных систем и т. п.) — это первый

и очень важный инженерный элемент проекта, который получает заказчик от поставщика. Шансы на выигрыш тендера у поставщика с низкокачественными спецификациями требований ниже при прочих равных.

Со стороны исполнителя в современных организациях спецификация требований является первоначальным документом, необходимым на входе во многие бизнес-процессы. Последствия низкого качества требований могут быть технически и экономически крайне серьезны, или даже фатальны для организации (потеря заказчиков, убыток и т. п.). Поэтому важно, чтобы спецификация требований имела высокое качество.

Многие описанные выше исследования направлены в том числе и на повышение качества спецификаций. В инженерии требований существуют методы трассировки и инспектирования требований в целях повышения их качества. Исследования в области качества спецификаций требований имеют неизменно высокую практическую значимость. Зная систематические причины дефектов в требованиях, можно выстраивать процессы для повышения качества. Поиском ответов на такие вопросы занимаются следующие исследования:

- разработка методов повышения качества спецификаций через построение дихотомии типовых ошибок, ее использование на этапе разработки требований и на этапе инспектирования в качестве контрольного списка [34];

- применение онтологий инженерии требований и онтологий предметных областей для автоматической проверки качества спецификаций требований [35];

- выявление незавершенных требований через низкий уровень детализации дочерних спецификаций компонентных требований [36];

- создание новых терминов [37]: "технический долг требований" (по аналогии с техническим долгом программного кода) — накопленные проблемы в требованиях, связанные с пренебрежением к качеству спецификаций, вызывающие дополнительные и не всегда предсказуемые трудозатраты в будущем; "вредность требований" — требования, неявно имеющие низкое качество, например, субъективное изложение, использование метафор и т. п., — такие "вредные" требования сложно выявить методами NLP.

Заказчик естественно хочет получить решение быстрое и качественное. Однако для того чтобы сделать решение качественно, исполнителю необходимо время. Это время значительно увеличивается по мере роста сложности создаваемой программной системы, в том числе в силу *проблемы коммуникации, координации и контроля*. Это объективный конфликт между заказчиком и исполнителем. Чтобы уменьшить издержки этого конфликта, нужно решить или уменьшить эту проблему, и таким образом сэкономить время. Метод из исследования [34] ("формирование контрольных списков через анализ типичных человеческих ошибок") затрагивает одну из сторон проблемы, в частности, сферу контроля. Такой подход является одним из компонентов при выработке способов разрешения проблемы. Подход из исследования [35]

("построение онтологий предметных областей для автоматической проверки качества спецификаций требований") может частично дать вклад в разрешение проблемы взаимодействия с экспертами. Если иметь онтологию и соответствующую базу знаний, гипотетически потребуется меньше взаимодействовать с релевантными экспертами. Однако этот подход создает проблему наполнения и поддержания базы знаний в новых или быстро изменяемых предметных областях. Также этот подход не дает вклада в разрешение проблемы контроля и координации.

Нефункциональные требования

Нефункциональные требования относятся к требованиям, которые описывают не поведение или функции целевой системы, а только дополнительные ограничения. Такие требования редко регламентируют способы достижения их. Например, система должна быть устойчивой к атакам злоумышленников. Чтобы реализовать такое требование, нужен всесторонний анализ и построение модели угроз, процессы, обеспечивающие на всех уровнях организации разработку безопасного кода. Другой пример — это производительность — программная система должна быть отзывчивой для пользователей. Чтобы понять, что значит "отзывчивость", лучше сформулировать такое требование в количественном виде. Например, указать, за какое время должен приходиться отклик от системы, либо какое число пользователей система должна выдерживать одновременно. Чтобы сделать нефункциональные требования ясными и точными, исследователи решают следующие задачи:

- выработка методов повторного использования знаний по нефункциональным требованиям путем создания онтологии, содержащей знания о возможных компромиссах по такому типу требований [38];
- определение требований к безопасности в обобщенном виде [39];
- создание методов детализации и верификации нефункциональных требований [40];
- создание методов конвертации нефункциональных требований в функциональные на основе выявления целей заинтересованных сторон [41];
- рассмотрение специфичных нефункциональных требований, включая требование приватности [42] (важно в социальных сетях) и обратное — требование прозрачности [43];
- рассмотрение мотивации пользователей как нефункционального требования, чтобы пользователи больше и эффективнее проводили время в работе с программной системой [44];
- систематизация нефункциональных требований для качества информации, используемой в системе [45], пример таких требований — хранящая информация о состоянии рынка ценных бумаг должна быть точной и актуальной.

Нефункциональные требования могут содержать в себе высокие риски и неопределенность. Многие такие требования "очевидны" с точки зрения здравого смысла заказчика, но не ясны с точки зрения реализации и могут негативно восприниматься раз-

работчиками. Это вызывает определенный конфликт между заказчиком и исполнителем. Этот конфликт существует при разработке требований для многих типов программных систем. Задача инженера по требованиям — распознать нефункциональные требования, цели и перевести их в параметризованные или функциональные требования. Решение этой задачи призвано уменьшить влияние проблемы коммуникации при разработке требований для промышленно создаваемых программных систем. Несмотря на результаты исследований в области нефункциональных требований, перевод одних типов требований в другие в настоящее время происходит, как правило, в ручном режиме.

Сбор и выявление требований

Классические методы сбора требований допускают, что заказчик осознает, какая программная система ему необходима. На практике видно, что заказчики не всегда открыты, не всегда имеют необходимую техническую квалификацию, не всегда доступны для обсуждения требований и т. д. Для работы с такими заказчиками и построения сложных инновационных систем требуются новые подходы. Это направление исследований будем называть "новые методы сбора и выявления требований":

- применение детективных методов (диаграммы связей — *mind map*), чтобы выявить упущенные требования, когда заказчик целенаправленно скрывает необходимые детали предметной области [46];
- применение техник "этнографии" в сложных организациях — т. е., когда инженер по требованиям находится в офисе заказчика, отслеживая внутренние процессы организации, чтобы самостоятельно выявить требования к информационным системам автоматизации бизнес-процессов [47];
- организация фокус-групп, состоящих из релевантных доменных экспертов, по различным аспектам требований создаваемой программной системы [48];
- применения методов "жидкой демократии" (ликвидной демократии), когда заинтересованное лицо, принимающее решение (ЛПР), недоступно для совместной работы на проекте и участия в утверждении спецификаций требований, тогда такое лицо наделяется правом делегирования принятия решения выделенному профессионалу, представляющему его интересы [49] — такой метод позволяет учесть точку зрения заинтересованного лица, при этом ускорить принятие решений и утверждение требований;
- ролевые игры — использование метафоры "боксерских поединков", когда в команде один участник занимает роль заказчика, а другой исполнителя, и проводится ролевой словесный поединок, чтобы улучшить требования [50];
- использование выделенных аналитиков для анализа состояния предметных областей в параллель к разработке требований [51], т. е. поиск дополнительных источников требований — это особенно актуально, когда система создается взамен существующей и уже существуют технологии, подходы и решения в предметной области;

• подходы к выявлению требований, когда системой должны пользоваться жители городов, регионов — использование социологических опросов [52];

• использование хакатонов для формирования требований и генерации идей [53] — описывается методика, шаги, необходимые ресурсы для организации хакатонов в инженерии требований, а также результаты проведения таких хакатонов.

Исследования [48, 50–52] затрагивают сферы коммуникации и координации. Эти исследования могут частично использоваться как компоненты или улучшения, дающие вклад в разрешение основной проблемы статьи. Например, в исследовании [52] рассматривается идея применения методов социологии для выявления требований к программным системам, которыми должны пользоваться жители городов. Подобные подходы могут быть использованы для проведения автоматизированных опросов среди пользователей и доменных экспертов по неоднозначным сценариям в целях выявления тенденций и поиска наиболее оптимальных решений.

Чем больше пользователей и больше социотехнических аспектов требуется учесть в промышленно создаваемой программной системе, тем больше будет междисциплинарных исследований и адаптации различных социальных, педагогических и менеджерских техник к инженерии требований.

Другие исследования в области инженерии требований

Представленный краткий обзор исследований сфокусирован на самых крупных направлениях (по числу публикаций). Однако в области инженерии требований ведутся исследования и в других направлениях, часть которых отмечены далее:

- управление знаниями и онтологии для упрощения работы с предметной областью (подобно подходам из работы [38]);
- поиск путей формирования требований к устойчивому развитию (модификации) создаваемой программной системы [54];
- методики моделирования в целях уточнения требований [55], в том числе симуляция работы системы до начала разработки, чтобы выявить ограничения;
- поиск новых подходов к более эффективному обучению инженеров по требованиям [56];
- адаптация практики реализации процессов инженерии требований к методологии Agile [57];
- подходы к формированию требований к адаптируемым программным системам, которые адаптируются под условия или пользователя [58];
- аспекты требований, специфичные для конкретных типов программных систем (автомобилей [31], систем искусственного интеллекта, систем безопасности, медицинских систем и т. п.).

Заключение

Разработка требований на ранних этапах жизненного цикла ПО сложнее и менее формализована чем на поздних этапах, потому что на начальной фазе проектирования вводные положения заказчика, как

правило, фрагментированы и плохо структурированы. Это усложняет сферу коммуникации. На фоне этих сложностей и увеличения масштаба создаваемой системы возникает *проблема коммуникации, координации и контроля при разработке требований для промышленно создаваемых программных систем*, т. е. значительное увеличение времени процесса разработки требований в силу роста сложности системы и увеличения числа участников и их вариантов взаимодействий. Обзор показал, что исследователи сталкиваются с этой *проблемой*, решая конкретные теоретические и прикладные задачи инженерии требований [6, 7, 28]. Несмотря на некоторые успехи исследователей в частных задачах [28], не было найдено системного исследования и существующего разрешения данной проблемы.

Чтобы приблизиться к разрешению проблемы или уменьшить влияние от проблемы необходимы дальнейшие исследования. Необходимо выявить и определить следующее: виды и структуры коммуникации различных участников процесса; типовые структуры координации, например, ревью, фокус-группы, мозговые штурмы, комитеты и т. п.; активности, которые дают наибольший вклад в критический путь процесса разработки требований. На основе анализа и проработки проблемы необходимо создать интегрированное средство поддержки коммуникации, координации и автоматизированного контроля, чтобы уменьшить объем коммуникации, и, как следствие, сократить критический путь процесса разработки требований.

Список литературы

1. Халл Э., Джексон К., Джерими Д. Разработка и управление требованиями: практическое руководство пользователя, 2-е изд.: Пер. с англ., 2005. — 240 с.
2. Косяков А., Свит У., Сеймур С., Бимер С. Системная инженерия. Принципы и практика / Пер. с англ. под ред. В. К. Батоврина. — М.: ДМК Пресс, 2014. — 636 с.
3. CMMI for Development. URL: https://resources.sei.cmu.edu/asset_files/TechnicalReport/2010_005_001_15287.pdf
4. ISO/IEC/IEEE 12207:2017 // ISO/IEC/IEEE 12207:2017(en) Systems and software engineering — Software life cycle processes. URL: <https://www.iso.org/obp/ui/#iso:std:iso-iec-ieee:12207:ed-1:vl:en>
5. Леффингуэлл Д., Уидриг Д. Принципы работы с требованиями к программному обеспечению. Унифицированный подход / Пер. с англ. — М: Издательский дом "Вильямс", 2002. — 448 с.
6. Samer R., Atas M., Felfernig A., Stettinger M., Falkner A., Schenner G. Group Decision Support for Requirements Management Processes // Proceedings of the 20th Configuration Workshop. — 2018. — P. 19—24.
7. Motger Q., Borrull R., Palomares C., Marco J. OpenReq-DD: A Requirements Dependency Detection Tool // Joint Proceedings of REFSQ-2019 Workshops, Doctoral Symposium, 2019. URL: http://ceur-ws.org/Vol-2376/NLP4RE19_paper01.pdf
8. Van Der Valk R., Pelliccione P., Lago P., Haldal R., Knauss E., Juul J. Transparency and contracts: Continuous integration and delivery in the automotive ecosystem // ACM/IEEE 40th International Conference on Software Engineering: Software Engineering in Practice. — 2018. — P. 23—32.
9. Lieble G., Tichy M., Knause E., Ljungkrantz O., Stlegibauer G. Organisation and communication problems in automotive requirements engineering // Requirements Engineering. — 2018. — Vol. 23. — P. 145—167.
10. Graicunas V. Relationship in Organization // Papers on the Science of Administration. — New York: Columbia University's Institute of Public Administration, 1937. — P. 181—186.
11. IBM Doors. Обзор Rational Doors. URL: https://www.ibm.com/support/knowledgecenter/ru/SSYQBZ_9.6.1/com.ibm.doors.requirements.doc/topics/c_welcome.html
12. Jama Connect. Product Development Lifecycle Solution | Jama Connect. URL: <https://www.jamasoftware.com/platform/jama-connect/>
13. Shah F., Pfahl D. Evaluation and Improving Software Quality Using Text Analysis Techniques — A Mapping Study // Joint Proceedings of REFSQ-2016 Workshops, Doctoral Symposium, 2016. URL: <http://ceur-ws.org/Vol-1564/paper21.pdf>

14. **Kenney O., Copper M.** Automating Requirements Quality Standard with QVscribe // Joint Proceedings of REFSQ-2020 Workshops, Doctoral Symposium, 2020. URL: <http://ceur-ws.org/Vol-2584/NLP4RE-paper3.pdf>
15. **Ryan M., Wheatcraft L., Zinni R., Dick J., Baksa K.** Guide for Writing Requirements, INCOSE-TP-2010-006-02. URL: <https://connect.incose.org/Pages/Product-Details.aspx?ProductCode=TechGuideWR2019Soft>
16. **Motger M., Palomares C., Marco J.** RESim — Automated Detection of Duplicated Requirements in Software Engineering Projects // Joint Proceedings of REFSQ-2020 Workshops, Doctoral Symposium, 2020. URL: <http://ceur-ws.org/Vol-2584/PT-paper1.pdf>
17. **Çevikol S., Aydemir F.** Detecting Inconsistencies of Natural Language Requirements in Satellite Ground Segment Domain // Joint Proceedings of REFSQ-2019 Workshops, Doctoral Symposium, 2019. URL: http://ceur-ws.org/Vol-2376/NLP4RE19_paper15.pdf
18. **Zichler K., Helke S.** R2BC: Tool-Based Requirements Preparation for Delta Analyses by Conversion into Boilerplates // Proceedings of the Workshops of the Software Engineering Conference, 2019. — P. 45–52.
19. **Unterkalmsteiner M., Yates A.** Expert-sourcing Domain-specific Knowledge: The Case of Synonym Validation // Joint Proceedings of REFSQ-2019 Workshops, Doctoral Symposium, 2019. URL: http://ceur-ws.org/Vol-2376/NLP4RE19_paper08.pdf
20. **Fantechi A., Gnesi S., Semini L.** From Generic Requirements to Variability // Joint Proceedings of REFSQ-2019 Workshops, Doctoral Symposium, 2019. http://ceur-ws.org/Vol-2376/NLP4RE19_paper16.pdf
21. **Krismayer T., Kronberger P., Rabiser R., Grünbacher P.** ReMinds-CMT: An Interactive Tool Supporting Constraint Mining for Requirements Monitoring // Joint Proceedings of REFSQ-2019 Workshops, Doctoral Symposium, 2019. URL: http://ceur-ws.org/Vol-2376/PT_paper_3.pdf
22. **Lohar S.** Supporting Natural Language Queries across the Requirements Engineering Process // Joint Proceedings of REFSQ-2016 Workshops, Doctoral Symposium, 2016. URL: <http://ceur-ws.org/Vol-1564/paper24.pdf>
23. **Khelifa A., Haoues M., Sellami A.** Towards a Software Requirements Change Classification using Support Vector Machine // Proceedings of the second Conference on Language Processing and Knowledge Management, 2018. URL: http://ceur-ws.org/Vol-2279/LPKM2018_paper_1.pdf
24. **Boschen M., Bogusch R., Fraga A., Rudat C.** Bridging the Gap between Natural Language Requirements and Formal Specifications // Joint Proceedings of REFSQ-2016 Workshops, Doctoral Symposium, 2016. URL: https://ret.cs.lth.se/16/downloads/RET_2016_paper_7.pdf
25. **Petersen K., Ali N.** Operationalizing the requirements selection process with study selection procedures from systematic literature reviews // Joint Proceedings of REFSQ-2015 Workshops, 2015. — P. 102–113.
26. **Seyff N., Stade M., Fotrousi F., Glinz M., Guzman E., Kolpondinos-Huber M., Arzapalo D., Oriol M., Schaniel R.** End-user Driven Feedback Prioritization // Joint Proceedings of REFSQ-2017 Workshops, Doctoral Symposium, 2017. URL: <http://ceur-ws.org/Vol-1796/priere-paper-4.pdf>
27. **Merten T., Mager B., Hubner P., Quirchmayr T., Paech B., Bursner S.** Requirements Communication in Issue Tracking Systems in Four Open-Source Projects // Joint Proceedings of REFSQ-2015 Workshops. — 2015. — P. 114–125.
28. **Busetta P.** Addressing Team Awareness By Means Of A Requirement Prioritization Tool // Joint Proceedings of REFSQ-2017 Workshops, Doctoral Symposium, 2017. URL: <http://ceur-ws.org/Vol-1796/priere-paper-3.pdf>
29. **Amin M.** TEBIB Gamifying Requirements Engineering for Better Practice // Joint Proceedings of REFSQ-2019 Workshops, Doctoral Symposium, 2019. URL: <http://ceur-ws.org/Vol-2376/DS-paper3.pdf>
30. **Hubner P.** Quality Improvements for Trace Links between Source Code and Requirements // Joint Proceedings of REFSQ-2016 Workshops, Doctoral Symposium, 2016. URL: <http://ceur-ws.org/Vol-1564/paper29.pdf>
31. **Maro S.** Improving Software Traceability in the Development of Automotive Embedded Systems A Research Abstract // Joint Proceedings of REFSQ-2016 Workshops, Doctoral Symposium, 2016. URL: <http://ceur-ws.org/Vol-1564/paper28.pdf>
32. **Anang Y., Watanabe Y.** Applying Layering Concept to the Software Requirements Analysis and Architectural Design // Joint Proceedings of REFSQ-2016 Workshops, Doctoral Symposium, 2016. — P. 45–50.
33. **Rath M., Akehurst D., Borowski C., Mader P.** Are Graph Query Languages Applicable for Requirements Traceability Analysis? // Joint Proceedings of REFSQ-2017 Workshops, Doctoral Symposium, 2017. URL: <http://ceur-ws.org/Vol-1796/poster-paper-2.pdf>
34. **Anu V., Walia G., Bradshaw G., Hu W., Carver J.** Using Human Error Abstraction Method for Detecting and Classifying Requirements Errors: A Live Study // Joint Proceedings of REFSQ-2017 Workshops, Doctoral Symposium, 2017. URL: <http://ceur-ws.org/Vol-1796/rmt-livestudy.pdf>
35. **Reytérou C.** Requirements quality in the incremental design processes: problems and perspectives. // Joint Proceedings of REFSQ-2016 Workshops, Doctoral Symposium, 2016. URL: <http://ceur-ws.org/Vol-1564/paper4.pdf>
36. **DeVries B., Betty H.** Cheng Using Models at Run Time to Detect Incomplete and Inconsistent Requirements // Proceedings of MODELS 2017 Satellite Event: Workshops, Posters, Doctoral Symposium, 2017. — P. 201–209.
37. **Lenarduzzi V., Fucci D., Mendez D.** On the Perceived Harmfulness of Requirement Smells: An Empirical Study // Joint Proceedings of REFSQ-2020 Workshops, Doctoral Symposium, 2020. URL: <http://ceur-ws.org/Vol-2584/LS-paper2.pdf>
38. **Veleda R., Luiz Cysneiros L.** An Initial Approach to Reuse Non-Functional Requirements Knowledge // Proceedings of the Eighth International i*Workshop — iStar 2015. — P. 25–30.
39. **Sandkuhl K., Matulevičius R., Ahmed N., Kirikova M.** Refining Security Requirement Elicitation from Business Processes using Method Engineering // Joint Proceedings of the BIR 2015 Workshops and Doctoral Consortium, 2015. — P. 98–109.
40. **Gaither D.** Verification of Non-Functional Requirements Using Formal Semantics // Proceedings of MODELS 2017 Satellite Event: Workshops, Posters, Doctoral Symposium, 2017. — P. 503–506.
41. **Haindl P., Plösch R., Körner C.** Tailoring and Evaluating Non-Functional Interests Towards Task-Oriented Functional Requirements // Combined Proceedings of the Workshops at Software Engineering 2020. URL: <http://ceur-ws.org/Vol-2581/aesp2020paper1.pdf>
42. **Wang J.** Privacy Requirement Modeling and Verification in Cloud Computing // Proceedings of the Doctoral Symposium at the 18th ACM/IEEE International Conference of Model-Driven Engineering Languages and Systems 2015. URL: <http://ceur-ws.org/Vol-1531/paper3.pdf>
43. **Hosseini M., Shahri A., Phalp K., Ali R.** Transparency as a Requirement // Joint Proceedings of REFSQ-2015 Workshops, 2015. — P. 222–223.
44. **Shahri A., Hosseini M., Phalp K., Ali R.** Motivation as a Supplementary Requirement // Joint Proceedings of REFSQ-2015 Workshops, 2015. — P. 220–221.
45. **Gharib M.** Dealing with Non-functional Requirements, the Case of Information Quality Requirements: Experience Report // Thirteenth International iStar Workshop, 2020. — P. 25–30.
46. **Hess A., Mennig P., Bartels N.** Conspiracy Walls in Requirements Engineering — Analyzing Requirements like a Detective // Joint Proceedings of REFSQ-2020 Workshops, Doctoral Symposium, 2020. URL: <http://ceur-ws.org/Vol-2584/CreARE-paper1.pdf>
47. **Desnica N., Regev G., Wegmann A.** Expanding Requirements Through Observation: An Experience Report // Proceedings of the 3rd International Workshop on Socio-Technical Perspective in IS development (STPIS'17), 2017. — P. 14–24.
48. **Daneva M.** Focus Groups: Cost-effective and Methodologically Sound Ways to Get Practitioners Involved in Your Empirical RE Research // Joint Proceedings of REFSQ-2015 Workshops, 2015. — P. 211–216.
49. **Seesink J., Hoppenbrouwers S.** Using a Liquid Democracy Tool for End-user Involvement in Continuous RE // Joint Proceedings of REFSQ-2017 Workshops, Doctoral Symposium, 2017. URL: <http://ceur-ws.org/Vol-1796/cre-paper-5.pdf>
50. **Oemig C.** The Box Fight Analogy: A Blueprint for Pre-Sales Requirements Engineering Workshops // Joint Proceedings of REFSQ-2015 Workshops, 2015. — P. 71–76.
51. **Kirikova M., Purmalietis K.** Analytics in Continuous Requirements Engineering // Joint Proceedings of REFSQ-2017 Workshops, Doctoral Symposium, 2017. URL: <http://ceur-ws.org/Vol-1796/cre-paper-1.pdf>
52. **Polst S., Elberzhager F.** Eliciting Requirements from Citizens — What Can We Learn from Other Disciplines? // Joint Proceedings of REFSQ-2020 Workshops, Doctoral Symposium, 2020. URL: <http://ceur-ws.org/Vol-1796/cre-paper-1.pdf>
53. **Mennig P., Elberzhager F.** How to Gather Requirements from the Crowd with Hackathons — Challenges, Experiences and Opportunities // Joint Proceedings of REFSQ-2020 Workshops, Doctoral Symposium, 2020. URL: <http://ceur-ws.org/Vol-2584/PT-paper6.pdf>
54. **Fernandez N., Lago P., Calero C.** How do Quality Requirements Contribute to Software Sustainability? // Joint Proceedings of REFSQ-2016 Workshops, Doctoral Symposium, 2016. — P. 7–10.
55. **Aiello F., Garro A., Lemmens Y., Dutré S.** Formal modeling of system properties for simulation-based verification of requirements: lessons learned // Proceedings of the 3rd INCOSE Italia Conference on Systems Engineering, 2017. — P. 54–61.
56. **Daun M., Tenbergen B.** Teaching Requirements Engineering with Industry Case Examples // Tagungsband des 17. Workshops "Software Engineering im Unterricht der Hochschulen", 2020. — P. 49–50.
57. **Nurdiani I.** Managing Requirements Interdependencies in Agile Software Development: A Preliminary Result // Joint Proceedings of REFSQ-2016 Workshops, Doctoral Symposium, 2016. URL: <http://ceur-ws.org/Vol-1564/paper36.pdf>
58. **Kneer F., Kamsties E.** A Framework for Prototyping and Evaluating Self-adaptive Systems — A Research Preview // Joint Proceedings of REFSQ-2016 Workshops, Doctoral Symposium, 2016. URL: <http://ceur-ws.org/Vol-1564/paper13.pdf>

Requirements Engineering for Software Systems: a Systematic Literature Review

A. A. Kodubets, alexey@kodubets.ru, I. L. Artemieva, iartemeva@mail.ru, Far Eastern Federal University, Vladivostok, 690922, Russian Federation

Corresponding author:

Kodubets Alexey A., Postgraduate Student, Far Eastern Federal University, Vladivostok, 690922, Russian Federation
E-mail: alexey@kodubets.ru

Received on March 24, 2021

Accepted on July 30, 2021

This article contains a systematic literature review of requirements engineering for software systems. The literature published within last 5 years was included into the review. A research question was defined as requirements development process of large scale software system (with thousands of requirements) and an interaction problem during this process (communication, coordination and control). The problem is caused by the fact that large-scale software system requirements process is a cross-disciplinary task and it involves multiple parties — stakeholders, domain experts, and suppliers with own goals and constrains, and thus, the interaction between them seriously slows down the overall requirements development process than writing the requirements specification itself. The research papers were classified by several research directions: Natural Language Processing for Requirements Engineering (NLP4RE), Requirement Prioritization, Requirements Traceability, Quality of Software Requirements, Non-functional Requirements and Requirements Elicitation. Motivation and intensity of each direction was described. Each direction was structured and represented with the key references. A contribution of each research direction into the research question was analyzed and summarized including potential further steps. It was identified that some researchers had met a part of the described problem in different forms during their researches. At the end, other researches were described additionally in a short overview. To approach the research question further potential direction was described.

Keywords: requirements engineering, software requirements specification, large scale software systems

For citation:

Kodubets A. A., Artemieva I. L. Requirements Engineering for Software Systems: A Systematic Literature Review, *Programmnyaya Ingeneriya*, 2021, vol. 12, no. 7, pp. 339—349.

DOI: 10.17587/prin.12.339-349

References

1. Hull E., Jackson K., Dick J. *Requirements Engineering. Second Edition*, Springer, 2005, 198 p.
2. Kossiakoff A., Sweet W. N., Seymour S. J., Bierner S. M. *System Engineering Principles and Practice*, 2nd Edition, Wiley-Interscience, 2011, 560 p.
3. Capability Maturity Model Integration for Development, available at: https://resources.sei.cmu.edu/asset_files/TechnicalReport/2010_005_001_15287.pdf
4. SO/IEC/IEEE 12207:2017 Systems and software engineering — Software life cycle processes, 2017, available at: <https://www.iso.org/obp/ui/#iso:std:iso-iec-ieee:12207:ed-1:v1:en>
5. Leffingwell D., Widrig D. *Managing Software requirements: a use case approach*, Addison-Wesley, Boston, 2003, 800 p.
6. Samer R., Atas M., Felfernig A., Stettinger M., Falkner A., Schenner G. Group Decision Support for Requirements Management Processes, *Proceedings of the 20th Configuration Workshop*, 2018, pp. 19—24.
7. Motger Q., Borrull R., Palomares C., Marco J. OpenReq-DD: A Requirements Dependency Detection Tool, *Joint Proceedings of REFSQ-2019 Workshops, Doctoral Symposium*, 2019, available at: http://ceur-ws.org/Vol-2376/NLP4RE19_paper01.pdf
8. Van Der Valk R., Pelliccione P., Lago P., Heldal R., Knauss E., Juul J. Transparency and contracts: Continuous integration and delivery in the automotive ecosystem, *ACM/IEEE 40th International Conference on Software Engineering: Software Engineering in Practice*, 2018, pp. 23—32.
9. Lieble G., Tichy M., Knause E., Ljungkrantz O., Stlegibauer G. Organisation and communication problems in automotive requirements engineering, *Requirements Engineering*, 2018, vol. 23, pp. 145—167.
10. Graicunas V. Relationship in Organization, *Papers on the Science of Administration*, New York, Columbia University's Institute of Public Administration, 1937, pp. 181—186.
11. IBM Rational DOORS, available at: https://www.ibm.com/support/knowledgecenter/ru/SSYQBZ_9.6.1/com.ibm.doors.requirements.doc/topics/c_welcome.html
12. "Jama Connect" — Product Development Lifecycle Solution, available at: <https://www.jamasoftware.com/platform/jama-connect/>
13. Shah F., Pfahl D. Evaluation and Improving Software Quality Using Text Analysis Techniques — A Mapping Study, *Joint Proceedings of REFSQ-2016 Workshops, Doctoral Symposium, Research Method Track, and Poster Track (REFSQ-JP 2016)*, 2016, available at: <http://ceur-ws.org/Vol-1564/paper21.pdf>
14. Kenney O., Copper M. Automating Requirements Quality Standard with QVscribe, *Joint Proceedings of REFSQ-2020 Workshops, Doctoral Symposium*, 2020, available at: <http://ceur-ws.org/Vol-2584/NLP4RE-paper3.pdf>
15. Ryan M., Wheatcraft L., Zinni R., Dick J., Baksa K. Guide for Writing Requirements, *INCOSE Publications Office*, 2019, available at: <https://connect.incose.org/Pages/Product-Details.aspx?ProductCode=TechGuideWR2019Soft>
16. Motger Q., Palomares C., Marco J. RESim — Automated Detection of Duplicated Requirements in Software Engineering Projects, *Joint Proceedings of REFSQ-2020 Workshops, Doctoral Symposium*, 2020, available at: <http://ceur-ws.org/Vol-2584/PT-paper1.pdf>
17. Çevikol S., Aydemir F. B. Detecting Inconsistencies of Natural Language Requirements in Satellite Ground Segment Domain, *Joint Proceedings of REFSQ-2019 Workshops, Doctoral Symposium*, 2019, available at: http://ceur-ws.org/Vol-2376/NLP4RE19_paper15.pdf

18. **Zichler K., Helke S.** R2BC: Tool-Based Requirements Preparation for Delta Analyses by Conversion into Boilerplates, *Proceedings of the Workshops of the Software Engineering Conference 2019*, 2019, pp. 45–52.
19. **Unterkalmsteiner M., Yates A.** Expert-sourcing Domain-specific Knowledge: The Case of Synonym Validation, *Joint Proceedings of REFSQ-2019 Workshops, Doctoral Symposium*, 2019, available at: http://ceur-ws.org/Vol-2376/NLP4RE19_paper08.pdf
20. **Fantechi A., Gnesi S., Semini L.** From Generic Requirements to Variability, *Joint Proceedings of REFSQ-2019 Workshops, Doctoral Symposium*, 2019, available at: http://ceur-ws.org/Vol-2376/NLP4RE19_paper16.pdf
21. **Krismayer T., Kronberger P., Rabiser R., Grünbacher P.** ReMinds-CMT: An Interactive Tool Supporting Constraint Mining for Requirements Monitoring, *Joint Proceedings of REFSQ-2019 Workshops, Doctoral Symposium*, 2019, available at: http://ceur-ws.org/Vol-2376/PT_paper_3.pdf
22. **Lohar S.** Supporting Natural Language Queries across the Requirements Engineering Process, *Joint Proceedings of REFSQ-2016 Workshops, Doctoral Symposium*, 2016, available at: <http://ceur-ws.org/Vol-1564/paper24.pdf>
23. **Khelifa A., Haoues M., Sellami A.** Towards a Software Requirements Change Classification using Support Vector Machine, *Proceedings of the second Conference on Language Processing and Knowledge Management*, 2018, available at: http://ceur-ws.org/Vol-2279/LPKM2018_paper_1.pdf
24. **Boschen M., Bogusch R., Fraga A., Rudat C.** Bridging the Gap between Natural Language Requirements and Formal Specifications, *Joint Proceedings of REFSQ-2016 Workshops, Doctoral Symposium*, 2016, available at: https://ret.cs.lth.se/16/downloads/RET_2016_paper_7.pdf
25. **Petersen K., Ali N. B.** Operationalizing the requirements selection process with study selection procedures from systematic literature reviews, *Joint Proceedings of REFSQ-2015 Workshops*, 2015, pp. 102–113.
26. **Seyff N., Stade M., Fotrousi F., Glinz M., Guzman E., Kolpindinos-Huber M., Arzapalo D. M., Oriol M., Schaniel R.** End-user Driven Feedback Prioritization, *Joint Proceedings of REFSQ-2017 Workshops, Doctoral Symposium*, 2017, available at: <http://ceur-ws.org/Vol-1796/priere-paper-4.pdf>
27. **Merten T., Mager B., Hubner P., Quirschmayr T., Paech B., Bursner S.** Requirements Communication in Issue Tracking Systems in Four Open-Source Projects, *Joint Proceedings of REFSQ-2015 Workshops*, 2015, pp. 114–125.
28. **Busetta P.** Addressing Team Awareness By Means Of A Requirement Prioritization Tool, *Joint Proceedings of REFSQ-2017 Workshops, Doctoral Symposium*, 2017, available at: <http://ceur-ws.org/Vol-1796/priere-paper-3.pdf>, pp. 1–4.
29. **Amin M. E.** TEBIB Gamifying Requirements Engineering for Better Practice, *Joint Proceedings of REFSQ-2019 Workshops, Doctoral Symposium*, 2019, available at: <http://ceur-ws.org/Vol-2376/DS-paper3.pdf>
30. **Hubner P.** Quality Improvements for Trace Links between Source Code and Requirements, *Joint Proceedings of REFSQ-2016 Workshops, Doctoral Symposium*, 2016, available at: <http://ceur-ws.org/Vol-1564/paper29.pdf>
31. **Maro S.** Improving Software Traceability in the Development of Automotive Embedded Systems A Research Abstract, *Joint Proceedings of REFSQ-2016 Workshops, Doctoral Symposium*, 2016, available at: <http://ceur-ws.org/Vol-1564/paper28.pdf>
32. **Anang Y., Watanabe Y.** Applying Layering Concept to the Software Requirements Analysis and Architectural Design, *Joint Proceedings of REFSQ-2016 Workshops, Doctoral Symposium*, 2016, pp. 45–50.
33. **Rath M., Akehurst D., Borowski C., Mader P.** Are Graph Query Languages Applicable for Requirements Traceability Analysis?, *Joint Proceedings of REFSQ-2017 Workshops, Doctoral Symposium*, 2017, available at: <http://ceur-ws.org/Vol-1796/poster-paper-2.pdf>
34. **Anu V., Walia G., Bradshaw G., Hu W., Jeffrey C.** Carver Using Human Error Abstraction Method for Detecting and Classifying Requirements Errors: A Live Study, *Joint Proceedings of REFSQ-2017 Workshops, Doctoral Symposium*, 2017, available at: <http://ceur-ws.org/Vol-1796/rmt-livestudy.pdf>
35. **Reytérou C.** Requirements quality in the incremental design processes: problems and perspectives, *Joint Proceedings of REFSQ-2016 Workshops, Doctoral Symposium*, 2016, available at: <http://ceur-ws.org/Vol-1564/paper4.pdf>
36. **DeVries B., Betty H. C.** Cheng Using Models at Run Time to Detect Incomplete and Inconsistent Requirements, *Proceedings of MODELS 2017 Satellite Event: Workshops, Posters, Doctoral Symposium*, 2017, pp. 201–209.
37. **Lenarduzzi V., Fucci D., Mendez D.** On the Perceived Harmfulness of Requirement Smells: An Empirical Study, *Joint Proceedings of REFSQ-2020 Workshops, Doctoral Symposium*, 2020, available at: <http://ceur-ws.org/Vol-2584/LS-paper2.pdf>
38. **Veleda R., Cysneiros L. M.** An Initial Approach to Reuse Non-Functional Requirements Knowledge, *Proceedings of the Eighth International i*Workshop — iStar 2015*, 2015, pp. 25–30.
39. **Sandkuhl K., Matulevičius R., Ahmed N., Kirikova M.** Refining Security Requirement Elicitation from Business Processes using Method Engineering, *Joint Proceedings of the BIR 2015 Workshops and Doctoral Consortium*, 2015, pp. 98–109.
40. **Gaither D.** Verification of Non-Functional Requirements Using Formal Semantics, *Proceedings of MODELS 2017 Satellite Event: Workshops, Posters, Doctoral Symposium*, 2017, pp. 503–506.
41. **Haindl Ph., Plösch R., Körner C.** Tailoring and Evaluating Non-Functional Interests Towards Task-Oriented Functional Requirements, *Combined Proceedings of the Workshops at Software Engineering 2020*, available at: <http://ceur-ws.org/Vol-2581/aesp2020paper1.pdf>
42. **Wang J.** Privacy Requirement Modeling and Verification in Cloud Computing, *Proceedings of the Doctoral Symposium at the 18th ACM/IEEE International Conference of Model-Driven Engineering Languages and Systems 2015 (MoDELS 2015)*, available at: <http://ceur-ws.org/Vol-1531/paper3.pdf>
43. **Hosseini M., Shahri A., Phalp K., Ali A.** Transparency as a Requirement, *Joint Proceedings of REFSQ-2015 Workshops*, 2015, pp. 222–223.
44. **Shahri A., Hosseini M., Phalp K., Ali A.** Motivation as a Supplementary Requirement, *Joint Proceedings of REFSQ-2015 Workshops*, pp. 220–221.
45. **Gharib M.** Dealing with Non-functional Requirements, the Case of Information Quality Requirements: Experience Report, *Thirteenth International iStar Workshop*, 2020, pp. 25–30.
46. **Hess A., Mennig P., Bartels N.** Conspiracy Walls in Requirements Engineering — Analyzing Requirements like a Detective, *Joint Proceedings of REFSQ-2020 Workshops, Doctoral Symposium*, 2020, available at: <http://ceur-ws.org/Vol-2584/Creare-paper1.pdf>, pp. 1–3.
47. **Desnica N., Regev G., Wegmann A.** Expanding Requirements Through Observation: An Experience Report, *Proceedings of the 3rd International Workshop on Socio-Technical Perspective in IS development (STPIS'17)*, 2017, pp. 14–24.
48. **Daneva M.** Focus Groups: Cost-effective and Methodologically Sound Ways to Get Practitioners Involved in Your Empirical RE Research, *Joint Proceedings of REFSQ-2015 Workshops*, 2015, pp. 211–216.
49. **Seesink J., Hoppenbrouwers S.** Using a Liquid Democracy Tool for End-user Involvement in Continuous RE, *Joint Proceedings of REFSQ-2017 Workshops, Doctoral Symposium*, 2017, available at: <http://ceur-ws.org/Vol-1796/cre-paper-5.pdf>
50. **Oemig C.** The Box Fight Analogy: A Blueprint for Pre-Sales Requirements Engineering Workshops, *Joint Proceedings of REFSQ-2015 Workshops*, 2015, pp. 71–76.
51. **Kirikova M., Purmalietis K.** Analytics in Continuous Requirements Engineering, *Joint Proceedings of REFSQ-2017 Workshops, Doctoral Symposium*, 2017, available at: <http://ceur-ws.org/Vol-1796/cre-paper-1.pdf>
52. **Polst S., Elberzhager F.** Eliciting Requirements from Citizens — What Can We Learn from Other Disciplines? *Joint Proceedings of REFSQ-2020 Workshops, Doctoral Symposium*, 2020, available at: <http://ceur-ws.org/Vol-2584/PT-paper7.pdf>
53. **Mennig P., Elberzhager F.** How to Gather Requirements from the Crowd with Hackathons — Challenges, Experiences and Opportunities, *Joint Proceedings of REFSQ-2020 Workshops, Doctoral Symposium*, 2020, available at: <http://ceur-ws.org/Vol-2584/PT-paper6.pdf>
54. **Fernandez N. C., Lago P., Calero C.** How do Quality Requirements Contribute to Software Sustainability? *Joint Proceedings of REFSQ-2016 Workshops, Doctoral Symposium*, 2016, pp. 7–10.
55. **Aiello F., Garro A., Lemmens Y., Dutré S.** Formal modeling of system properties for simulation-based verification of requirements: lessons learned, *Proceedings of the 3rd INCOSE Italia Conference on Systems Engineering*, 2017, pp. 54–61.
56. **Daun M., Tenbergen B.** Teaching Requirements Engineering with Industry Case Examples, *Tagungsband des 17. Workshops "Software Engineering im Unterricht der Hochschulen"*, 2020, pp. 49–50.
57. **Nurdiani I.** Managing Requirements Interdependencies in Agile Software Development: A Preliminary Result, *Joint Proceedings of REFSQ-2016 Workshops, Doctoral Symposium*, 2016, available at: <http://ceur-ws.org/Vol-1564/paper36.pdf>
58. **Kneer F., Kamsties E.** A Framework for Prototyping and Evaluating Self-adaptive Systems — A Research Preview, *Joint Proceedings of REFSQ-2016 Workshops, Doctoral Symposium*, 2016, available at: <http://ceur-ws.org/Vol-1564/paper13.pdf>

С. С. Шевелев, канд. техн. наук, доц. кафедры, schewelew@mail.ru, Юго-Западный государственный университет, Курск

Устройство выполнения логических и арифметических операций

Разработано устройство, выполняющее логические и арифметические операции, которое может использоваться для создания высокопроизводительных, быстродействующих вычислительных систем. Специализированные блоки выполняют логические операции И, ИЛИ, НЕ, а также арифметические операции сложения и вычитания двоичных чисел. Арифметические операции выполняются в прямых кодах с фиксированной запятой. Представлены структурная схема устройства, структурно-функциональные схемы блоков и алгоритм работы устройства.

Ключевые слова: двоичная система счисления, логические элементы, сумматор-вычитатель, перенос, заем, сумма, разность

Введение

Вычислительную открытую развиваемую асинхронную модульную систему можно представить как иерархию универсальной ЭВМ, систему специализированных вычислительных модулей и гибкую подсистему интерфейсных каналов параллельной передачи информации между устройствами. Реконфигурация вычислительной системы обеспечивается изменением числа вычислительных модулей, их структурами и составами, связями каналов различных типов. Вычислительные системы, основанные на принципах модульности и близкодействия, удовлетворяют требованиям асинхронности, децентрализованности и распределенности. Способность системы к динамической реконфигурации приведет к ее высокой универсальности, при этом достигается заданный уровень производительности [1].

В предложенной вычислительной открытой асинхронной системе применяются неоднородные модули, в которых явно выражена функциональная специализация устройств. Универсальная вычислительная машина, реализующая любой алгоритм, не может конкурировать по скорости обработки со специализированными вычислительными модулями при решении определенных задач. Вычислительная открытая развиваемая асинхронная модульная система имеет параллельную структуру за счет использования интерфейсных каналов соединения вычислительных модулей, характеризуется асинхронностью работы составляющих ее устройств, имеет упорядоченную спецификацию модулей и интерфейсов [2].

Основным принципом развития в архитектуре вычислительных систем является распараллеливание и совмещение процессов разных типов. Одновременно с этим увеличивается число решающих модулей. В связи с этим возникает проблема организации коммутации и обменов между процессором и устройствами, а также функциональное соедине-

ние между вычислительными модулями. В устройстве отдельный специализированный модуль становится самостоятельным, функционально полным вычислительным микропроцессором, так как имеет собственные средства обработки, оперативное запоминающее устройство для хранения данных, локальную коммутацию и блок управления. Вычислительные специализированные процессоры не менее чем на порядок дешевле равномоного универсального устройства [3].

Специализированные модули эффективно решают задачи по ускоренному выполнению арифметических и логических операций. Поразрядное выполнение логических операций применимо к каждой паре битов, которые стоят на одинаковых позициях в двоичном представлении чисел. Эти операции применяются при сравнении чисел, сложении и вычитании. Реализация поразрядных операций выполняется в арифметико-логическом устройстве процессора. Сдвиги влево и вправо используются при умножении и целочисленном делении на два и для выделения двух разрядов множителя при выполнении ускоренного умножения.

Арифметико-символьный процессор вычислительной системы выполняет основные арифметические операции сложения, вычитания, умножения и деления чисел в прямых кодах с фиксированной и плавающей запятой, осуществляет поисковые функции и операции замены, а также выполняет логические операции: инверсию, конъюнкцию, дизъюнкцию и исключающее ИЛИ.

В блоке выполнения логических операций последовательность выполнения булевых функций определяется установкой управляющих сигналов на входах электронных ключей в единичное состояние, которые отпирают соответствующие ключи и через их значения переменных и ранее полученные результаты других блоков поступают на входы очередного блока устройства.

Устройство выполнения логических операций содержит систему электронных ключей, которая служит для передачи переменных и управляющих сигналов из блока управления: блоки конъюнкторов, дизъюнкторов, операции исключающее ИЛИ, инверторов. Эти блоки выполняют следующие логические операции: конъюнкцию, дизъюнкцию, исключающее ИЛИ, инверсию. Электронные ключи конъюнкторов, дизъюнкторов, операции исключающее ИЛИ, инверторов выполняют функцию по передаче переменных в блоки устройства, блок хранения результатов служит для записи и хранения в нем результатов выполнения логических операций, блок управления логическими операциями служит для генерации управляющих сигналов устройства. Функционально полная система булевых функций может быть определена как совокупность булевых функций f_1, f_2, \dots, f_n , таких, что произвольная булева функция может быть записана в виде формы через функции этой совокупности. Любая система логических функций, через функции которой можно выразить конъюнкцию, дизъюнкцию и отрицание, будет полной.

Функциональная полнота базисов показывает, что произвольная логическая сеть может быть построена из простых функциональных элементов, например, стрелки Пирса или штриха Шеффера.

Свойство полноты функции позволяет промышленности массово выпускать ограниченный набор логических элементов, из которых можно строить любые логические схемы [4].

Специализированные модули имеют ориентацию на ускоренное выполнение определенных функций, что позволяет резко увеличить эффективную производительность при решении только определенных задач. Среди специализированных микропроцессоров можно выделить различные микроконтроллеры, ориентированные на выполнение сложных последовательностей логических операций. Математические микропроцессоры предназначены для повышения производительности при выполнении арифметических операций. С помощью специализированных модулей можно эффективно решать новые сложные задачи параллельной обработки данных. Например, они позволяют осуществить более сложную математическую обработку сигналов, чем широко используемые методы корреляции, также они дают возможность в реальном масштабе времени находить соответствие для сигналов изменяющейся формы путем сравнения их с различными эталонными сигналами для эффективного выделения полезного сигнала на фоне шума.

Один из наиболее эффективных подходов увеличения производительности вычислительных систем состоит в увеличении числа дополнительных специализированных процессоров или специализированных интегральных схем, предназначенных для выполнения отдельных затратных и часто требующихся операций, если такие операции эффективнее выполнять аппаратно.

Более эффективным является использование специальных аппаратных средств, которые называются специализированными сопроцессорами, выполняющими арифметические операции над числами

с плавающей запятой. Математический сопроцессор используется для ускорения выполнения операций над двоичными числами с плавающей запятой, над двоично-кодированными десятичными числами, для вычисления тригонометрических функций. Математический сопроцессор имеет свою систему команд и работает параллельно с основным микропроцессором, но под его управлением. В результате происходит ускорение выполнения операций в десятки раз.

Для повышения производительности вычислительных устройств используются новые архитектурные решения. Наиболее распространенным решением является использование программируемых логических интегральных схем, которые обладают большим вычислительным потенциалом [5].

Для повышения производительности вычислительного устройства необходимо обеспечить распараллеливание вычислительных процессов. В структуре блочного арифметико-логического устройства вводят специальные блоки, что позволяет вести параллельно процесс обработки информации. В предложенном устройстве эффективная производительность увеличивается при выполнении последовательностей логических операций вследствие увеличения числа блоков, которые обрабатывают информацию в параллельном режиме. Арифметические операции выполняются сумматором-вычитателем в прямом коде с меньшим числом вентилях по сравнению с аналогами.

Структура устройства выполнения логических и арифметических операций

Устройство поразрядного выполнения логических и арифметических операций выполняет следующие логические поразрядные операции: конъюнкцию, дизъюнкцию, исключающее ИЛИ, эквиваленцию, инверсию двоичных чисел, арифметические операции суммирование и вычитание (рис. 1) [6].

Блок ввода чисел содержит шифратор ШФ, сумматоры по модулю два. Этот блок позволяет вводить двоичные числа. С выхода шифратора формируется первое двоичное число А и его знаковый разряд Z_nPA , второе двоичное число В и его знаковый разряд Z_nPB и код операции КОП суммирование или вычитание. Знаковые разряды чисел и код операции с выхода шифратора поступают на входы сумматоров по модулю два. Сигнал суммирования-вычитания СУМ-ВЫЧ формируется на выходе сумматоров по модулю два, которые выполнены на булевых элементах. Выходной сигнал вычисляется по формуле

$$\text{СУМ} - \text{ВЫЧ} = (Z_nPB \oplus \text{КОП}) \oplus Z_nPA. \quad (1)$$

Выходными сигналами блока ввода чисел являются двоичные коды операндов ДКЧ, представленные в прямых кодах, и признак операции суммирование-вычитание СУМ-ВЫЧ.

Блок регистра первого числа содержит двоичные триггеры. Этот блок предназначен для хранения двоичного кода первого числа. Входным информационным сигналом блока является сигнал управления СУ. Параллельно на все входы триггеров поступают

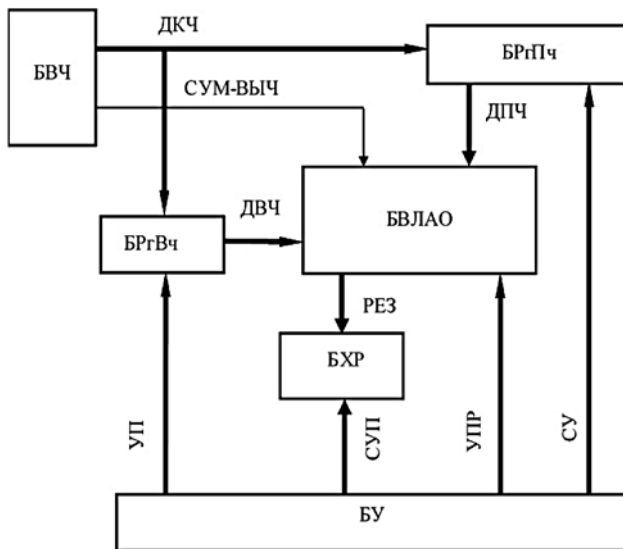


Рис. 1. Структурная схема устройства:

БВЧ — блок ввода чисел; БРПЧ — блок регистра первого числа; БВЛАО — блок выполнения логических и арифметических операций; БРГВЧ — блок регистра второго числа; БХР — блок хранения результатов; БУ — блок управления

управляющие сигналы синхронизации, установки в "0", записи и выдачи результата. Перед началом работы сумматора-вычитателя происходит обнуление всех триггеров блока.

Блок выполнения логических и арифметических операций состоит из блоков конъюнкции БКОН, дизъюнкции БДИЗ, инверсии первого числа БИН А, инверсии второго числа БИН В, операции суммы по модулю два БСумМ 2, эквиваленции БЭКВ, арифметических операций суммирования и вычитания БСУМ-ВЫЧ (рис. 2). Входными информационными сигналами блока являются двоичные разряды первого и второго чисел, которые поступают параллельно на входы регистров блоков.

Входной информационный сигнал управления УПР поступает с выхода блока управления. В структуру этого сигнала входят управляющие сигналы выполнения логических операций: конъюнкции УпКН, дизъюнкции УпДЗ, инверсии первого числа УпИН А, инверсии второго числа УпИН В, суммы по модулю два УпСум 2, эквиваленции УпЭКВ, выполнения арифметических операций суммирования и вычитания УпСУМ-ВЫЧ. Выходным информационным сигналом является сигнал результата РЕЗ выполнения логических и арифметических операций блоками. Структурно информационный сигнал РЕЗ состоит из результатов выполнения логических операций конъюнкции РезКОН, дизъюнкции РезДИЗ, инверсии первого числа РезНЕ А, инверсии второго числа РезНЕ В, суммы по модулю два РезСумМ 2, эквиваленции РезЭКВ, арифметической операции суммирования и вычитания РезСУМ-ВЫЧ. Сигнал результата РЕЗ блока поступает на вход блока хранения результатов БХР.

Блок выполнения логической операции конъюнкции БКОН состоит из логических схем И. Входными информационными сигналами блока являются дво-

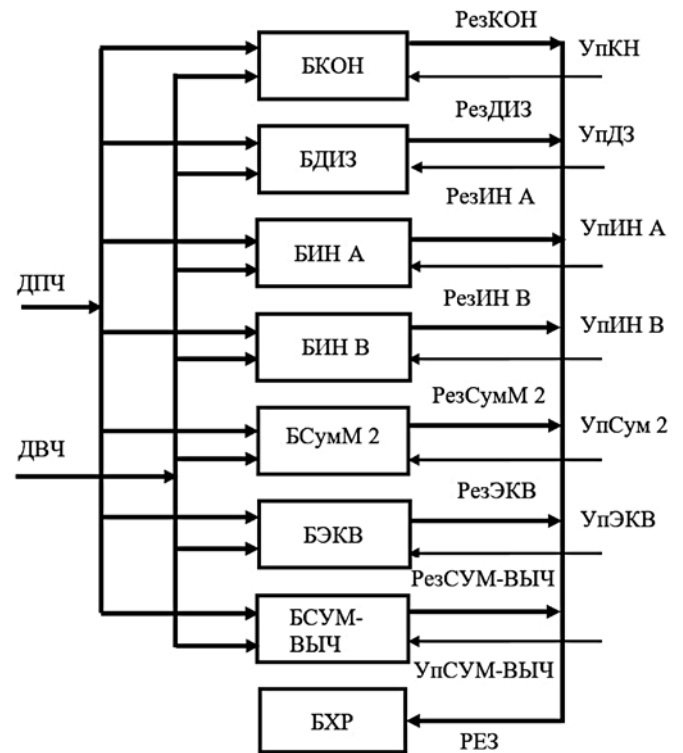


Рис. 2. Структурная схема блока выполнения логических и арифметических операций

ичные разряды первого и второго чисел. Схемы И выполняют функции электронных ключей. Двоичные переменные a_n и b_n поступают на первые входы логических элементов И.

Управляющим сигналом для логических элементов блока является сигнал управления выполнения логической операции конъюнкции УпКН, который поступает на вторые управляющие входы логических элементов, выполняющих функции электронных ключей. Если управляющий сигнал равен единице, то электронные ключи будут открыты и входные двоичные переменные a_n , b_n поступают на входы логических схем И. На выходе этих элементов формируются результаты выполнения логической операции И над входными переменными. На выходе логического элемента И определяется сигнал первого результата конъюнкции РКН1 над переменными a_1 и b_1 , на выходе логического элемента И определяется сигнал результата конъюнкции РКНn над переменными a_n и b_n . Если управляющий сигнал равен нулю, то электронные ключи будут закрыты, логическая операция И не выполняется. Выходной информационный сигнал РезКОН является результатом выполнения логической операции И блоком конъюнкции БКОН.

Блок выполнения логической операции дизъюнкции БДИЗ состоит из логических схем И, ИЛИ. Входными информационными сигналами блока являются двоичные разряды первого и второго чисел. Логические элементы И выполняют функции электронных ключей, на первые входы которых поступают двоичные переменные a_n и b_n . Управляющим сигналом для элементов является сигнал управления выполнения логической операции дизъюнкции УпДЗ,

который поступает на вторые управляющие входы электронных ключей.

Если управляющий сигнал УпДЗ равен единице, то электронные ключи будут открыты и входные двоичные переменные a_n, b_n поступают на входы логических элементов ИЛИ. На выходе этих элементов формируются результаты выполнения логической операции дизъюнкции над входными переменными a_n и b_n . Если управляющий сигнал УпДЗ равен нулю, то электронные ключи будут закрыты, логическая операция ИЛИ на выходе блока выполняться не будет. Выходной информационный сигнал РезДИЗ является результатом выполнения логической операции ИЛИ блоком дизъюнкции БДИЗ.

Блок выполнения логической операции инверсии первого числа БИН А состоит из логических элементов И и схем НЕ. Входным информационным сигналом блока являются двоичные разряды первого числа ДПЧ. Логические элементы И выполняют функции электронных ключей. Двоичные переменные a_1, \dots, a_n поступают на первые входы логических элементов И, управляющим сигналом для которых является сигнал управления выполнением логической операции инверсии первого числа УпИН А. Этот сигнал поступает на вторые управляющие входы элементов И. Если управляющий сигнал УпИН А равен единице, то электронные ключи будут открыты и входные двоичные переменные a_1, \dots, a_n поступают на входы схем НЕ, которые выполняют логическую операцию инверсии входных переменных, инвертируют двоичные разряды. На выходе этих элементов формируются результаты выполнения логической операции инверсии входных переменных $\bar{a}_1, \dots, \bar{a}_n$. Если управляющий сигнал УпИН А равен нулю, то электронные ключи будут закрыты, логическая операция инверсии первого числа на выходе блока выполняться не будет. Выходной информационный сигнал РезИН А является результатом выполнения логической операции инверсии блоком инверсии первого числа БИН А.

Блок выполнения логической операции инверсии второго числа БИН В состоит из логических элементов И и схем НЕ. Входным информационным сигналам блока являются двоичные разряды второго числа ДВЧ. Логические элементы И выполняют функции электронных ключей. Двоичные переменные b_1, \dots, b_n поступают на первые входы элементов И, управляющим сигналом которых является сигнал управления выполнением логической операции инверсии второго числа УпИН В, который поступает на вторые управляющие входы элементов И, выполняющих функции электронных ключей. Если управляющий сигнал УпИН В равен единице, то электронные ключи будут открыты и входные двоичные переменные b_1, \dots, b_n поступают на входы логических элементов НЕ, которые выполняют логическую операцию инверсии входных переменных, инвертируют двоичные разряды. На выходе этих элементов формируются результаты выполнения логической операции инверсии входных переменных $\bar{b}_1, \dots, \bar{b}_n$. Если управляющий сигнал УпИН В равен нулю, то электронные ключи будут закрыты, логическая операция инверсии второго числа на выходе блока выполняться не будет. Выходной информационный сигнал РезИН В явля-

ется результатом выполнения логической операции инверсии.

Блок выполнения логической операции суммы по модулю два БСумМ построен на логических элементах. Входными информационными сигналами блока являются двоичные разряды первого и второго чисел. Логические схемы И выполняют функции электронных ключей. Двоичные переменные a_n и b_n поступают на первые входы логических элементов И. Управляющим сигналом элементов является сигнал управления выполнением логической операции суммы по модулю два УпСум, который поступает на вторые управляющие входы электронных ключей. Если управляющий сигнал УпСум равен единице, то электронные ключи будут открыты и входные двоичные переменные поступают на входы логических элементов, которые выполняют логическую операцию суммы по модулю два. На выходе этих элементов формируются результаты выполнения логической операции суммы по модулю два над входными переменными. Если управляющий сигнал УпСум равен нулю, то электронные ключи будут закрыты, логическая операция сумма по модулю два на выходе блока выполняться не будет. Выходной информационный сигнал РезСумМ является результатом выполнения логической операции суммы по модулю два блоком суммы по модулю два БСумМ.

Блок выполнения логической операции эквиваленции БЭКВ состоит из логических элементов. Входными информационными сигналами блока являются двоичные разряды первого и второго чисел. Логические элементы И выполняют функции электронных ключей. Двоичные переменные поступают на первые входы логических элементов. Управляющим сигналом для электронных ключей является сигнал управления выполнением логической операции эквиваленции УпЭКВ, который поступает на вторые управляющие входы логических элементов, выполняющих функции электронных ключей. Если управляющий сигнал УпЭКВ равен единице, то электронные ключи будут открыты и входные двоичные переменные поступают на входы логических элементов, которые выполняют логическую операцию эквиваленции. На выходе этих элементов формируются результаты РезЭКВ выполнения логической операции эквиваленции над входными переменными. Если управляющий сигнал УпЭКВ равен нулю, то электронные ключи будут закрыты, логическая операция эквиваленция на выходе блока выполняться не будет. Выходной информационный сигнал РезЭКВ является результатом выполнения логической операции эквиваленции блоком эквиваленции БЭКВ.

Арифметический вычислитель

Блок БСУМ-ВЫЧ состоит из схемы определения знака результата СОЗР и схемы сумматоров-вычитателей СУМ-ВЫЧ (рис. 3). В структуру схемы сумматоров-вычитателей входят схемы коммутации и определения знака результата СОЗР. Этот блок выполняет арифметические операции суммирования и вычитания, определяет знаковый разряд результата. На вход блока поступают модули n -разрядных двоичных первого ДПЧ и второго ДВЧ чисел, признак опе-

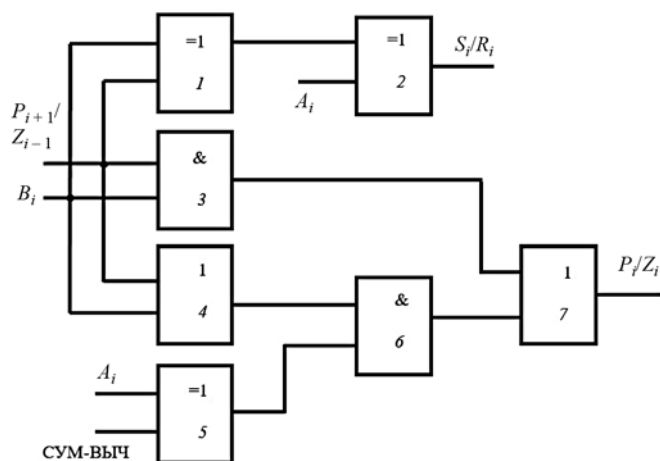


Рис. 3. Принципиальная схема одноразрядного сумматора-вычитателя

рации суммирования-вычитания сигнал СУМ-ВЫЧ и сигнал управления работой сумматоров-вычитателей УпСУМ-ВЫЧ, которые поступают параллельно на все входы схем. Если сигнал признака операции СУМ-ВЫЧ равен нулю, то схемы блока выполняют суммирование двоичных чисел, если этот сигнал равен единице, то выполняется поразрядная операция вычитания [7].

На входы каждой схемы сумматора-вычитателя СУМ-ВЫЧ₁ поступают одноименные двоичные разряды чисел a_i и b_i , сигнал управления работой сумматоров-вычитателей УпСУМ-ВЫЧ, признак операции сигнал СУМ-ВЫЧ, перенос P_i из младших разрядов в старшие и заем Z_i из старших разрядов в младшие. На схему определения знака результата СОЗР поступают знаковые разряды первого ЗнРА и второго ЗнРВ двоичных чисел, сигнал суммирования-вычитания СУМ-ВЫЧ. Сигнал заема из знакового разряда ЗмЗнР поступает на вход первого сумматора-вычитателя СУМ-ВЫЧ₁, который выполняет операции со старшими разрядами входных чисел. Этот сигнал формируется при вычитании чисел. Если заем ЗмЗнР равен нулю, то первое двоичное число по модулю больше второго. Если сигнал заема равен единице, то происходит вычитание от меньшего числа по модулю большего, а для получения разности необходимо поменять местами входные числа. Информационный сигнал результата РЕЗ является выходным сигналом блока.

Схема определения знака результата СОЗР определяет знаковый разряд результата. В структуру схемы входят логические элементы. Входные сигналы признак операции суммирования-вычитания СУМ-ВЫЧ и заем из знакового разряда результата ЗмЗнР поступают на вход логической схемы И. С выхода этого элемента сигнал поступает на управляющий вход логического элемента И, на второй вход которого поступает знаковый разряд второго числа ЗнРВ, и на вход инвертора, с выхода которого сигнал поступает на второй управляющий вход логического элемента И, на первый вход которого поступает знаковый разряд первого числа ЗнРА. Выходные сигналы логических элементов И поступают на входы логической схемы ИЛИ.

Если сигнал суммирования-вычитания СУМ-ВЫЧ равен нулю, то выполняется арифметическая

операция суммирования двоичных чисел. Входные двоичные числа суммируются по модулю, знаку суммы присваивается знак первого числа. Знаковый разряд первого числа ЗнРА через открытый логический элемент И поступит на первый вход схемы ИЛИ. Знаковый разряд результата ЗнРЕЗ, который является выходом логической схемы ИЛИ, будет равен знаковому разряду первого числа ЗнРА.

Если сигнал суммирования-вычитания СУМ-ВЫЧ равен единице, то выполняется арифметическая операция вычитания двоичных чисел, от большего по модулю вычитается меньшее. Определяющим сигналом в сравнении модулей чисел является сигнал заема из знакового разряда первого числа ЗмЗнР. Знаку разности присваивается знак большего числа. Если сигнал заема ЗмЗнР равен нулю, то первое число по модулю больше второго, в этом случае вычисляется разность между первым и вторым двоичными числами.

Если сигнал заема ЗмЗнР равен единице, то первое число по модулю меньше второго, в этом случае вычисляется разность между вторым и первым двоичными числами. Знаковый разряд второго большего числа ЗнРВ поступит на второй вход логического элемента И, выход которого поступает на второй вход схемы ИЛИ. Знаковый разряд результата ЗнРЕЗ равен знаковому разряду большего второго числа ЗнРВ.

Схема коммутации меняет местами входные двоичные разряды при выполнении операции вычитания, если от меньшего по модулю числа вычитается большее, вследствие чего формируется сигнал заема из знакового разряда первого числа ЗмЗнР. После анализа этого сигнала и выполнения перестановки чисел вследствие этого от большего второго числа по модулю вычитается меньшее первое.

В структуру схемы коммутации входят логические элементы И, ИЛИ. Входные сигналы — управляющий сигнал суммирования и вычитания УпСУМ-ВЫЧ, заем из знакового разряда первого числа ЗмЗнР и двоичные разряды чисел поступают на входы схем сумматоров-вычитателей СУМ-ВЫЧ₁, где выполняются арифметические операции одноименных разрядов двоичных чисел.

Сигнал суммирования и вычитания УпСУМ-ВЫЧ параллельно поступает на управляющие входы логических элементов И, которые выполняют функцию электронных ключей. Если сигнал управления УпСУМ-ВЫЧ равен нулю, то электронные ключи будут закрыты. Входные двоичные разряды не поступают на входы схем сумматора-вычитателя. Если сигнал управления равен единице, электронные ключи будут открыты и двоичные разряды поступают на вход схем сумматора-вычитателя для выполнения арифметических операций. Сигнал заема из знакового разряда первого числа ЗмЗнР параллельно поступает на входы инверторов и на входы логических схем И. Двоичные разряды чисел поступают на входы логических схем И, выходы которых поступают на входы логической схемы ИЛИ.

Одноразрядный сумматор-вычитатель

Полный одноразрядный сумматор предназначен для сложения трех одноразрядных двоичных чисел.

Устройство имеет три входа: двоичные разряды a_i, b_i , результат переноса P_{i+1} предыдущего сумматора, и два выхода: результат сложения чисел S_i и результат переноса в старший разряд P_i [8].

Сумма S_i определяется формулой

$$S_i = \bar{P}_{i+1} \bar{A}_i B_i \vee \bar{P}_{i+1} A_i \bar{B}_i \vee P_{i+1} \bar{A}_i \bar{B}_i \vee P_{i+1} A_i B_i = P_{i+1} \oplus A_i \oplus B_i. \quad (2)$$

Перенос P_i из младшего разряда в старший при сложении чисел определяется формулой

$$P_i = P_{i+1} A_i \vee A_i B_i \vee P_{i+1} B_i = A_i (P_{i+1} \vee B_i) \vee P_{i+1} B_i. \quad (3)$$

Полный одноразрядный вычитатель вычисляет разность трех одноразрядных двоичных чисел. Входными двоичными разрядами одноразрядного вычитателя являются заем Z_{i-1} , поступающий из соседнего более младшего разряда, уменьшаемое A_i , вычитаемое B_i , разность R_i , заем Z_i , возникающий в данном i -м разряде.

Разность чисел R_i определяется формулой

$$R_i = \bar{Z}_{i-1} \bar{A}_i B_i \vee \bar{Z}_{i-1} A_i \bar{B}_i \vee Z_{i-1} \bar{A}_i \bar{B}_i \vee Z_{i-1} A_i B_i = Z_{i-1} \oplus A_i \oplus B_i. \quad (4)$$

Затем Z_i из старшего разряда в младший определяется формулой

$$Z_i = \bar{A}_i Z_{i-1} \vee \bar{A}_i B_i \vee B_i Z_{i-1} = \bar{A}_i (Z_{i-1} \vee B_i) \vee B_i Z_{i-1}. \quad (5)$$

Формулы, по которым вычисляются сумма S_i и разность R_i , одинаковые, два сумматора по модулю два выполнены на элементах 1 и 2 булевой логики (см. рис. 3).

При выполнении операции сложения перенос P_i из младшего разряда в старший и заем Z_i из старшего разряда в младший вычисляются с помощью логических схем И, ИЛИ и сумматора по модулю два.

На входы одноразрядного сумматора-вычитателя подаются двоичные разряды чисел a_i, b_i и признак выполнения арифметической операции суммирования или вычитания сигнал СУМ-ВЫЧ. Если сигнал СУМ-ВЫЧ равен нулю, то выполняется операция суммирования двоичных чисел. Если сигнал СУМ-ВЫЧ равен единице, то выполняется операция вычитания двоичных чисел. При выполнении операции суммирования элемент 5 выполняет операцию повторителя, на выходе формируется сигнал A_i . При выполнении операции вычитания элемент 5 выполняет логическую операцию отрицания, на выходе формируется сигнал \bar{A}_i по формуле $A_i \oplus 1 = \bar{A}_i$.

На входы сумматора по модулю два элемента 1 подаются двоичные разряды переноса/заема P_{i+1}/Z_{i-1} и второго числа b_i . Цифровой элемент 1 выполняет операцию суммирования входных двоичных разрядов. Результат суммы поступает на первый вход сумматора по модулю два элемента 2. На второй вход этого элемента поступает двоичный разряд первого числа a_i . На выходе элемента 2 вычисляется сумма S_i при сложении и разность R_i при вычитании входных двоичных разрядов по формуле $S_i/R_i = P_{i+1}/Z_{i-1} \oplus A_i \oplus B_i$.

Сигнал переноса/заема P_{i+1}/Z_{i-1} и второго двоичного числа b_i параллельно поступают на входы схемы И элемента 3, выход которой подается на первый вход схемы ИЛИ элемента 7, и схемы ИЛИ элемента 4, результат выполнения этой операции подается на первый вход схемы И элемента 6, на второй вход схемы И поступает сигнал с выхода элемента 5. Выходной сигнал схемы И элемента 6 поступает на второй вход схемы ИЛИ элемента 7, на выходе которого формируется сигнал переноса/заема P_{i+1}/Z_{i-1} одноразрядного сумматора-вычитателя. Принципиальная схема сумматора-вычитателя построена согласно формулам (2)–(5) (рис. 3).

Блок регистра второго числа содержит двоичные триггеры, в которых хранится двоичный код второго числа. Входным информационным сигналом блока является сигнал управления УП. Параллельно на все входы триггеров поступают управляющие сигналы: синхронизации, установки в "0", записи и выдачи результата. Перед началом работы сумматора-вычитателя происходит обнуление всех триггеров блока по приходу из блока информационного сигнала УП.

Блок хранения результатов БХР содержит оперативное запоминающее устройство ОЗУ, двоичные счетчики Сч1 и Сч2, формирующие адреса строк и столбцов. Входным информационным сигналом блока является сигнал СУП, который поступает с выхода блока управления.

Сигнал состоит из следующих управляющих сигналов: обнуления ОБ; установки в ноль УС "0"; сигналов прямоугольных импульсов ГИ, ТИ управления работой ОЗУ устройства выбора кристалла ВК и чтение/запись Сч/Зп. Двоичные счетчики в начале работы устройства обнулены управляющими сигналами ОБ, УС "0" соответственно. На входы счетчиков поступают прямоугольные импульсы ГИ, ТИ. Счетчики формируют адреса столбцов АД СТЛ и строк АД СТР, по которым записываются результаты логических и арифметических операций РЕЗ, поступающие на вход Вх оперативного запоминающего устройства. Сигналы выбора кристалла и чтения/запись в ОЗУ соответственно при записи принимают нулевые значения ВК = 0, Сч/Зп = 0.

Параллельный сумматор комбинационного типа строится на основе каскадного соединения n одноразрядных сумматоров-вычитателей. Проведем сопоставление разработанного сумматора-вычитателя арифметического устройства с аналогами по числу вентиляей, используемых в схеме для 64-разрядной сетки [9].

В таблице приведены результаты сопоставления разработанного сумматора-вычитателя арифметического устройства с аналогами по числу вентиляей.

Число вентиляей в цифровых устройствах

Устройства	Число вентиляей
Сумматор-вычитатель устройства выполнения логических и арифметических операций	5560
Полный двоичный сумматор	5687
Параллельный сумматор-вычитатель со сквозным переносом	5765
Сумматор Когга—Стоуна (Kogge—Stoneadder)	6570

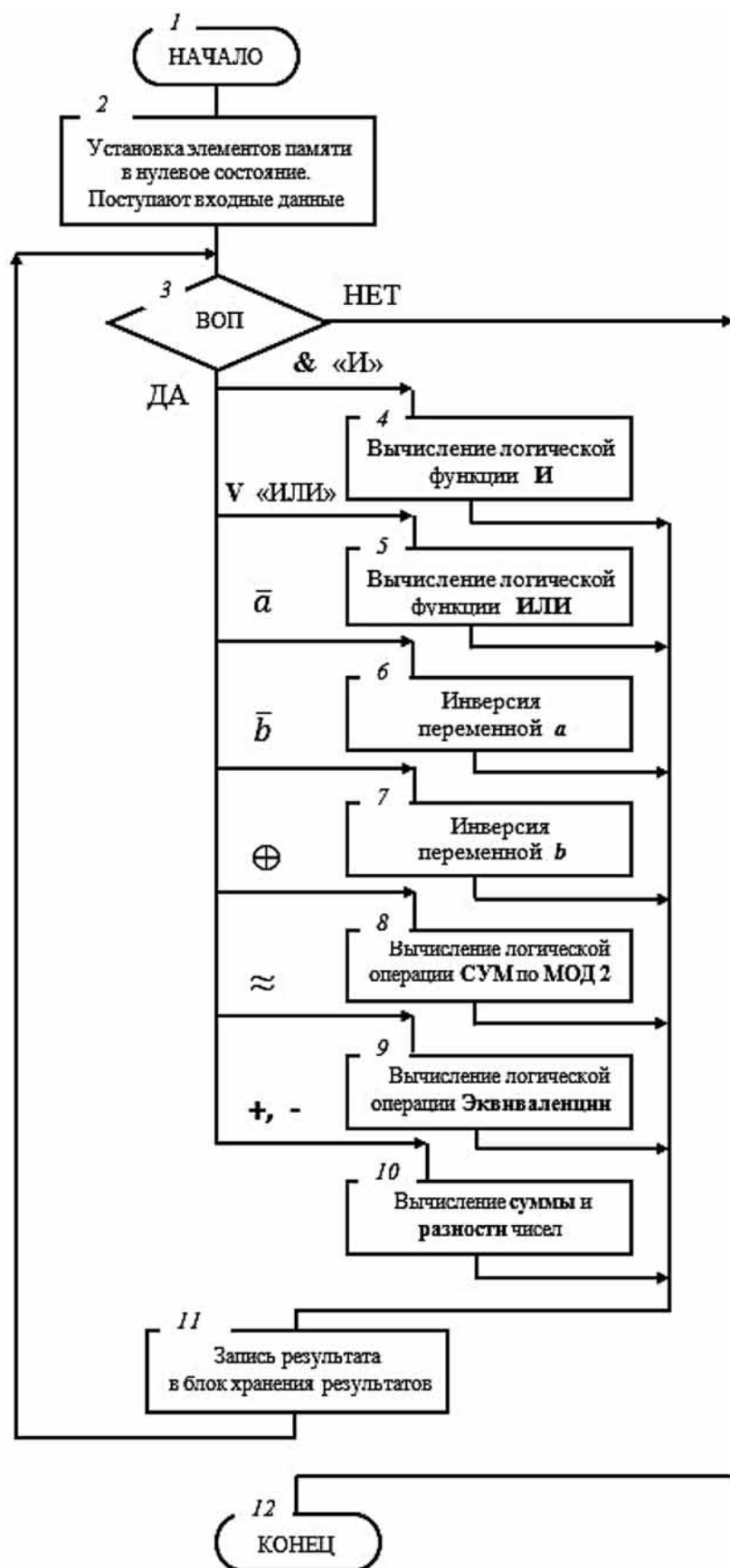


Рис. 4. Блок-схема алгоритма работы устройства выполнения логических и арифметических операций

Наименьшее число вентилях для построения арифметических вычислителей применяется в сумматоре-вычитателе устройства вычисления логических и арифметических операций. По результатам имитационного моделирования разработанное устройство имеет двукратное скоростное преимущество по сравнению с последовательной организацией переноса при сложении (вычитании) в обратных или дополнительных кодах.

Алгоритм работы устройства

Блок-схема алгоритма работы устройства выполнения логических и арифметических операций приведена на рис. 4.

Блок 1 алгоритма является начальным. В блоке 2 алгоритма происходит установка всех элементов памяти в нулевое состояние. На вход устройства выполнения логических и арифметических операций поступают входные данные. В блоке 3 анализируется признак выбора логических и арифметических операций ВОП. В блоке 4 алгоритма выполняется поразрядная логическая операция И (конъюнкция) над двоичными числами. В блоке 5 алгоритма выполняется поразрядная логическая операция ИЛИ (дизъюнкция) над двоичными числами. В блоке 6 алгоритма выполняется поразрядная логическая операция отрицания переменной a . В блоке 7 алгоритма выполняется поразрядная логическая операция отрицания переменной b . В блоке 8 алгоритма выполняется поразрядная логическая операция суммы по модулю 2. В блоке 9 алгоритма выполняется поразрядная логическая операция эквиваленции над числами. В блоке 10 алгоритма выполняются арифметические операции сложения и вычитания двоичных чисел. В блоке 11 алгоритма выполняется запись полученного результата в блок хранения результатов устройства. Блок 12 алгоритма является конечным.

Заключение

Разработанное автором устройство выполняет логические и арифметические операции, которые реализуются отдельными блоками. Цифровые устройства работают одновременно, что повышает скорость выполнения операций, но возрастают затраты на оборудование. Арифметические операции выполняются в прямом коде. Устройство можно применить в виде специализированного микропроцессора в вычислительной системе,

который выполняет арифметические и логические операции над числами с фиксированной точкой [10].

Практическая значимость полученных результатов заключается в использовании устройства в перестраиваемой вычислительной системе, что позволит повысить быстродействие выполнения арифметических операций за счет реализации конвейеризации и использования параллелизма при обработке данных, в устройстве используется распределенное децентрализованное управление блоками устройства [11].

Список литературы

1. **Хорошевский В. Г.** Архитектура вычислительных систем. — М.: Изд-во МГТУ им. Н. Э. Баумана, 2008. — 520 с.
2. **Шевелев С. С.** Вычислительная открытая развиваемая асинхронная модульная система // Проблемы информационной безопасности. Компьютерные системы Санкт-Петербургского государственного политехнического университета. — 2013. — № 4. — С. 86–92.
3. **Шевелев С. С.** Реконфигурируемая вычислительная модульная система // Известия Юго-Западного государственного университета. — 2019. — № 2. — С. 138–152.

4. **Шевелев С. С.** Арифметико-символьный процессор // Приборы и системы. Управление, контроль, диагностика. — 2012. — № 8. — С. 13–19.

5. **Шевелев С. С., Добрица В. П.** Арифметический вычислитель // Известия Юго-Западного государственного университета. — 2011. — № 3 (36). — С. 72–76.

6. **Шевелев С. С.** Математический вычислитель операций с плавающей запятой на нейронах // Нейрокомпьютеры: разработка, применение. — 2015. — № 6. — С. 27–31.

7. **Шевелев С. С., Дорошенко Е. Ю., Вин Хла.** Троичный математический вычислитель // Нейрокомпьютеры: разработка, применение. — 2015. — № 6. — С. 32–36.

8. **Хокни Р., Джессхоуп К.** Параллельные ЭВМ. Архитектура, программирование и алгоритмы. — М.: Радио и связь, 1986. — 392 с.

9. **Shevelev S. S., Dobritsa V. P.** Information Sorting by a Data Decoding Method // Telecommunications and Radio Engineering. — 2012. — Vol. 71. — P. 1475–1483.

10. **Устройство** поразрядного вычисления логических и арифметических операций: пат. 2739343 РФ. № 2020129326 / Шевелев С.С.; заявл. 04.09.2020; опубл. 23.12.2020, Бюл. № 36.

11. **Параллельно-последовательный сумматор-вычитатель** старшими разрядами вперед на нейронах: пат. 2708501 РФ. № 2019115593 / Шевелев С. С.; заявл. 21.05.2019; опубл. 09.12.2019, Бюл. № 34.

Device for Calculating Logical and Arithmetic Operations

S. S. Shevelev, schewelew@mail.ru, Southwestern State University, Kursk, 305040, Russian Federation

Corresponding author:

Shevelev Sergey S., Associate Professor, Southwestern State University, Kursk, 305040, Russian Federation

E-mail: schewelew@mail.ru

Received on June 01, 2021

Accepted on July 17, 2021

A device has been developed that performs logical and arithmetic operations, which can be used to create high-performance, high-speed computing systems. Specialized blocks perform logical operations: AND, OR, NOT, arithmetic operations: addition and subtraction of binary numbers. Arithmetic operations are performed in direct fixed-point codes. The device is presented in the form of a structural scheme, structural and functional schemes of blocks and an algorithm for the operation of the device.

Keywords: binary number system, logical elements, adder-subtractor, transfer, loan, sum, difference

For citation:

Shevelev S. S. Device for Calculating Logical and Arithmetic Operations, *Programmnyaya Ingeneria*, 2021, vol. 12, no. 7, pp. 350–357.

DOI: 10.17587/prin.12.350-357

References

1. **Khoroshevsky V. G.** Architecture of computational systems, Moscow, Bauman Moscow State Technical University, 2008, 520 p. (in Russian).

2. **Shevelev S. S.** Computational open-developed asynchronous modular system, *Problemy informacionnoj bezopasnosti. Komp'yuternye sistemy Sankt-Peterburgskogo gosudarstvennogo politekhnicheskogo universiteta*, 2013, no. 4, pp. 86–92 (in Russian).

3. **Shevelev S. S.** Reconfigurable numerical modular system, *Izvestiya Yugo-Zapadnogo gosudarstvennogo universiteta*, 2019, no. 2, pp. 138–152 (in Russian).

4. **Shevelev S. S.** Arithmetic-character processor, *Pribory i sistemy. Upravlenie, kontrol', diagnostika*, 2012, no. 8, pp. 13–19 (in Russian).

5. **Shevelev S. S., Dobritsa V. P.** Arithmetic computer, *Izvestiya Yugo-Zapadnogo gosudarstvennogo universiteta*, 2011, no. 3 (36), pp. 72–76 (in Russian).

6. **Shevelev S. S.** Mathematical calculator of floating-point operations on neurons, *Nejrokompyutery: razrabotka, primeneniye*, 2015, no. 6, pp. 27–31 (in Russian).

7. **Shevelev S. S., Doroshenko E. Yu., Khla Vin.** Ternary mathematical computer, *Nejrokompyutery: razrabotka, primeneniye*, 2015, no. 6, pp. 32–36 (in Russian).

8. **Hockney R., Jesshope K.** Parallel computers. Architecture, programming and algorithms, Moscow, Radio i svyaz, 1986, 392 p (in Russian).

9. **Shevelev S. S., Dobritsa V. P.** Information Sorting by a Data Decoding Method, *Telecommunications and Radio Engineering*, 2012, vol. 71, pp. 1475–1483.

10. **Device** for bit-by-bit calculation of logical and arithmetic operations: pat. 2739343 Ros. Federation. No. 2020129326 / Shevelev S. C.; application 04.09.2020; publ. 23.12.2020, Bul. No. 36.

11. **Parallel-sequential sum-mater-subtractor** with the highest digits forward on neurons: pat. 2708501 Growth rate. No. 2019115593 / Shevelev S. From; for-yavl. 21.05.2019; publ. 09.12.2019, Bul. No. 34.

Е. В. Орлова, д-р техн. наук, проф., ekorl@mail.ru, Уфимский государственный авиационный технический университет

Предиктивная аналитика кредитных рисков на основе данных цифровых следов заемщиков и методов статистического машинного обучения

Рассматривается задача снижения кредитных рисков банка, связанных с неплатежеспособностью заемщиков — физических лиц. Представлены новый подход, модели и методы оценки профиля риска заемщиков с учетом дополнительных факторов, обусловленных данными их цифрового следа, позволяющие осуществить комплексную оценку портрета платежеспособности заемщика. Применяются методы иерархической кластеризации и k -средних для формирования однородных групп заемщиков. Углубленный анализ данных проводится в модели классификации, основанной на методе стохастического градиентного бустинга для прогнозирования кредитного риска потенциального заемщика.

Ключевые слова: аналитика данных, машинное обучение, кластеризация, классификация, кредитоспособность, цифровой след, снижение кредитных рисков

Введение

Сфера финансовых технологий (финтех) включает разработку и практическое применение инновационных технологий в банковской сфере и других сегментах финансового сектора. Использование открытых интерфейсов (Open API) и других технологий дистанционного доступа, анализ больших данных, блокчейн, робоэдвайзинг, машинное обучение и искусственный интеллект делают финансовую индустрию в России одним из наиболее инновационных секторов экономики.

С развитием финансового сектора растет и совокупный кредитный портфель. Однако с ростом объемов кредитования увеличивается и просроченная кредиторская задолженность, что способствует росту кредитных рисков. Национальные регуляторы банковских систем, в том числе Банк России, осуществляют свою политику в соответствии с рекомендациями Базельского комитета по банковскому надзору. Комитет определяет кредитный риск как "вероятность невыполнения заемщиком или контрагентом своих обязательств в соответствии с оговоренными условиями" [1]. Вопрос методологии оценки кредитного риска прошел значительный путь развития, что нашло отражение в документах Базельского комитета и российской практике их применения. История подходов к оценке кредитного риска — это история развития подходов к нормированию банковских капиталов как основного способа обеспечения стабильности банковской системы.

В основе базельских стандартов Basel I, II, III [1–3] к достижению устойчивости финансовой системы и регулированию рисков лежит требование к достаточности капитала банков. Достаточность капитала

рассматривается как главный критерий стабильности банковской деятельности, а единственным ограничителем достаточности капитала банка является кредитный риск активов банка. Методы управления регулятора (Банка России) направлены на соблюдение резервных требований и нормативов, предъявляемых к уровню кредитного риска, и определены в нормативных документах [4, 5].

Необходимо отметить, что банковское кредитование делится на розничное кредитование, или кредитование физических лиц, и включает ипотечное кредитование, необеспеченное потребительское кредитование и автокредитование, а также корпоративное кредитование (кредиты нефинансовым организациям, финансовым организациям и индивидуальным предпринимателям) [6]. В настоящем исследовании изучается сегмент кредитования физических лиц.

Традиционно для анализа кредитоспособности клиентов кредитные организации используют скоринговые модели, с помощью которых на основе анализа кредитных историй предыдущих клиентов составляется рейтинг заемщиков и определяется вероятность возврата кредита в срок потенциальным заемщиком [7, 8]. Основные проблемные вопросы и решаемые в научных исследованиях задачи, связанные с построением и использованием скоринговых моделей при принятии решений, можно объединить в две группы.

Первая группа вопросов связана с подбором адекватного сложности решаемых задач инструментария, формированием и обоснованием системы факторов, включаемых в модель. Известные модели для оценки кредитного риска используют статистический подход и базируются на обработке эмпирической информации прошлых кредитных историй. Однако они отли-

чаются способами построения аппроксимирующих зависимостей — нейросетевые, нечеткие и гибридные алгоритмы для построения приближения [9, 10]; эконометрические методы [10—18]. Обсуждению подвергаются способы сбора необходимой информации, необходимость и число включаемых качественных характеристик в модельные конструкции для более точного отражения портрета заемщика, методы идентификации моделей, анализ их качества и прогностических свойств [8—10].

Вторая группа вопросов связана с разработкой систем автоматизации процессов сбора, обработки и хранения информации о заемщиках, с проектированием систем поддержки принятия управленческих решений банка [19—22], формированием баз данных клиентов. В условиях большого потока разнородных клиентов основным требованием при разработке таких систем является скорость принятия решений.

Анализ существующих методических подходов и аналитического инструментария показал, что существующие модели оценки риска невозврата кредитов не позволяют учесть социальные и поведенческие особенности заемщиков, выявить тенденции в поведении определенных категорий (групп) заемщиков, имеющих схожий поведенческо-финансово-экономический портрет и близкие уровни рискованности [23, 24]. Формирование таких групп позволит: во-первых, выявлять общие закономерности поведения; во-вторых, сформировать совокупность дифференцированных требований, предъявляемых к заемщикам отдельных групп с учетом их специфики; в-третьих, учесть склонность к риску лица, принимающего решения, по отдельным элементам кредита — объемам, срокам и процентной ставке.

Решение задач на этих направлениях неизбежно требует использования современных и эффективных инструментов, обеспечивающих минимальные потери из-за кредитных рисков. Машинное обучение как методы и инструменты основано на известных свойствах, извлеченных из обученных данных, предоставляет значительные возможности для обработки данных финансовых организаций. При этом постоянно совершенствуется способность методов более точно и эффективно анализировать данные, классифицировать информацию и делать предположения, что дает возможность принимать лучшие решения, основанные на данных. Финансовые организации используют различные алгоритмы машинного обучения для решения перечисленных далее задач.

1. *Извлечение информации* [25—27]. Это задачи информационного поиска, целью которых является автоматическое получение структурированных данных при обработке неструктурированной или слабоструктурированной информации, что является главным при обработке финансовых данных.

2. *Кредитный скоринг* [28—32]. Машинное обучение используется для прогнозирования кредитоспособности клиентов, а также для построения моделей кредитных рисков. Для определения кредитоспособности используются следующие алгоритмы машинного обучения: многослойный перцептрон, логистическая регрессия, метод опорных векто-

ров, алгоритм усиления классификаторов (*adaptive boosting*), квантизация векторов при обучении и др.

3. *Принятие решений* [32—35]. Финансовые вычисления и принятие решений осуществляются посредством алгоритмов машинного обучения, которые позволяют эффективнее и быстрее обрабатывать данные и принимать решения в сфере кредитования с учетом факторов, оценивающих возможности потенциального мошенничества, риск невыполнения обязательств и вероятность долгосрочных отношений с клиентами.

Качество портфеля кредитов банка может быть улучшено за счет формирования новой методики оценивания кредитоспособности заемщиков (физических лиц), обеспечивающей полную их идентификацию на основе стандартных показателей, дополненную новыми социометрическими данными, такими как данные цифровых следов. Гибкие и адаптивные системы оценки и прогнозирования кредитоспособности позволят повысить надежность и сократить кредитные риски финансовой организации.

Целью исследования, результаты которого представлены в настоящей статье, является разработка подхода, моделей и инструментальных средств оценки кредитоспособности физических лиц с учетом данных цифрового следа, позволяющих с высокой степенью надежности прогнозировать кредитные риски заемщиков, и являющиеся основой для поддержки принятия решений по формированию гибкой системы кредитования.

1. Предлагаемый концептуальный подход к оценке кредитоспособности

Предлагается подход в виде информационной технологии, основанный на поэтапной обработке информации и моделировании, отражающем антропометрические показатели, финансовые показатели и данные цифрового следа, предназначенный для оценки кредитоспособности (КС) заемщиков — физических лиц. Концептуально схема технологии представлена на рис. 1.

Этап 1. Качественный анализ заемщика и сбор исходных данных. Осуществляется анализ финансового состояния заемщика, оценка его антропометрических, социально-экономических, финансовых характеристик, а также данных цифрового следа. Результатом этого этапа являются данные о трех группах показателей — антропометрических и социально-экономических, финансовых и данных цифровых следов.

Этап 2. Выбор и обоснование факторов, влияющих на кредитоспособность заемщика. Проводится разведочный анализ данных, осуществляется оценка влияния факторов на риск заемщика на основе корреляционного анализа.

Этап 3. Формирование однородных групп заемщиков и построение типичных профилей риска заемщиков. Формирование однородных групп заемщиков осуществляется с позиции схожести значений исследуемых характеристик. Результатом данного



Рис. 1. Концептуальная схема технологии оценки кредитоспособности физических лиц

этапа являются типичные профили риска заемщиков в качественно однородных группах. На этом же этапе осуществляется классификация нового заемщика, в результате чего определяется его профиль риска.

Этап 4. Разработка комплекса управленческих решений, направленных на формирование условий кредитования — процентной ставки по кредиту, максимально возможного размера кредита, сформированные для каждой однородной группы заемщиков и спроецированные на конкретного заемщика.

Предложенная технология анализа и моделирования апробирована на данных крупного банка РФ. Анализ данных и моделирование осуществлялось с использованием средств программного комплекса Statistica 10.0.

Качественный анализ заемщика и исходные данные

Исходная информация для анализа включает данные о трех группах показателей — антропометрических и социально-экономических, финансовых и данных цифровых следов. Первые две группы показателей — это закрытые данные финансовых организаций, полученные от заемщиков в результате их первичного анкетирования, а также предоставления ими документов, предписываемых правилами финансовой организации. Третья группа показателей — это открытые данные цифрового следа заемщика, которые можно получить из данных социальных сетей, интернет-сайтов.

Сбор открытых данных социальных сетей можно реализовывать на основе прямого исследования аккаунтов пользователей социальных сетей, изучения их подписки на ресурсы различной тематики. Профили пользователей (заемщиков) в социальных сетях можно уточнять по известному адресу электронной почты. Кроме этого, можно использовать методы сбора открытой информации на основе протокола HTTP API, а также специальное программное обеспечение для извлечения информации из этих данных — PolyAnalyst, Gephy. Также для анализа посе-

щаемости сайтов можно использовать специальные инструментальные средства и метрики (ЯндексМетрика, GoogleAnalytics, LiveInternet, Spywords, Alexa и др.). Существует ряд сервисов для аналитической оценки посещаемости чужих сайтов (SimilarWeb, SemRush, SpyFu, WebInformer и др.) [36, 37].

Исходная информация о заемщиках, необходимая для анализа, формируется из разных источников и включает (табл. 1):

- антропометрическую и социально-экономическую информацию — пол, возраст, уровень образования, профессию, семейное положение, наличие детей;
- финансовую информацию о заемщике — наличие регулярного дохода, размер дохода, сумма просроченной задолженности (на основе кредитных историй), желаемую сумму кредита;
- данные о цифровом следе заемщика, получаемые из социальных сетей и поисковых систем.

Анализ данных социальных сетей позволит сформировать цифровой профиль заемщика.

Перевод качественных значений признаков в количественные осуществляется на основе двоичного кодирования (0 и 1), при этом количественное значение признака увеличивается по мере интенсификации его качественной характеристики.

2. Отбор и обоснование факторов, влияющих на кредитоспособность (разведочный анализ)

Для анализа были использованы фактические данные о потребительском кредитовании одного из крупных банков РФ. База исследуемых данных включает 100 заемщиков со всеми исследуемыми характеристиками (см. табл. 1). Необходимо отметить, что по каждому заемщику дополнительно к предоставленной банком информации был осуществлен сбор данных их цифровых следов с использованием различных сервисов и инструментов.

Проведено "несплошное" обследование, и выборочная совокупность заемщиков образована из

Исследуемые показатели, обозначения и диапазон значений

Группа показателей	Исследуемый показатель	Обозначение	Значение
Антропометрические и социально-экономические показатели	Пол	gender	Женский (1), мужской (0)
	Возраст	age	18...65
	Уровень образования	edu	Среднее, среднее специальное (0), высшее (1)
	Профессия	prof	Любая профессия (1), домохозяйка, студент (0)
	Семейное положение	mar	Неженат/незамужем (0), женат/замужем (1)
	Наличие детей	child	0, 1, 2, 3 и т. д.
Финансовые показатели	Наличие регулярного дохода	avinc	Да (1), нет (0)
	Размер дохода	aminc	0...1 000 000
	Желаемая сумма займа	dessum	0...1 000 000
	Сумма просроченной задолженности (риск)	risk	0...1 000 000
Данные цифрового следа	Вредные привычки	bad hab	Да (0), нет (1)
	Интересы	ints	Карьера, семья, философия (1); антиколлектор, азартные игры (0)
	"Плохое" окружение ¹	bad env	1 и более (0); 0 (1)
	Стиль музыки	mus	Классика, поп, джаз (1); тюремного характера, запрещенные в РФ (0)
	Жанр фильмов	mov	Комедии, семейные, драмы (1), запрещенные в РФ (0)
	Реальность доходов	inc	Соответствует (1); отличается (0)
	"Идеальный семьянин" ²	ideal fam	Да (1); нет (0)
	Оценка профиля заемщика	profile	Соответствует (1); отличается (0)
	Частота входов на сайт по тематике мошенничества	fraud	1 и более (0); меньше 1 (1)
	Частота входов на сайт по тематике болезней	illness	1 и более (0); меньше 1 (1)
	Частота входов на сайт по тематике азартных игр	gambling	1 и более (0); меньше 1 (1)
	Частота входов на сайт по тематике распространения и употребления наркотических средств	drugs	1 и более (0); меньше 1 (1)
	Частота входов на сайт по тематике запрещенных организаций в РФ	forbidden	1 и более (0); меньше 1 (1)
Частота входов на сайт по тематике развития бизнеса и саморазвития	career	1 и более (0); меньше 1 (1)	

¹ Характеризует круг общения заемщика. Анализу подвергаются профили в социальных сетях не только самого заемщика, но и его друзей.

² Характеризует заемщика, который демонстрирует развитие в своей карьере и проявляет большой интерес к своей семье (по фотографиям и записям в социальных сетях).

сравнительно небольшого числа заемщиков. Для обеспечения репрезентативности выборки применен случайный отбор заемщиков, при котором вероятность попадания заемщиков в выборку одинакова. Доверительные интервалы для генеральной средней количественных признаков, кроме переменной "risk" при 5%-ном уровне значимости, не широкие. Численные оценки нижней и верхней границ доверительных интервалов отличаются от выборочной средней не более чем на $\pm(5...7)$ %. Переменная "risk" обладает значительной вариацией, что можно объяснить тем, что ряд заемщиков не имеют задолженностей по кредитам и их кредитный риск равен нулю. Другая часть заемщиков, напротив, имеет значительные задержки платежей и, соответственно, характеризуется значительным кредитным риском. Исходя из этого можно предполагать, что данный объем выборки обеспечивает достоверность и приемлемую точность результатов измерения.

Для определения и визуального представления предполагаемого воздействия качественных признаков на показатель просроченной задолженности (риска) заемщиков построены диаграммы размаха (рис. 2).

Анализ статистических характеристик величины риска в зависимости от пола заемщика (рис. 2, а) показал, что мужчины-заемщики обладают большими рисками, чем женщины. Среднее значение величины риска

у мужчин составляет 1572 денежные единицы в год, а у женщин — 900. На рис. 2 также видно, что вариация риска у мужчин тоже выше. При этом число просрочек среди женщин почти в 2 раза выше, что обеспечивает примерно одинаковые суммы просроченной задолженности у мужчин и женщин в целом по совокупности.

Анализ зависимости риска заемщика от наличия у него нежелательных привычек показал, что совокупный риск у заемщиков, не имеющих вредных привычек, составляет 44 400 денежных единиц, что почти на 63 % ниже, чем у заемщиков, у которых были выявлены вредные привычки (70 500 денежных единиц). Центры распределения и вариабельность риска статистически не отличаются у лиц, имеющих и не имеющих вредные привычки, рис. 2, б. У одиноких заемщиков риски выше почти на 110 %. Так, совокупный риск у таких заемщиков равен 82 400 денежных единиц, а у замужних (женатых) составляет 32 500 денежных единиц. Уровень образования не влияет на риск и составляет примерно одинаковые суммы для лиц, имеющих высшее образование, и для всех остальных (около 57 500 денежных единиц). При этом вариабельность риска у более образованных заемщиков значительно ниже. Это дает основание сделать вывод, что более образованные заемщики имеют более высокую финансовую дисциплину и их индивидуальные риски в среднем по совокупности ниже.

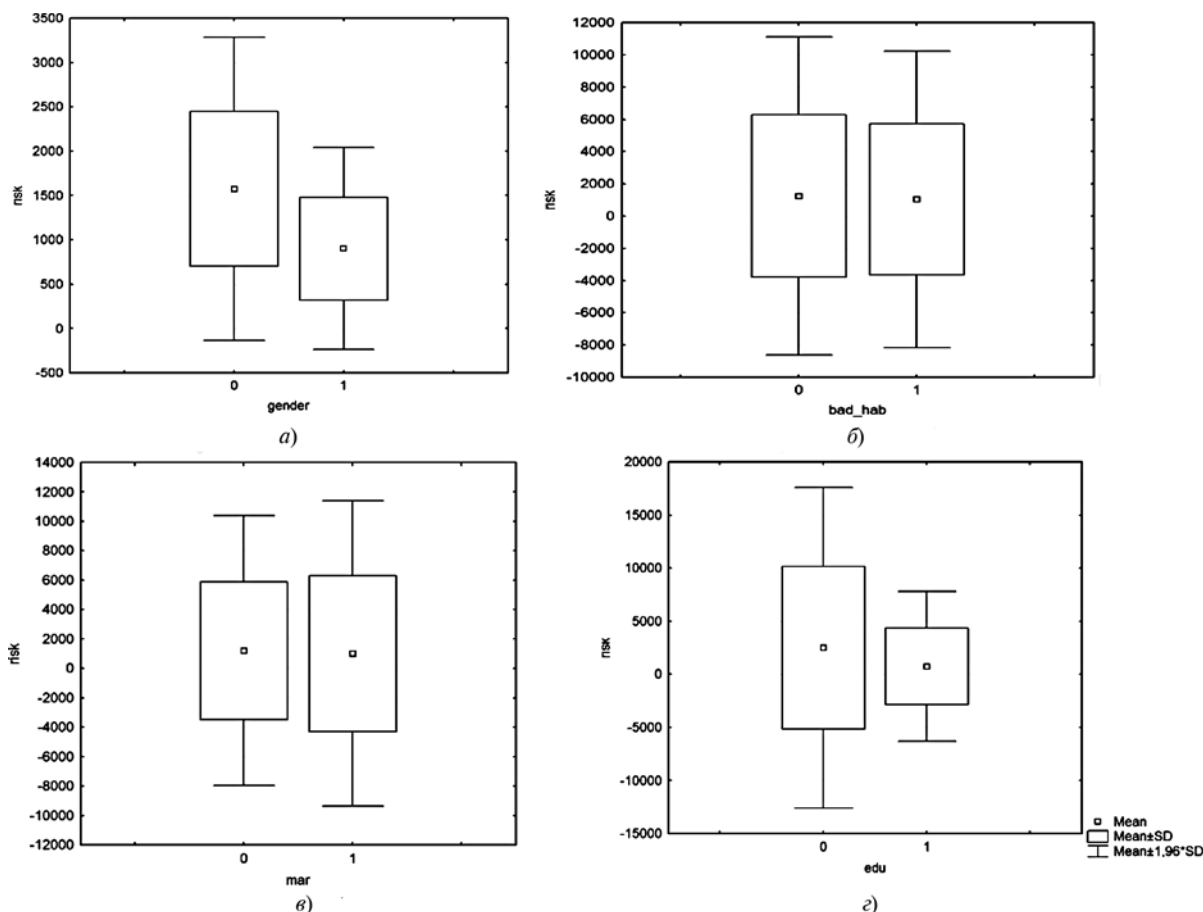


Рис. 2. Диаграммы размаха по группам факторов gender и risk (а), bad_hab и risk (б), mar и risk (в), edu и risk (г)

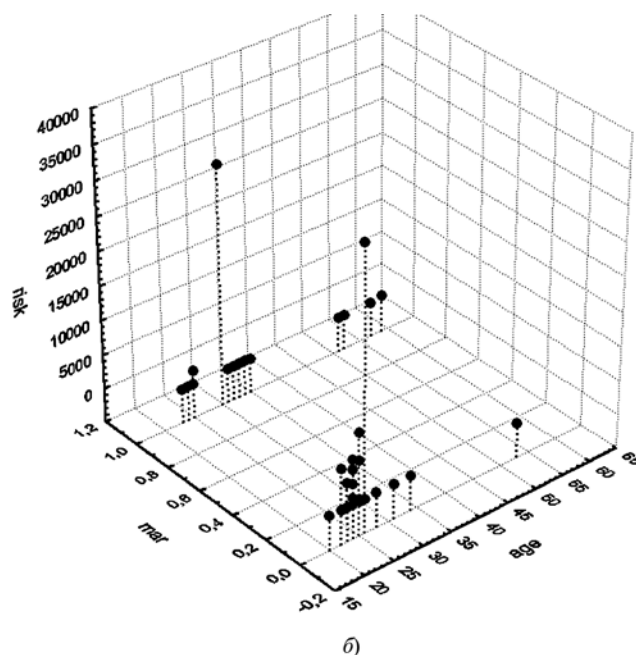
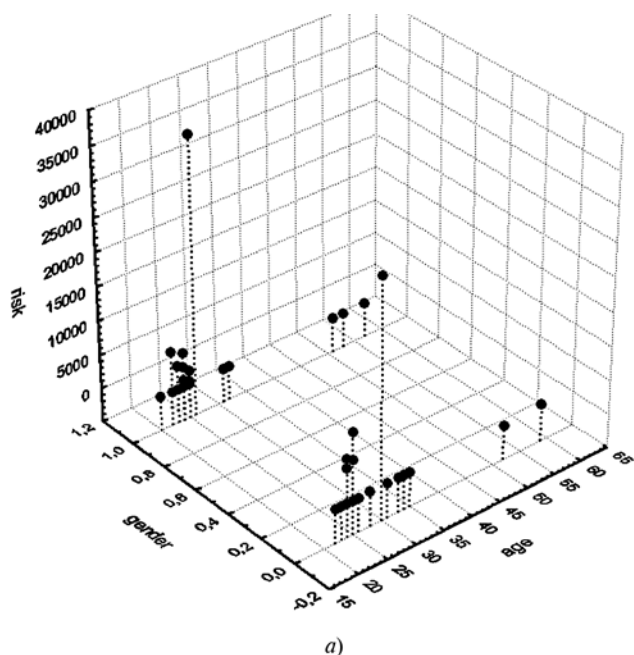


Рис. 3. 3D-диаграммы рассеяния по группам факторов:
 а — age, gender, risk; б — age, mar, risk

Трехмерная диаграмма рассеяния на рис. 3 демонстрирует, что в основном риски присущи молодым заемщикам в возрасте до 35 лет. Более возрастные заемщики обладают меньшими рисками, что можно объяснить более высокой финансовой ответственностью и дисциплинированностью. Распределение данных по показателям age, mar и risk показывает, что большие риски свойственны молодым одиноким заемщикам. График зависимости величины риска от величины кредита приведен на рис. 4, из которого видно, что значительные риски присущи крупным кредитам, но по суммарному значению потерь преобладают невысокие займы.

Описательные статистики показателя риска (табл. 2) характеризуют значительную неоднородность исследуемых данных. Поэтому для выявления зависимостей и закономерностей в данных необходимо

Таблица 2

Описательные статистики показателя "риск"

Показатель	Расчетное значение	Показатель	Расчетное значение
Среднее арифметическое, ед.	1149	Среднее квадратическое отклонение, ед.	4864
Максимальное значение, ед.	35 800	Коэффициент вариации, %	423

разбиение исходной совокупности на качественно однородные группы на основе применения методов классификации.

Для выявления парных взаимосвязей факторов с величиной просроченной задолженности (риском) заемщика проводится корреляционный анализ. Необходимо отметить, что обследуемые показатели измерены в разных шкалах — риск, возраст и число детей — в непрерывной метрической шкале, уровень образования — в порядковой (ранговой шкале), а остальные показатели — в номинальной, дихотомической шкале. Поэтому анализ взаимосвязей факторов в целях выявления значимого влияния их на моделируемый показатель, которым является риск заемщика, должен осуществляться с использованием разных метрик. Для измерения связи между риском, возрастом и числом детей, например, используется коэффициент корреляции Пирсона, для оценки влияния уровня образования на риск — коэффициент ранговой корреляции Спирмена, а для оценки влияния категориальных (дихотомических) переменных на риск — многофакторный дисперсионный анализ.

Оценка коэффициента Пирсона для пар возраст-риск (age-risk), доход-риск (aminc-risk), размер кре-

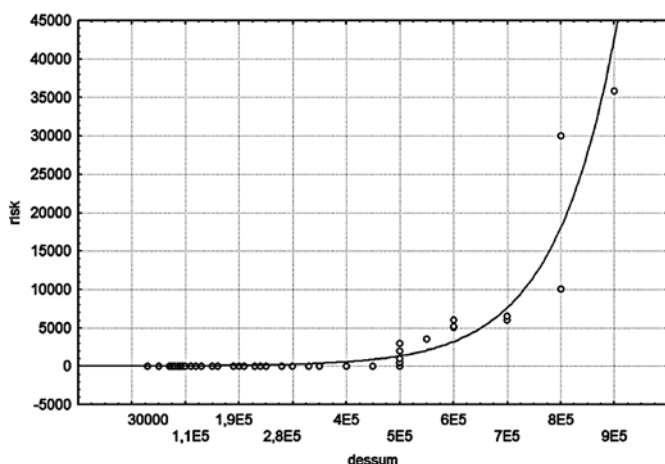


Рис 4. График зависимости риска от размера займа

дита — риск (dessum-risk) показала, что эти факторы в отдельности не имеют статистически значимого влияния на размер риска (расчетное значение t -критерия Стьюдента меньше табличного при уровне значимости 0,05). Расчет коэффициентов корреляции Спирмена для оценки влияния неколичественных переменных на уровень риска выявил наличие корреляции между сферой интересов заемщика и его надежностью, а также увлеченностью определенным жанром музыки и надежностью. При этом показатель "жанр музыки" тесно связан с рядом факторов — уровень образования, сфера занятости, объем кредита, сфера интересов, негативная сфера окружения, частота входов на сайты по тематике мошенничеств и азартных игр. Показатель "сфера интересов" статистически значимо связан с показателями пол, уровень образования, семейный статус, число детей, наличие плохих привычек, "плохое" окружение заемщика, характеристика "идеальный семьянин", а также частота входов на сайты по тематике мошенничеств и азартных игр.

Для исключения ложных корреляций рассчитана матрица частных корреляций, которая выявила, что переменные *ints* — интересы, *mus* — стиль музыки и *gambling* — увлеченность азартными играми значимо влияют на риск заемщика. Кроме того, тесные частные корреляции наблюдаются между факторами *dessum* и *gender*, *gambling* и *gender*, *bad_hab* и *gender*, *mar* и *ideal_fam*, а также между парами факторов *gambling* и *ints*. Этот факт объясняется тем, что наличие плохих привычек зависит от пола заемщика, семейные заемщики, как правило, имеют стабильные отношения в семье, а частота входов на сайты по тематике азартных игр часто связана с наличием нежелательных привычек у заемщика.

3. Модели кластеризации и классификации заемщиков

Модель кластеризации заемщиков

Для анализа используется массив полученных на этапе 1 технологии данных о качественных и количественных показателях, отражающих финансовые характеристики заемщиков. Такие характеристики включают: наличие регулярного дохода, размер дохода, желаемую сумму кредита заемщика, сумму просроченной задолженности (риск), антропометрические и социальные характеристики. К числу социальных характеристик относятся: пол, возраст, образование, семейное положение, наличие детей, профессия и данные цифрового следа. Данные цифрового следа — вредные привычки, интересы, "плохое" окружение, стиль музыки, жанр фильмов, реальность доходов, "идеальный семьянин", оценка профиля, частота входов на сайт по тематике мошенничества, частота входов на сайт по тематике болезней, частота входов на сайт по тематике азартных игр, частота входов на сайт по тематике распространения и употребления наркотических средств, частота входов на сайт по тематике запрещенных организаций в РФ, частота входов на сайт по тематике развития бизнеса и саморазвития.

Для того чтобы сгладить выявленные неоднородности данных, а также упорядочить сложные взаимодействия факторов друг с другом необходима процедура разбиения данных на однородные группы. Это позволит более подробно исследовать данные и выявить закономерности в полученных однородных группах. Возможно, что в разных группах будут иметь место разные по силе воздействия и численности факторы, определяющие рост или снижение уровня риска.

Для выбора наиболее эффективных методов, позволяющих решить задачу кластеризации в контексте второго этапа представляемой технологии, дополнительно решается подзадача по сопоставлению эффективности разных методов статистического машинного обучения и выбор одного из методов, обладающих максимальным значением критерия качества. Сравнивают две модели кластеризации — модель, основанную на последовательном разбиении на кластеры и использующую метод иерархической классификации и метод k -средних (модель 1), и модель на основе метода машинного обучения — *Automated Neural Networks* и самоорганизующихся карт Кохонена (модель 2).

Качественные картины полученных разбиений по модели 1 и по модели 2 схожи. Однако анализ устойчивости используемых моделей показал следующее. Были внесены два типа изменений в исходные данные в обучающей выборке — осуществлялось укорочение выборки на 10 % и был добавлен дополнительный фактор, характеризующий число дней просрочки выплаты кредита. Далее, с учетом этих изменений получены новые разбиения на кластеры по модели 1 и по модели 2. Кластеризация по модели 1 показала большую стабильность групп с позиции их одинакового состава до и после изменений, чем кластеризации на основе модели 2. Поэтому для решения поставленной задачи кластеризации заемщиков и формирования их однородных групп наиболее целесообразно использовать модель 1. Далее рассмотрен процесс кластеризации по модели 1.

Кластеризация проводится в два этапа — качественный анализ с помощью иерархических методов, затем анализ с помощью метода k -средних. Разведочный анализ по выяснению возможного числа групп проводится с помощью иерархической классификации с использованием разных мер сходства и различия объектов в группах — евклидово расстояние, расстояние Манхэттена, расстояние Чебышева — для оценки степени близости объектов внутри групп и мер расстояний между кластерами — одиночной, средней и полной связи. Изменяя меры расстояния при формировании кластеров, можно качественно оценить возможный состав кластеров и их число. Анализ различных разбиений исходной выборки методом иерархической классификации показал, что можно образовать от трех до пяти кластеров (рис. 5).

Для более обоснованной группировки объектов необходимо использование методов кластеризации, использующих количественные критерии для оценки качества разбиения. К таким методам относят метод k -средних. Строго говоря, алгоритм k -средних

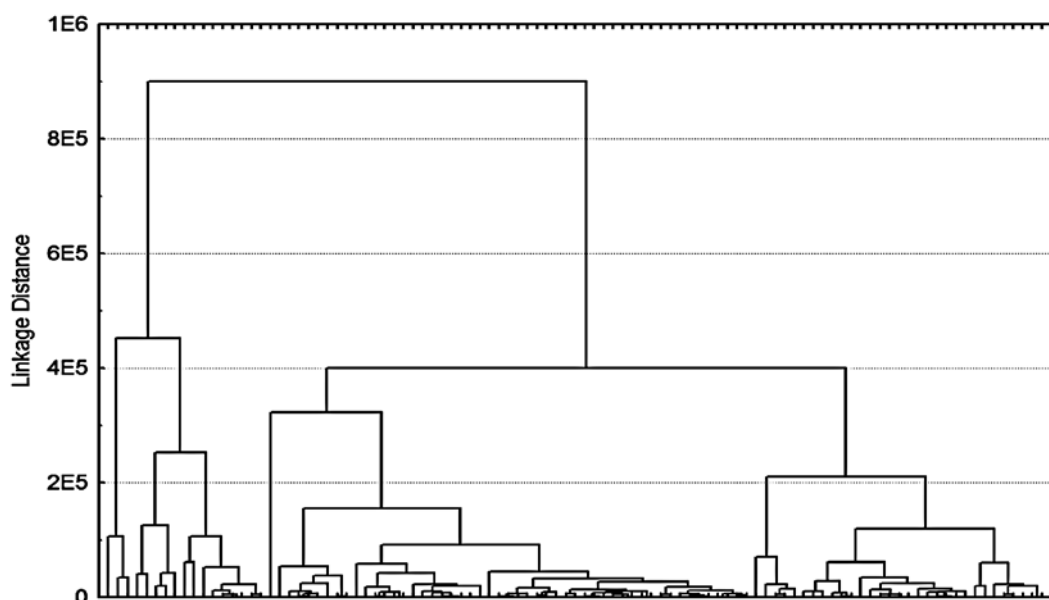


Рис. 5. Итоговая диаграмма иерархической кластеризации заемщиков, мера сходства объектов в группах — евклидово расстояние, мера расстояний между кластерами — метод полной связи

(*k-means*) применим для разбиения на кластеры только числовых данных [38]. В случае наличия в исходных данных категориальных переменных используются модификации этого алгоритма — алгоритмы *k-modes* и *k-prototypes*. Они отличаются тем, что используют другие меры близости объектов — процент несогласия, смешанное расстояние Евклида—Хэмминга. В работе применяется смешанное расстояние Евклида—Хэмминга, а в качестве функционала, отражающего критерий оптимальности разбиения и выражающего уровни желательности различных альтернативных разбиений в процессе кластерного анализа, выступает центроидный метод. В табл. 3 представлены результаты кластеризации, в которой содержится четыре кластера ($k = 4$).

Распределение заемщиков в полученных кластерах по уровням исследуемых факторов позволяет провести более детальный анализ их кредитоспособности, выявив отличительные особенности заемщиков, принадлежащих разным группам. Это позволяет сформировать профили заемщиков каждого кластера в целях дальнейшего моделирования стратегий по управлению условиями кредитования.

Описательные статистики количественных показателей характеризуют значимую однородность полученных кластеров заемщиков (табл. 4). Категоризованные гистограммы по качественным признакам в каждом кластере заемщиков показаны на рис. 6.

Таким образом, на данном этапе реализации технологии получена информация о числе кластеров заемщиков и их детальных характеристиках. В первом, самом многочисленном кластере преобладают замужние женщины в среднем 29 лет с высшим образованием, имеющие работу с регулярным доходом со средним размером 31 531,2 руб. Заемщики этого кластера не имеют вредных привычек, с приемлемыми интересами в музыке и в фильмах (не имеют пристрастия к запрещенному в РФ контенту). В среднем желаемая сумма кредита 248 125 руб.,

Таблица 3

Средние значения показателя в кластере (центры кластеров)

Показатель	Номер кластера			
	1	2	3	4
age	29,2	24,2	23,7	27,6
child	0,62	0,03	0	0,22
aminc	31 531,2	25 137,93	38 941,2	38 500
gender	1	1	0	0
edu	1	1	0	1
empl	1	1	1	1
mar	1	0	0	0
avinc	1	1	1	1
bad hab	1	1	0	0
ints	1	1	0	1
bad env	1	1	1	0
mus	1	1	1	1
mov	1	1	1	1
inc	1	1	1	1
ideal fam	1	0	0	0
profile	1	1	1	1
fraud	1	1	1	1
illness	1	1	1	1
gambling	1	1	1	1
drugs	1	1	1	1
forbidden	1	1	1	1
career	0	1	0	1
Численность кластера	32	29	17	22

Описательные статистики количественных показателей

Показатель	Статистическая метрика	Номер кластера			
		1	2	3	4
Возраст, лет	Minimum	20	22	20	22
	Maximum	60	32	26	59
	Mean	29,2	24,2	23,7	27,7
	Standart deviation	9,87	1,68	1,44	8,1
Число детей	Minimum	0	0	0	0
	Maximum	2	1	0	2
	Mean	1	0,1	0	0,3
	Standart deviation	0,4	0,02	0	0,05
Размер дохода, руб.	Minimum	10 000	2000	100 000	18 000
	Maximum	100 000	60 000	300 000	109 000
	Mean	31 531	25 137	138 941	38 500
	Standart deviation	17 542	10 252	67 755	21 054
Желаемая сумма кредита, руб.	Minimum	70 000	50 000	100 000	30 000
	Maximum	700 000	900 000	550 000	800 000
	Mean	248 125	178 448	165 300	340 681
	Standart deviation	167 166	181 339	53 569	232 083
Сумма просроченной задолженности, руб.	Minimum	0	0	0	0
	Maximum	30 000	5000	35 800	3500
	Mean	917	172	3735	159
	Standart deviation	5300	928	8837	745

а средняя просроченная задолженность составляет 937,5 руб. Профили заемщиков данного кластера в Интернете соответствуют реальности, т. е. данные цифрового следа вполне достоверны. Заемщики этого класса демонстрируют себя как идеальные семьянины и не интересуются такими темами, как мошенничество, азартные игры, наркотические средства. Этот кластер является наименее рискованным из всех кластеров и характеризуется отсутствием кредитного риска.

Во втором кластере преобладают незамужние женщины 24 лет с высшим образованием, имеющие работу с подтвержденным регулярным доходом, составляющим в среднем 25 137,93 руб., не имеющие вредных привычек, с хорошими интересами в музыке и в фильмах. В среднем желаемая сумма кредита 178 448,3 руб., а средняя просроченная задолженность составляет 172,41 руб. Они демонстрируют себя неидеальными семьянинами, их профили в Интернет соответствуют реальности и они не интересуются темами мошенничества, азартных игр и наркотических средств. Второй кластер обладает низким уровнем риска.

В третьем кластере больше неженатых мужчин 23 лет со средним специальным образованием, имеющих работу с регулярным доходом со средним размером 38 941,2 руб.

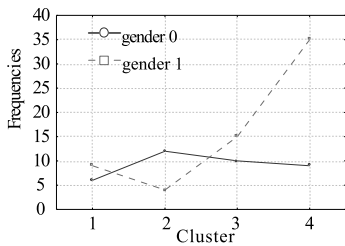
В среднем желаемая сумма кредита составляет 165 300 руб., а средняя просроченная задолженность

равна 3735,29 руб. Заемщики этого кластера имеют вредные привычки, но при этом их характеризует хорошее окружение и приемлемые интересы в музыке и в фильмах. Их профили и доходы соответствуют реальности, но при этом они демонстрируют себя неидеальными семьянинами. Заемщики кластера не интересуются темами мошенничества, азартных игр, наркотических средств и т. д. Данный кластер является самым рискованным, имеющим высокий уровень кредитного риска.

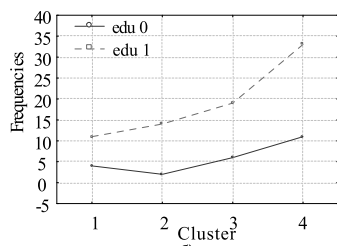
В четвертом кластере преобладают неженатые мужчины 27 лет с высшим образованием, имеющие работу с регулярным доходом со средним размером 38 500,1 руб. В среднем желаемая сумма кредита 340 681,8 руб., а средняя просроченная задолженность составляет 159,09 руб. Имеют вредные привычки и плохое окружение, но с хорошими интересами в музыке и фильмах. Демонстрируют себя как неидеальные семьянины и не интересуются темами мошенничества, азартных игр, наркотических средств и т. д. Кластер имеет средний уровень риска. Итоговое распределение кластеров по уровням кредитного риска представлено в табл. 5.

Рис. 6. Распределение заемщиков в кластерах по уровням категоризованных переменных:

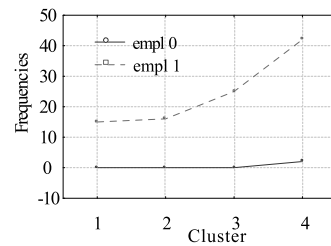
a – gender; *b* – edu; *v* – empl; *z* – mar; *d* – avinc; *e* – bad_hab; *жс* – 12 + ints; *з* – bad_env; *и* – mus; *к* – mov; *л* – inc; *м* – ideal_fam; *н* – profile; *о* – fraud; *п* – illness *р* – gambling; *с* – career; *т* – drugs



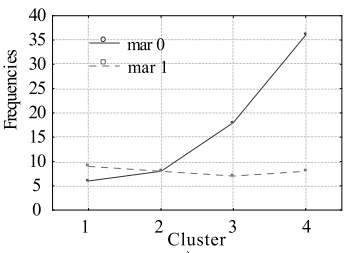
a)



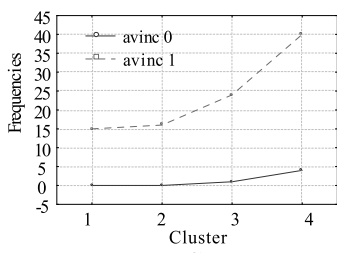
б)



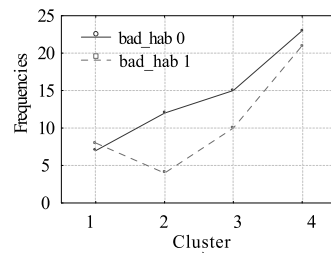
в)



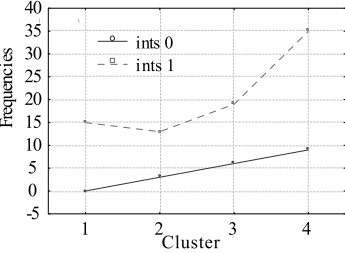
г)



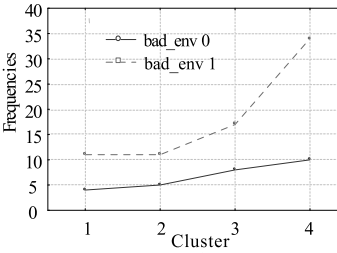
д)



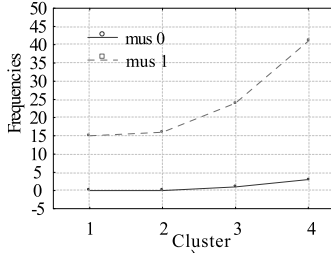
е)



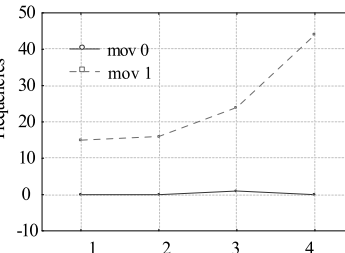
ж)



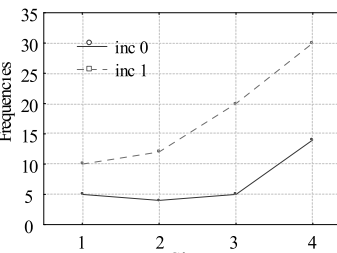
з)



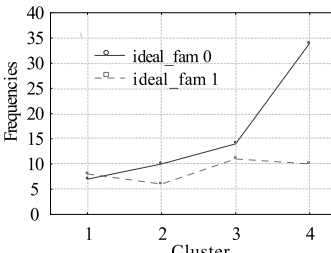
и)



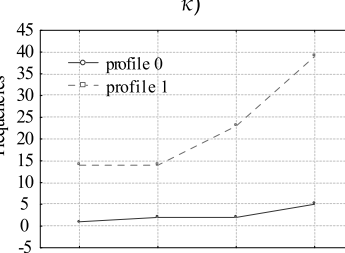
к)



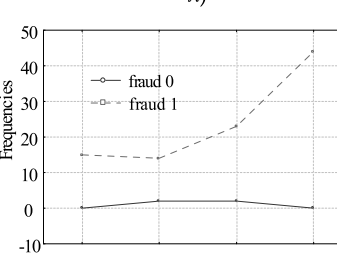
л)



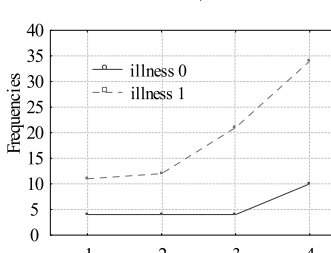
м)



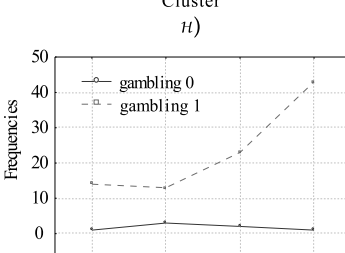
н)



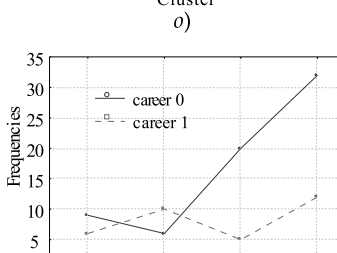
о)



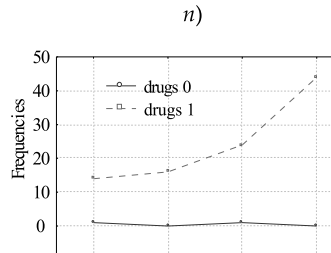
п)



р)



с)



м)



Таблица 5

Кластеры заемщиков с соответствующими характеристиками кредитного риска и надежности

Кластер	Уровень кредитного риска	Надежность заемщика
1	Риск отсутствует	Очень высокая
2	Низкий	Высокая
3	Высокий	Низкая
4	Средний	Средняя

На основании выявленного уровня риска (высокий, средний, низкий риск, риск отсутствует) в соответствии с Инструкцией Национального Банка России можно оценить премии по кредитным рискам с учетом достаточности капитала банка [4, 5]. Учитывая, что процентная ставка банка по кредиту определяется исходя из стоимости заемного ресурса для банка, премии за риск, расходов банка на оформление кредита и прибыли банка, а премия за риск может достигать до 50...70 % от значения процентной ставки, ее обоснованный расчет является значимым и необходимым.

Модель классификации заемщиков

Получив четыре однородных класса заемщиков, можно сформировать их профили риска (как совокупность значений характеристик, однозначно отличающих заемщиков разных кластеров друг от друга) для дальнейшего обоснованного построения адекватных стратегий управления кредитными рисками. Возникает задача определения группы и соответственно профиля, который будет иметь новый потенциальный заемщик, изначально отсутствующий в исходной выборке данных. Для решения этой задачи применяется классификация с учетом уже обоснованных четырех кластеров заемщиков, под которой понимается отнесение каждого из заданного множества объектов (заемщиков) к одному из заранее известных классов. Для этого необходимо выбрать методы и определить их сравнительную эффективность при проведении классификации. Классификация должна быть устойчива к шумам в исходных данных и давать высокоточные результаты.

Рассматриваются следующие типы классификаторов: метрический, линейный, бустинг. Метрические классификаторы просты в применении, в качестве методов обучения используют анализ сходств объектов в выборке, но не обладают гибкостью, неустойчивы к шумам и выбросам в исходных данных. Линейные классификаторы являются гибкими алгоритмами, однако они ограничены тем, что относят объекты к одному из двух классов, т. е. используются для бинарной классификации. В случае задачи, решение которой представлено в настоящей публикации, этот классификатор не подходит. Третий тип классификаторов — бустинг — позволяет объединить слабые классификаторы в один сильный [39] и на основе такого объединения позволяет устранить недостатки каждого алгоритма.

Применение метрического классификатора на основе алгоритма KNN (*k-Nearest Neighbors*) и бустинга

на базе алгоритма SGB (*Stochastic Gradient Boosting*) для решения задачи классификации нового заемщика показал следующие результаты точности. Оценка качества классификации оценивается числом верных предсказаний — кластера, к которому относится заемщик в тестовой выборке (75 % данных о заемщиках используется в качестве обучающей выборки, а 25 % — в качестве тестовой выборки). Так, в модели на основе KNN получено 83,4 % верных отнесений заемщиков к кластерам, а в модели на основе SGB получено 94,1 % правильных предсказаний. В данном исследовании показано, что SGB-модель в условиях множества категориальных предикторов — все переменные, отражающие цифровой след заемщиков, прогнозирует с высокой точностью класс заемщика, т. е. его профиль риска. KNN-модель наиболее пригодна для прогнозирования, когда имеется множество количественных предикторов. Далее более подробно рассматривается модель классификации на основе алгоритма SGB.

Постановка задачи классификации

Имеется множество объектов — заемщиков X — и множество непересекающихся классов кредитного риска Y , к которым принадлежат заемщики. Существует целевая функция $y^* : X \rightarrow Y$, значения которой $y_i = y^*(x_i)$ известны только на конечном подмножестве объектов $\{x_1, \dots, x_l\} \subset X$. Совокупность пар $X^l = (x_i, y_i)_{i=1}^l$ образует обучающую выборку заемщиков с номерами классов риска.

В общем случае задачей обучения является: по выборке X^l восстановить зависимость y^* , т. е. построить решающую функцию (алгоритм) $a : X \rightarrow Y$, которая приближает целевую функцию $y^*(x)$ не только на объектах обучающей выборки, но и на всем множестве X . В решаемой задаче классификации имеется M непересекающихся классов $\{y_1, \dots, y_M\} \subset Y$. В этом случае все множество объектов X разбивается на классы $P_y = \{x \in X : y^*(x) = y\}$, а алгоритм $a(x)$ дает ответ на вопрос о том, к какому классу принадлежит заемщик x .

При решении задач классификации часто бывает, что ни один из используемых алгоритмов не обеспечивает требуемой точности предсказания. Одним из альтернативных вариантов решения может быть построение композиций этих алгоритмов для компенсации недостатков каждого из них. Композицией K алгоритмов $a_k(x) = C(b_k(x))$, $k = 1, \dots, K$ будем называть суперпозицию алгоритмических операторов $b_k : X \rightarrow R$, корректирующей операции $F : R^K \rightarrow R$ и решающего правила $C : R \rightarrow Y$: $a(x) = C(F(b_1(x), \dots, b_K(x)))$, $x \in X$. Алгоритмическая композиция примет вид

$$a(x) = C(F(b_1(x), \dots, b_K(x))) = \arg \max_{y \in Y} \sum_{k=1}^K a_k b_k(x), x \in X. \quad (1)$$

С учетом изложенного, алгоритм классификации $a_k : X \xrightarrow{b_k} R \xrightarrow{C} Y$ имеет следующую структуру

и последовательность шагов: сначала $b(x)$ вычисляет некоторую оценку попадания заемщика в тот или иной класс; далее, с помощью решающего правила алгоритм переводит их в конечный результат — номер класса. С помощью пространства оценок R расширяется множество допустимых корректирующих операций, так как при определении F как отображения $Y^l \rightarrow Y$ возникает сложность выбора приемлемого F как агрегирующей функции или металагоритма. При комбинировании ответов алгоритмических операторов операция использует оценки принадлежности заемщика к классам, которые более точны. Будем использовать линейные комбинации (взвешенное голосование) и настраивать для каждого базового алгоритма свой коэффициент.

Функционал качества алгоритма (1) определяется как число ошибок, допускаемых на обучающей выборке:

$$Q_K = \sum_{j=1}^l \left(\arg \max_{y \in Y} \sum_{k=1}^K a_k b_k(x_j) \neq y_j \right). \quad (2)$$

Задача заключается в минимизации функционала (2). Для упрощения введем эвристику: пороговая функция потерь функционала качества заменяется непрерывно дифференцируемой оценкой сверху $L(M)$. Эта оценка является одним из варьируемых параметров:

$$Q_K \leq Q'_K = \sum_{i=1}^l L \left(\sum_{k=1}^K a_k b_k(x_i), y_i \right).$$

В целях минимизации функционала (2) введем еще одну эвристику: при добавлении k -го слагаемого оптимизируются только k -й базовый алгоритм и коэффициент при нем, все ранее введенные слагаемые остаются фиксированными. С помощью этого приема оптимизируется набор базовых алгоритмов, т. е. при обучении следующего алгоритма повышается вес объектов, на которых совершалась ошибка классификации. Таким образом, удаётся учесть ошибки предыдущих базовых алгоритмов. С учетом темпа обучения (градиентного шага) η имеем:

$$Q(\eta, b; X^l) = \sum_{i=1}^l L \left(\sum_{k=1}^{K-1} a_k b_k(x_i) + \eta b(x_i), y_i \right) \rightarrow \min_{\eta, b}.$$

Дополнительно введем следующие обозначения:

$$f_{K-1} = (f_{K-1,i})_{i=1}^l = \left(\sum_{k=1}^{K-1} a_k b_k(x_i) \right)_{i=1}^l \quad \text{— текущее}$$

приближение,

$$f_K = (f_{K,i})_{i=1}^l = \left(\sum_{k=1}^{K-1} a_k b_k(x_i) + \eta b(x_i) \right)_{i=1}^l \quad \text{— следу-}$$

ющее приближение.

Для минимизации функционала $Q(f)$ будем использовать градиентный метод, первоначально не обращая внимания на тот факт, что f_K имеет непроизвольные координаты. Положим начальное приближение:

$$f_0 := 0, f_{K,i} := f_{K-1,i} - \eta g_i, i = 1, \dots, l,$$

где $g_i = L'(f_{K-1,i}, y_i)$ — компоненты вектора — градиента.

Будем искать такой алгоритм b_K , чтобы вектор $(b_k(x_i))_{i=1}^l$ приближал вектор антиградиента $(-g_i)_{i=1}^l$:

$$b_K := \arg \max_b \sum_{i=1}^l (b(x_i) + g_i)^2. \quad (3)$$

Шаг (3) отражает главную идею бустинга — последовательное построение композиции алгоритмов, при котором каждый следующий алгоритм стремится компенсировать недостатки композиции всех предыдущих. С помощью градиентного шага минимизируется функционал, в результате чего получается новый базовый алгоритм.

Формальный алгоритм метода представляется в следующем виде:

Вход: обучающая выборка X^l , число итераций K , темп обучения η .

Выход: базовые алгоритмы и их веса $a_k b_k, k = 1, \dots, K$.

Шаг 1. Инициализация $f_i := 0, i = 1, \dots, l$.

Шаг 2. Для всех $k = 1, \dots, K$.

Шаг 3. Найти базовый алгоритм, приближающий антиградиент:

$$b_K := \arg \min_b \sum_{i=1}^l (b(x_i) + L'(f_i, y_i))^2.$$

Шаг 4. Решить задачу одномерной минимизации:

$$a_k := \arg \min_{\eta > 0} \sum_{i=1}^l L'(f_i + \eta b_K(x_i), y_i)^2.$$

Шаг 5. Обновить значения композиции на объёмах выборки

$$f_i := f_i + a_k b_K(x_i), i = 1, \dots, l.$$

Объект из обучающей выборки выбирается случайным образом. Функция потерь задается в виде логарифмической функции. Основными инструментальными средствами для настройки алгоритма SGB являются число используемых базовых алгоритмов и шаг градиентного метода.

Таким образом, сформированные на данном этапе технологии однородные группы заемщиков, имеющих содержательно разные профили риска, дают основание для разработки дифференцированного пакета решений (стратегий) по управлению кредитными рисками банка. Такие стратегии разрабатываются отдельно для каждого из четырех однородных кластеров. Управленческие решения могут быть направлены на текущий контроль, мониторинг и профилактику невозврата кредитов заемщиками.

Заключение

Поставленная в результате исследования цель по формированию подхода, моделей и инструментальных средств для надежной оценки кредитоспособности физических лиц с учетом данных цифровых

следов достигнута. Разработан концептуальный подход и методика оценки кредитоспособности физических лиц, обеспечивающие снижение кредитных рисков банка и повышение его доходности и финансовой устойчивости. В рамках подхода осуществляется выбор и обоснование факторов на основе данных цифрового следа, позволяющих совместно с антропометрическими и финансовыми характеристиками осуществлять комплексную оценку профилей кредитного риска заемщика. Построена модель кластеризации заемщиков, позволяющая выявить однородные группы (кластеры) заемщиков на основании схожести значений финансовых, антропометрических и социально-экономических показателей и факторов цифрового следа, а каждому кластеру удается поставить в соответствие уровень кредитного риска. Предложена модель кластеризации для оценки принадлежности нового потенциального заемщика к одной из групп риска. Получены также перечисленные далее новые результаты.

1. Составлены, экономически и финансово обоснованы новые факторы для комплексной оценки профилей риска заемщиков — физических лиц. Использование данных о заемщиках, собранных на основании их цифровых следов, позволит составить более полный, адекватный портрет заемщика, что в дальнейшем обеспечит банку формирование индивидуальных кредитных траекторий для каждого заемщика и повысит качество выдаваемых кредитов.

2. Предложен новый подход и методика оценки кредитоспособности заемщиков — физических лиц, позволяющие снизить кредитные риски банка и повысить его финансовую устойчивость.

3. Построена модель кластеризации, позволяющая сформировать однородные группы (кластеры) заемщиков, имеющие схожие значения факторов кредитоспособности и модель классификации для оценивания принадлежности нового потенциального заемщика к одному из выявленных кластеров. Применение этих моделей в составе методики оценки кредитоспособности обеспечит повышение эффективности принимаемых финансовых решений.

Практическую значимость имеют результаты апробации предлагаемых теоретических положений, методов и инструментария на фактических данных о деятельности крупного банка РФ. Сравнительный анализ эффективности предлагаемого и традиционного подходов и методик для оценки кредитоспособности заемщиков показали следующие результаты. Совокупный ущерб, связанный с неверной оценкой уровня кредитоспособности и риска по существующей методике, составляет по представленной выборке 275 000 руб., а по всей совокупности заемщиков банка ущерб может достигнуть более 5 млн руб. Единовременные затраты, требуемые на внедрение предлагаемой методики с учетом стоимости программного обеспечения и обучения сотрудников кредитного отдела, составят около 1 млн руб. Чистый доход для банка при внедрении предлагаемого подхода и методики с учетом сопоставления единовременных затрат и годового ущерба, возникающего в результате невозврата в срок кредитов, составит более 4 млн руб. в год.

1. **Principles** for the Management of Credit Risk. Basel Committee on Banking Supervision. Basel. September, 2000. URL: <https://www.bis.org/publ/bcbs75.pdf>

2. **Basel Committee on Banking Supervision** "International Convergence of Capital Measurement and Capital Standards. A Revised Framework". Consultative document, Basel: Bank for International Settlements, 2004. URL: <https://www.bis.org/publ/bcbs128.pdf>

3. **Basel Committee on Banking Supervision** "International regulatory framework for banks", Consultative document, Basel: Bank for International Settlements, 2010. URL: <https://www.bis.org/bcbs/basel3.htm>

4. **Инструкция** Банка России от 29.11.2019 № 199-И (ред. от 26.03.2020) "Об обязательных нормативах и надбавках к нормативам достаточности капитала банков с универсальной лицензией". URL: <https://rudata.info/files/rudata/add-in/documents/199-i.pdf>

5. **Положение** Банка России от 28.06.2017 № 590-П (ред. от 16.10.2019) "О порядке формирования кредитными организациями резервов на возможные потери по ссудам, ссудной и приравненной к ней задолженности". URL: https://cbr.ru/faq_ufr/dbrnfaq/doc/?number=590-%D0%9F

6. **Обзор** российского финансового сектора и финансовых инструментов. Аналитический отчет Банка России. Москва: Банк России, 2021. 76 с. URL: http://www.cbr.ru/collection/collection/file/32168/overview_2020.pdf

7. **Orlova E. V.** Economic Efficiency of the Mechanism for Credit Risk Management // Proceedings of the Workshop on Computer Modelling in Decision Making (CMDM 2017), CEUR Workshop Proceedings. — 2017. — Vol. 2018. — P. 139—150.

8. **Niu B., Ren J., Li X.** Credit Scoring Using Machine Learning by Combining Social Network Information: Evidence from Peer-to-Peer Lending // Information. — 2019. — Vol. 10, No. 12. — Article 397.

9. **Lavrushin O. I.** The development of the banking sector and its infrastructure in the Russian economy. М.: KNORUS, 2017. — 176 p.

10. **Tobin P., Brown A.** Estimation of Liquidity Risk in Banking // ANZIAM Journal. — 2004. — Vol. 45. — P. 519—533.

11. **Банкова К. В.** Использование скоринговых моделей для оценки кредитоспособности заемщиков в России // Известия Академии управления: теория, стратегии, инновации. — 2011. — № 4. — С. 14—16.

12. **Глинкина Е. В.** Кредитный скоринг как инструмент эффективной оценки кредитоспособности // Финансы и кредит. — 2011. — № 16. — С. 43—47.

13. **Лебедев Е. А.** Синтез скоринговой модели методом системно-когнитивного анализа // Политематический сетевой электронный научный журнал Кубанского государственного аграрного университета. — 2007. — № 29. — С. 17—30.

14. **Макаренко Т. М.** Сочетание сценарного прогнозирования с процедурами динамического ранжирования экспертов при оценке кредитного риска заемщика — физического лица в банке // Вестник Ленинградского государственного университета им. А. С. Пушкина. — 2012. — Том 6, № 3. — С. 56—63.

15. **Crone S. F., Finlay S.** Instance Sampling in Credit Scoring: an Empirical Study of Sample Size and Balancing // International Journal of Forecasting. — 2012. — Vol. 28. — P. 224—238.

16. **Crook J. N., Edelman D. B., Thomas L. C.** Recent Developments in Consumer credit Risk Assessment // European Journal of Operational Research. — 2007. — Vol. 3. — P. 1447—1465.

17. **Mircea G., Pirtea M., Neamtu M., Bazavan S.** Discriminant Analysis in a Credit Scoring Model // Recent Advances in Applied & Biomedical Informatics and Computational Engineering in Systems Applications. — 2011. — Vol. 2. — P. 56—69.

18. **Ong C., Huang J., Tzeng G.** Building Credit Scoring Models Using Genetic Programming // Expert Systems with Applications. — 2005. — Vol. 9. — P. 41—47.

19. **Aebi V., Sabato G., Schmid M.** Risk management, corporate governance, and bank performance in the financial crisis // Journal of Banking and Finance. — 2012. — Vol. 12. — P. 3213—3226.

20. **Berger A. N., Sedunov J.** Bank liquidity creation and real economic output // Journal of Banking and Finance. — 2017. — Vol. 81. — P. 3213—3226.

21. **Caporale G. M., Cerratot M., Xuan Zhang.** Analyzing the Determinants of Insolvency Risk for General Insurance Firms in the UK // Journal of Banking and Finance. — 2017. — Vol. 84. — P. 107—122. URL: <http://www.sciencedirect.com/science/article/pii/S0378426617301711>

22. **Basulin M. A.** Analysis Software "Sas Credit Scoring" for the Commercial Bank // Innovative Information Technologies. — 2013. — Vol. 2. — P. 32–37.
23. **Orlova E. V.** The AI Model for Identification the Impact of Irrational Factors on the Investor's Risk Propensity // Proc. of the 30th International Business Information Management Association Conference (IBIMA). Vision 2020: Sustainable Economic development, Innovation Management, and Global Growth. — Madrid, 2017. — P. 713–721.
24. **Orlova E. V.** Mechanism for Credit Risk Management. Proceedings of the 30th International Business Information Management Association Conference (IBIMA). Vision 2020: Sustainable Economic development, Innovation Management, and Global Growth. — Madrid, 2017. — P. 827–837.
25. **Bhatore S., Mohan L., Reddy Y. R.** Machine learning techniques for credit risk evaluation: a systematic literature review // J BANK FINANC TECHNOL. — 2020. — Vol. 4. — P. 111–138.
26. **Machine Learning for Asset Management: New Developments and Financial Applications.** ISTE Ltd, 2020. URL: <https://onlinelibrary.wiley.com/doi/book/10.1002/9781119751182>
27. **Orlova E. V.** Decision-Making Techniques for Credit Resource Management Using Machine Learning and Optimization // Information. — 2020. — Vol. 11. — Iss. 3. — Article 144.
28. **Maheswari P., Narayana C. V.** Predictions of Loan Defaulter — A Data Science Perspective // Proceedings of the 5th International Conference on Computing, Communication and Security (ICCCS), India, Patna, 2020. — P. 1–4. DOI: 10.1109/ICCCS49678.2020.9277458.
29. **Sivasree M. S.** Loan Credibility Prediction System Based on Decision Tree Algorithm // Int. J. Eng. Res. Technol. — 2015. — Vol. 4, Iss. 09. — P. 825–830.
30. **Krichene A.** Using a naive Bayesian classifier methodology for loan risk assessment // Journal of Economics Finance and Administrative Science. — 2017. — Vol. 22, No. 42. — P. 3–24.
31. **Namvar A., Siami M., Rabhi F., Naderpour M.** Credit risk prediction in an imbalanced social lending environment // International Journal of Computational Intelligence Systems. — 2018. — Vol. 11. — P. 925–935.
32. **Sudhamathy G.** Credit Risk Analysis and Prediction Modelling of Bank Loans Using R // International Journal of Engineering and Technology. — 2016. — Vol. 8. — P. 1954–1966.
33. **Orlova E. V.** Model for Operational Optimal Control of Financial Recourses Distribution in a Company // Computer Research and Modeling. — 2019. — No. 2. — P. 343–358.
34. **Orlova E. V.** Technology for Control an Efficiency in Production and Economic System // Proceedings of the 30th International Business Information Management Association Conference (IBIMA). Vision 2020: Sustainable Economic development, Innovation Management, and Global Growth. — Madrid, 2017. — P. 811–818.
35. **Orlova E. V.** Synergetic Approach for the Coordinated Control in Production and Economic System // Proceedings of the 30th International Business Information Management Association Conference (IBIMA). Vision 2020: Sustainable Economic development, Innovation Management, and Global Growth. — Madrid, 2017. — P. 704–712.
36. **Бондарик А.** Платформа для работы в условиях неопределенности // Открытые системы. СУБД. — 2021. — № 2. — С. 13–17.
37. **Прозоров А, Шнырев Р., Волков В.** Архитектура цифровых платформ будущего // Открытые системы. СУБД. — 2021. — № 2. URL: <https://www.osp.ru/os/2021/02/13055934>
38. **Jinchao Ji, Wei Pang, Chunguang Zhou, Xiao Han, Zhe Wang.** A fuzzy k-prototype clustering algorithm for mixed numeric and categorical data // Knowledge-Based Systems. — 2012. — Vol. 30. — P. 129–135.
39. **Friedman J.** Stochastic Gradient Boosting // Computational Statistics and Data Analysis. — 1999. — Vol. 38. — P. 367–378.

Predictive Credit Risk Analytics Using Borrowers' Digital Footprint and Methods of Statistical Machine Learning

E. V. Orlova, ekorl@mail.ru, Ufa State Aviation Technical University, Ufa, 450008, Russian Federation

Corresponding author:

Orlova Ekaterina V., Professor, Ufa State Aviation Technical University, Ufa, 450008, Russian Federation
E-mail: ekorl@mail.ru

*Received on May 20, 2021
Accepted on June 21, 2021*

The article considers the problem of reducing the bank's credit risks associated with the insolvency of borrowers — individuals using financial, socio-economic factors and additional data about borrowers' digital footprint. A critical analysis of existing approaches, methods and models in this area has been carried out and a number of significant shortcomings identified that limit their application. There is no comprehensive approach to identifying a borrower's creditworthiness based on information, including data from social networks and search engines.

The new methodological approach for assessing the borrowers' risk profile based on the phased processing of quantitative and qualitative data and modeling using methods of statistical analysis and machine learning is proposed. Machine learning methods are supposed to solve clustering and classification problems. They allow to automatically determine the data structure and make decisions through flexible and local training on the data. The method of hierarchical clustering and the k-means method are used to identify similar social, anthropometric and financial indicators, as well as indicators characterizing the digital footprint of borrowers, and to determine the borrowers' risk profile over group. The obtained homogeneous groups of borrowers with a unique risk profile are further used for detailed data analysis in the predictive classification model. The classification model is based on the stochastic gradient boosting method to predict the risk profile of a potential borrower. The suggested approach for individuals' creditworthiness assessing will reduce the bank's credit risks, increase its stability and profitability.

The implementation results are of practical importance. Comparative analysis of the effectiveness of the existing and the proposed methodology for assessing credit risk showed that the new methodology provides predictive analytics of heterogeneous information about a potential borrower and the accuracy of analytics is higher. The proposed techniques are the core for the decision support system for justification of individuals' credit conditions, minimizing the aggregate credit risks.

Keywords: data analytics, machine learning, clustering, classification, creditworthiness, digital footprint, reduction of credit risks

For citation:

Orlova E. V. Predictive Credit Risk Analytics Using Borrowers' Digital Footprint and Methods of Statistical Machine Learning, *Programmnyaya Ingeneriya*, 2021, vol. 12, no. 7, pp. 358—372.

DOI: 10.17587/prin.12.358-372

References

1. **Principles** for the Management of Credit Risk. Basel Committee on Banking Supervision. Basel. September, 2000, available at: <https://www.bis.org/publ/bcbs75.pdf>
2. **Basel Committee on Banking Supervision** "International Convergence of Capital Measurement and Capital Standards. A Revised Framework". Consultative document, Basel: Bank for International Settlements, 2004, available at: <https://www.bis.org/publ/bcbs128.pdf>
3. **Basel Committee on Banking Supervision** "International regulatory framework for banks", Consultative document, Basel: Bank for International Settlements, 2010, available at: <https://www.bis.org/bcbs/basel3.htm>
4. **Instruction** of the Bank of Russia "On mandatory ratios and surcharges to capital adequacy ratios for banks with a universal license", Moscow, Bank of Russia, 2019, available at: <https://rudata.info/files/rudata/add-in/documents/199-i.pdf> (in Russian).
5. **Provision** on the procedure for the formation by credit organizations of reserves for possible losses on loans, loan debt and equivalent debt, available at: https://cbr.ru/faq_ufr/dbrnfaq/doc/?number=590-%D0%9F (in Russian).
6. **Review** of the Russian financial sector and financial instruments. Analytical report of the Bank of Russia. Moscow, Bank of Russia, 2021, 76 p., available at: http://www.cbr.ru/collection/collection/file/32168/overview_2020.pdf (in Russian).
7. **Orlova E. V.** Economic Efficiency of the Mechanism for Credit Risk Management, *Proceedings of the Workshop on Computer Modelling in Decision Making (CMDM 2017), CEUR Workshop Proceedings*, 2017, vol. 2018, pp. 139—150.
8. **Niu B., Ren J., Li X.** Credit Scoring Using Machine Learning by Combining Social Network Information: Evidence from Peer-to-Peer Lending, *Information*, 2019, vol. 10, no. 12, article 397.
9. **Lavrushin O. I.** *The development of the banking sector and its infrastructure in the Russian economy*. Moscow (KNORUS), 2017, 176 p.
10. **Tobin P., Brown A.** Estimation of Liquidity Risk in Banking, *ANZIAM Journal*, 2004, vol. 45, pp. 519—533.
11. **Bankova V. K.** Scoring Models to Assess the Creditworthiness of Borrowers in Russia, *Izvestiya of the Academy of Management: Theory, Strategy, Innovation*, 2011, vol. 4, pp. 14—16 (in Russian).
12. **Glinkina E. V.** Credit Scoring as a Tool for Effective Credit Assessment, *Finance and Credit*, 2011, vol. 16, pp. 43—47 (in Russian).
13. **Lebedev E. A.** Synthesis of Scoring Models Method of Systemic-Cognitive Analysis, *Polythematic Network Electronic Scientific Journal of the Kuban State Agrarian University*, 2007, vol. 29, pp. 17—30 (in Russian).
14. **Makarenko T. M.** The Combination of Scenario Forecasting Procedures with the Dynamic Ranking of Experts when Assessing the Credit Risk of the Borrower — Physical Persons in the Bank, *Bulletin of Leningrad state University of A. S. Pushkin*, 2012, vol. 3, pp. 56—63 (in Russian).
15. **Crone S. F., Finlay S.** Instance Sampling in Credit Scoring: an Empirical Study of Sample Size and Balancing, *International Journal of Forecasting*, 2012, vol. 28, pp. 224—238.
16. **Crook J. N., Edelman D. B., Thomas L. C.** Recent Developments in Consumer credit Risk Assessment, *European Journal of Operational Research*, 2007, vol. 3, pp. 1447—1465.
17. **Mircea G., Pirtea M., Neamtu M., Băzavan S.** Discriminant Analysis in a Credit Scoring Model, *Recent Advances in Applied & Biomedical Informatics and Computational Engineering in Systems Applications*, 2011, vol. 2, pp. 56—69.
18. **Ong C., Huang J., Tzeng G.** Building Credit Scoring Models Using Genetic Programming, *Expert Systems with Applications*, 2005, vol. 9, pp. 41—47.
19. **Aebi V., Sabato G., Schmid M.** Risk management, corporate governance, and bank performance in the financial crisis, *Journal of Banking and Finance*, 2012, vol. 12, pp. 3213—3226.
20. **Berger A. N., Sedunov J.** Bank liquidity creation and real economic output, *Journal of Banking and Finance*, 2017, vol. 81, pp. 3213—3226.
21. **Caporale G. M., Cerratot M., Xuan Zhang.** Analyzing the Determinants of Insolvency Risk for General Insurance Firms in the UK, *Journal of Banking and Finance*, 2017, vol. 84, pp. 107—122, available at: <http://www.sciencedirect.com/science/article/pii/S0378426617301711>
22. **Basulin M. A.** Analysis Software "Sas Credit Scoring" for the Commercial Bank, *Innovative Information Technologies*, 2013, vol. 2, pp. 32—37.
23. **Orlova E. V.** The AI Model for Identification the Impact of Irrational Factors on the Investor's Risk Propensity, *Proc. of the 30th International Business Information Management Association Conference (IBIMA). Vision 2020: Sustainable Economic development, Innovation Management, and Global Growth*, Madrid, 2017, pp. 713—721.
24. **Orlova E. V.** Mechanism for Credit Risk Management, *Proceedings of the 30th International Business Information Management Association Conference (IBIMA). Vision 2020: Sustainable Economic development, Innovation Management, and Global Growth*, Madrid, 2017, pp. 827—837.
25. **Bhatore S., Mohan L., Reddy Y. R.** Machine learning techniques for credit risk evaluation: a systematic literature review, *J BANK FINANC TECHNOL*, 2020, vol. 4, pp. 111—138.
26. **Machine Learning for Asset Management: New Developments and Financial Applications**. ISTE Ltd, 2020, available at: <https://onlinelibrary.wiley.com/doi/book/10.1002/9781119751182>
27. **Orlova E. V.** Decision-Making Techniques for Credit Resource Management Using Machine Learning and Optimization, *Information*, 2020, vol. 11, iss. 3, article 144.
28. **Maheswari P., Narayana C. V.** Predictions of Loan Defaulter — A Data Science Perspective, *Proceedings of the 5th International Conference on Computing, Communication and Security (ICCCS)*, India, Patna, 2020, pp. 1-4, DOI: 10.1109/ICCCS49678.2020.9277458.
29. **Sivasree M. S.** Loan Credibility Prediction System Based on Decision Tree Algorithm, *Int. J. Eng. Res. Technol*, 2015, vol. 4, iss. 09, pp. 825—830.
30. **Krichene A.** Using a naive Bayesian classifier methodology for loan risk assessment, *Journal of Economics Finance and Administrative Science*, 2017, vol. 22, no. 42, pp. 3—24.
31. **Namvar A., Siami M., Rabhi F., Naderpour M.** Credit risk prediction in an imbalanced social lending environment, *International Journal of Computational Intelligence Systems*, 2018, vol. 11, pp. 925—935.
32. **Sudhamathy G.** Credit Risk Analysis and Prediction Modelling of Bank Loans Using R. *International Journal of Engineering and Technology*, 2016, vol. 8, pp. 1954—1966.
33. **Orlova E. V.** Model for Operational Optimal Control of Financial Recourses Distribution in a Company, *Computer Research and Modeling*, 2019, no. 2, pp. 343—358.
34. **Orlova E. V.** Technology for Control an Efficiency in Production and Economic System, *Proceedings of the 30th International Business Information Management Association Conference (IBIMA). Vision 2020: Sustainable Economic development, Innovation Management, and Global Growth*, Madrid, 2017, pp. 811—818.
35. **Orlova E. V.** Synergetic Approach for the Coordinated Control in Production and Economic System, *Proceedings of the 30th International Business Information Management Association Conference (IBIMA). Vision 2020: Sustainable Economic development, Innovation Management, and Global Growth*, Madrid, 2017, pp. 704—712.
36. **Bondarik A.** Platform for work in conditions of uncertainty, *Open systems. DBMS*, 2021, no. 2, pp. 13—17.
37. **Prozorov A., Shnyrev R., Volkov V.** Architecture of digital platforms of the future, *Open systems. DBMS*, 2021, no. 2, available at: <https://www.osp.ru/os/2021/02/13055934>
38. **Jinchao Ji, Wei Pang, Chunguang Zhou, Xiao Han, Zhe Wang.** A fuzzy k-prototype clustering algorithm for mixed numeric and categorical data, *Knowledge-Based Systems*, 2012, vol. 30, pp. 129—135.
39. **Friedman J.** Stochastic Gradient Boosting, *Computational Statistics and Data Analysis*, 1999, vol. 38, pp. 367—378.

Д. Д. Рухович, аспирант, daniel-rukhovich@yandex.ru, Московский государственный университет имени М. В. Ломоносова

Проекция изображений в пространство для монокулярной и мультиракурсной трехмерной детекции объектов в уличных сценах

Предложен новый метод трехмерной детекции объектов. В отличие от существующих методов аналогичного назначения, предложенный метод не опосредует задачу трехмерной детекции через задачу детекции объектов на изображении, работая сразу с трехмерными представлениями. Предложенный метод превосходит все существующие методы трехмерной детекции объектов по монокулярным изображениям на наборах данных KITTI и NuScenes, содержащих городские сцены.

В настоящей работе впервые представлена мультиракурсная постановка задачи трехмерной детекции, когда задача детекции объектов решается на основе нескольких изображений статичной сцены с различных ракурсов, также предложен способ решения этой задачи.

Предложенный метод детекции объектов работает в режиме реального времени, что позволяет использовать его в составе систем навигации автономных мобильных устройств без ограничений. Результаты настоящего исследования могут быть применены при решении различных задач, в частности, при навигации робототехнического устройства на сцене в режиме реального времени, при планировании траектории движения, а также при построении трехмерной реконструкции сцен с нанесенной семантической разметкой объектов.

Ключевые слова: машинное обучение, глубокое обучение, трехмерная детекция объектов

Введение

Процедура детекции в контексте настоящей статьи заключается в обнаружении объектов, классификации объектов по наперед заданным категориям и определении местоположения и ориентации этих объектов. Благодаря последним разработкам в области визуальной детекции [1–4] задача обнаружения объектов на изображении может быть решена с высокой точностью даже на мобильном устройстве в условиях ограниченных вычислительных ресурсов.

Трехмерная детекция объектов [5–7] является более сложной задачей, поскольку по сравнению с изображением трехмерная сцена задается большим числом параметров, содержит больше неопределенности и, соответственно, является более структурно сложным объектом анализа.

Методы трехмерной детекции решают задачу определения расположения объектов в пространстве, а также их геометрических параметров: пространственных размеров и ориентации в пространстве. Ориентация в пространстве, как правило, задается угловыми координатами, а именно азимутальным углом и углом между оптической осью камеры и пространственным вектором, направленным из центра камеры на центр объекта.

Источниками информации об окружающем пространстве могут служить данные, поступающие

с различных датчиков записывающей установки. Это могут быть монокулярные или стереокамеры, лидары, возвращающие облако точек сцены, а также инерциальные датчики (IMU) и системы глобального позиционирования (GPS, GLONASS).

Обычные камеры относительно дешевы, долговечны, имеют малый вес и габаритные размеры. Как следствие, они могут быть легко смонтированы практически в любом месте автономного транспортного средства любого размера, дизайна и назначения. Тем не менее данные с камеры не содержат полную информацию о геометрии сцены, о местоположении объекта в сцене или о его размерах в абсолютных значениях. Стереокамера позволяет оценить расстояния до точек сцены и таким образом восстановить недостающую информацию. Однако не всегда возможно реализовать стереозрение. Например, в беспилотных летательных аппаратах это технически трудноосуществимо, кроме того, установка более чем одной камеры может сделать нерентабельным недорогое робототехническое устройство.

Использование подходов на основе лидар-технологий ограничено возможностями применения лидаров. Лидар является технически сложным и дорогостоящим устройством, имеет короткий срок службы, а его масса зачастую слишком велика для установки на небольшие робототехнические устройства. Как следствие, лидары не являются универсальным

решением и занимают небольшую нишу в области технологий восприятия информации автономных транспортных средств.

Инерциальные навигационные системы имеют в своем составе датчики линейного ускорения (акселерометры) и угловой скорости (гироскопы или пары акселерометров, измеряющих центробежное ускорение). С помощью гироскопов можно определить отклонение связанной с корпусом прибора системы координат от системы координат, связанной с Землей, получив при этом углы ориентации — рыскание (курс), тангаж и крен. Акселерометр, в свою очередь, измеряет угловое отклонение координат, а именно широту, долготу и высоту. Высокоточные инерциальные датчики используются в системах навигации самолетов и ракет. Оснащение подобными датчиками беспилотных автомобилей, дронов или мобильных робототехнических устройств не представляется возможным вследствие их больших габаритных размеров и массы, а также высокой стоимости. В то же время мобильные инерциальные датчики компактны и универсальны, однако они отличаются низкой точностью.

Системы глобального позиционирования позволяют быстро и надежно определить приблизительное местоположение объекта на поверхности земли по сигналам от искусственных спутников Земли. Системы глобального позиционирования широко используются для локализации и навигации различных транспортных средств, а также для составления пеших маршрутов. Однако погрешность оценки расположения объекта измеряется в метрах, что делает системы глобального позиционирования непригодными в качестве основного источника данных для высокоточных систем управления.

Из сказанного выше следует, что монокулярные камеры имеют ряд преимуществ по отношению к прочим датчикам в качестве базовой технологии восприятия информации о среде. Программное решение на основе изображений пригодно для внедрения в качестве компонента программного обеспечения робототехнического устройства, работающего как снаружи, так и внутри помещений.

Однако информация о сцене и находящихся в ней объектах, которая может быть извлечена из изображений с монокулярной камеры, по умолчанию является неполной. Соответственно, для решения задачи трехмерной детекции по изображению с монокулярной камеры требуются более сложные алгоритмы, позволяющие дополнить частичную информацию о сцене на основе априорных знаний. Однако повышение сложности метода — увеличение числа обучаемых параметров нейросетевой модели, добавление дополнительных этапов вычислений — неизбежно влечет за собой снижение производительности, что делает невозможным использование подобных методов в устройствах, взаимодействующих со средой в режиме реального времени.

К моменту написания настоящей работы не существовало достаточно точного и в то же время простого и вычислительно эффективного нейросетевого метода трехмерной детекции объектов по изображениям с монокулярной камеры.

Многие современные нейросетевые методы трехмерной детекции объектов устроены следующим образом: сначала они генерируют множество гипотез о расположении объектов на изображении, а затем предсказывают положение и ориентацию объекта в трехмерном пространстве. В настоящей работе описывается предложенный автором новый нейросетевой метод трехмерной детекции объектов. В отличие от других современных методов трехмерной детекции объектов, предложенный метод не опосредует задачу трехмерной детекции через задачу детекции объектов на изображении, работая сразу с трехмерными представлениями. Хорошие результаты предложенного метода детекции объектов, полученные в ходе экспериментальной проверки, показывают, что этап визуальной детекции объектов на изображении является избыточным и даже вредным для решения целевой задачи трехмерной детекции объектов.

В настоящей работе впервые дана *мультиракурсная* постановка задачи трехмерной детекции, когда задача детекции объектов решается на основе нескольких изображений статичной сцены с различных ракурсов, также представлен метод решения этой задачи.

Предложенный метод детекции объектов работает в режиме реального времени. Это позволяет использовать его в составе систем навигации автономных мобильных устройств без существенных ограничений. Результаты настоящего исследования могут быть применены при решении различных задач, а именно:

- навигации робототехнического устройства на сцене в режиме реального времени;
- планировании траектории движения робототехнического устройства по сцене;
- построении трехмерной реконструкции сцен с нанесенной семантической разметкой объектов.

Постановка задачи трехмерной детекции изложена в разд. 1. Раздел 2 посвящен обзору тематических публикаций. Он содержит классификацию существующих методов трехмерной детекции, а также сравнительный анализ преимуществ и недостатков различных методов. В разд. 3 описан новый нейросетевой метод трехмерной детекции объектов. Раздел 4 содержит описание условий экспериментальной проверки предложенного метода. В разд. 5 приведены результаты экспериментов в виде сводных таблиц, графиков и других способов их визуализации. Было проведено сравнение результатов предложенного метода с результатами других существующих методов, и на этом основании дана оценка предложенному методу.

1. Постановка задачи

Нейросетевой метод трехмерной детекции объектов представляет собой алгоритм, отображающий цифровое трехканальное изображение $I \in R^{w \times h \times 3}$ во множество *ограничивающих параллелепипедов объектов* $B = \{(x_k, y_k, z_k, w_k, l_k, h_k, \theta_k)\}_{k=1}^n$, соответствующих объектам, содержащимся на этом изображении.

Каждый ограничивающий параллелепипед объекта представляет собой трехмерный параллелепи-

пед, который задается координатами центра параллелепипеда (x, y, z) , шириной w , длиной l , высотой h и углом поворота θ относительно оси x . Подразумевается, что ось z направлена перпендикулярно опорной плоскости, на которой размещены объекты (в данном случае — поверхность земли). В свою очередь, опорной плоскости параллельны все ограничивающие параллелепипеды объектов.

В данной работе дополнительно рассматривается постановка задачи, в которой метод трехмерной детекции объектов принимает на вход не одно изображение I , а некоторое множество изображений $\{I_k\}_{k=1}^m$ одной статичной сцены. Далее данная постановка задачи будет называться мультиракурсной.

2. Обзор публикаций по теме

Данный раздел содержит обзор современных нейросетевых методов трехмерных объектов, разбитых на группы в соответствии с набором датчиков, используемых в качестве источника информации о сцене.

2.1. Методы, использующие облака точек

Разреженные облака точек — одно из возможных представлений сцены, совокупность точек сцены в трехмерном пространстве. Облака точек могут быть построены с использованием лидар-технологий.

Методы трехмерной детекции объектов этой группы решают задачу обнаружения объектов посредством преобразования разреженных облаков точек в различные представления. Так, некоторые методы, например [6, 7], проецируют облака точек на плоскость, получая таким образом вид сцены сверху. Далее к полученному двумерному представлению применяется стандартный метод визуальной детекции объектов.

Другие методы, такие как VoxelNet [8], работают с представлением трехмерного пространства в виде трехмерного параллелепипеда вокселей. Для этого область трехмерного пространства, где располагаются наблюдаемые объекты, разбивается на более мелкие области пространства трехмерной сеткой, параллельной координатным осям. Эти области пространства называются *вокселями* (сокр. от англ. *volumetric pixel*, буквально — "объемный пиксел").

2.2. Методы, использующие облака точек и изображения

За последние несколько лет были разработаны различные способы объединения информации с лидаров и камер [9–12]. Эти методы могут быть разделены на перечисленные далее три группы.

Методы первой группы [10, 11] выполняют детекцию объектов на изображении и используют полученные двумерные предсказания, чтобы определить область трехмерного пространства, где, предположительно, находится объект. Из облака точек выбирается подмножество точек, находящихся в заданной области. Это подмножество точек поступает на вход нейросетевой модели, решающей задачу трехмерной детекции на основе облака точек.

Методы второй группы сначала генерируют трехмерные гипотезы о расположении объектов на основе облаков точек. Затем эти гипотезы проецируются на изображение, определяя таким образом область видимости объекта. Из изображения вырезается соответствующий участок, из которого затем извлекаются признаки с помощью нейросетевой модели. Полученные двумерные признаки далее используют для получения финальных предсказаний. К подобным методам можно отнести, например, метод Deep Sliding Shapes [12].

Методы третьей группы, такие как MVXNet [9], одновременно используют как двумерные, так и трехмерные признаки для генерации гипотез о расположении объекта в пространстве, объединяя эти признаки на ранних этапах вычислений внутри нейросетевой модели. Такой подход позволяет использовать максимальное количество информации из разных источников.

2.3. Методы, использующие стереоизображения

В одной из первых работ 3DOP [13], посвященных трехмерной детекции по стереоизображениям, предлагается генерировать трехмерные гипотезы о расположении объектов на основе стереорекострукции, локальных особенностей карт глубины и априорных знаний о размерах объектов. Метод TLNet [14] реализует триангуляцию для объединения гипотез для левого и правого изображений стереопары, полученных с помощью нейросетевой модели. В методе Stereo R-CNN [15] для генерирования гипотез о расположении объекта используются оба изображения стереопары. Далее требуется найти и сопоставить ключевые точки на левом и правом изображениях, чтобы затем путем триангуляции определить область в трехмерном пространстве, в которой находится объект. Наконец, обнаруженный таким образом объект дополнительно выравнивается относительно стереоизображения, что позволяет повысить точность метода.

2.4. Методы, использующие монокулярные изображения

Детекция трехмерных объектов по единственному изображению является наиболее сложной задачей из рассмотренных.

Стандартным подходом к созданию метода трехмерной детекции является добавление к методу визуальной детекции объектов на изображении дополнительной ветви нейросетевой модели, которая оценивает позу объекта в пространстве [16–19]. Так, например, метод Mono3D [16] генерирует трехмерные якоря, а именно — априорные гипотезы о расположении объектов в пространстве, используя семантическую сегментацию изображений и поиск контуров объектов, также формируются дополнительные предположения о расположении объектов внутри сцены. В методе Deep3DBox [17] вводится дискретизация на основе интервалов для оценки локальной ориентации каждого объекта, а также задаются гибкие ограничения на связь между расположением объекта в пространстве и его проекцией на плоскость

изображения. В методе MonoGRNet [18] предлагается декомпозировать задачу детекции на независимые задачи: оценку расстояния до объекта, оценку трехмерного расположения объектов и оценку позы объекта соответственно. Отдельные компоненты затем объединяются и оптимизируются для получения итогового результата. Метод 3D-RCNN [19] также использует трехмерные модели объектов для перевода результатов детекции объектов на изображении в пространство: размер и ориентация трехмерного объекта восстанавливаются через минимизацию ошибки проекции трехмерного объекта на плоскость изображения. Другие методы, такие как [20, 21], переводят гипотезы о расположении объекта на изображении к другим представлениям. Например, метод AM3D [20] реконструирует трехмерное облако точек из изображения с помощью карты глубины, затем нейросетевая модель PointNet [22] используется для оценки трехмерных размеров, местоположения и ориентации объектов в пространстве.

Все перечисленные выше методы решают поставленную задачу в два этапа, что приводит к дополнительной погрешности в финальную оценку. Для снижения этой погрешности в разных методах использовались синтетические данные [23] и дополнительная оптимизация трехмерных оценок расположения объектов с учетом их проекций на плоскость изображения [24].

Едиственный к моменту написания настоящей работы одноэтапный метод трехмерной детекции объектов M3D-RPN [24] одновременно генерирует гипотезы о расположении объектов и на изображении, и в пространстве. Набор гипотез служит входом для нейросетевой модели, а полученные с ее помощью результаты подвергаются дополнительной оптимизации с учетом глубины. Метод OFTNet [25] сопоставляет изображение с двумерной проекцией облака точек на плоскость (вид сцены сверху), используя ортогональное преобразование объектов и регрессируя каждую трехмерную переменную независимо.

Резюмируя представленные выше сведения, можно сделать вывод, что ни один из перечисленных выше методов не может точно оценить расположение объекта в пространстве без промежуточного генерирования гипотезы о расположении объекта на изображении. Кроме этого, ни один из упомянутых методов трехмерной детекции объектов на основе монокулярных изображений не решает задачу в мультиракурсной постановке.

3. Предложенный метод

Предложенный метод монокулярной и мультиракурсной трехмерной детекции объектов ImVoxelNet наследует большинство качеств (свойств) современных методов трехмерной детекции объектов. Так, вначале из изображений с помощью предварительно обученной нейросетевой модели извлекаются двумерные признаки, которые затем отображаются в трехмерное пространство. Полученные таким образом трехмерные признаки подаются в заключительные слои нейросетевой модели, возвращающие итоговые предсказания.

Проецирование двумерных признаков в трехмерный параллелепипед вокселей ранее уже применялось в ряде нейросетевых методов, решающих другие задачи компьютерного зрения. Основным новшеством предложенного метода является применение подобного проецирования для решения задачи обнаружения объектов.

Последовательность преобразований и вычислений, выполняемую в рамках предложенного метода трехмерной детекции объектов, можно условно разделить на четыре этапа (табл. 1). Эти этапы будут подробнее описаны далее.

На рис. 1, см. вторую сторону обложки, изображена принципиальная схема предложенного метода. Разными цветами помечены разные этапы работы нейросетевых методов.

3.1. Извлечение двумерных признаков из изображений

Первый этап работы большинства современных нейросетевых методов детекции объектов заключается в извлечении признаков из изображений с помощью предварительно обученной нейросетевой модели. Для извлечения признаков часто используются модели архитектуры VGG [26], ResNet [27] и др. Предложенный метод использует нейросетевую модель ResNet. В сверточных нейросетевых моделях с прямолинейной структурой, к каким принадлежат в том числе и нейросетевые модели архитектуры ResNet, область видимости сверточных слоев увеличивается от первого слоя к последнему. Соответственно, признаки, вычисленные на начальных слоях, содержат информацию о небольших участках входного изображения, в то время как признаки, вычисленные на последних слоях, содержат

Таблица 1

Этапы работы предложенного метода трехмерной детекции объектов

Этап	Описание	Входные данные	Выходные данные
I	Извлечение двумерных признаков из изображений	Одно или несколько монокулярных изображений	Тензор двумерных признаков для каждого входного изображения
II	Отображение двумерных признаков в параллелепипед вокселей	Один или несколько тензоров двумерных признаков изображений	Параллелепипед вокселей, каждому из которых назначена проекция двумерных признаков
III	Вычисление трехмерных признаков	Параллелепипед вокселей со спроецированными двумерными признаками	Тензор трехмерных признаков
IV	Получение финальных предсказаний	Тензор трехмерных признаков	Набор ограничивающих параллелепипедов

информацию о более крупных областях. Размеры объектов могут значительно различаться, соответственно различаются и занимаемые ими области на изображении. Чтобы эффективно распознавать как большие, так и очень маленькие объекты, необходимо обрабатывать области видимости разного размера. Это можно реализовать путем агрегирования признаков с нескольких слоев нейросетевой модели. Предложенный метод использует признаки с нескольких слоев, являющихся заключительными слоями сверточных блоков, обозначенных синим цветом на рис. 1 (см. вторую сторону обложки). После этих слоев происходит понижение размерности карты признаков, в силу этого именно данное подмножество слоев часто выбирается при извлечении признаков из изображений с помощью нейросетевой модели архитектуры ResNet. Так, признаки, извлеченные из изображения с шириной W и высотой H пикселей, имеют следующие размеры: $W/4 \times H/4 \times c_0$, $W/8 \times H/8 \times 2c_0$, $W/16 \times H/16 \times 4c_0$ и $W/32 \times H/32 \times 8c_0$, где c_0 — число каналов ResNet. Существует несколько версий архитектуры ResNet, отличающихся в том числе и числом каналов. Соответственно, значение c_0 зависит от версии архитектуры ResNet. В предложенном методе применяется наиболее часто используемая версия ResNet-50, для которой c_0 составляет 256.

В ряде методов [1–4], использующих признаки, извлеченные из изображений с помощью нейросетевой модели архитектуры ResNet, для агрегирования этих признаков применяется дополнительная нейросетевая модель FPN (*Feature Pyramid Network*). Модель FPN принимает на вход четыре тензора признаков и возвращает также четыре тензора разного размера, обозначенных зеленым цветом на рис. 1 (см. вторую сторону обложки). Для проекции признаков в трехмерные воксели используется самый большой по пространственным измерениям выходной тензор. Как показано в работе [28], использование других выходных тензоров избыточно, поскольку не приводит к увеличению точности решения целевой задачи. Таким образом, после выполнения этапа I входное изображение оказывается преобразовано в тензор размера $W/4 \times H/4 \times c_1$. Значение параметра c_1 в соответствии с работой [28] выбрано равным 64.

3.2. Отображение двумерных признаков в трехмерный параллелепипед вокселей

Для каждого набора данных может быть определена рабочая область пространства, в которой располагаются видимые объекты. В поле зрения камеры, установленной, например, на беспилотном автомобиле, попадает несколько десятков метров дорожного полотна и некоторое число объектов, находящихся по обе стороны дороги.

До того как задать рабочую область пространства, необходимо установить порядок и взаимное расположение координатных осей. Здесь и далее предполагается, что ось z направлена вверх перпендикулярно поверхности земли или пола, ось x направлена вперед по лучу зрения камеры, ось y перпендикулярна им обеим, составляя с осями x и z правую тройку векторов.

Итак, рабочую область пространства, в которой могут располагаться объекты, можно ограничить параллелепипедом с фиксированными размерами сторон по каждой из осей и известным сдвигом относительно расположения камеры. Данный параллелепипед может быть задан через минимальную и максимальную координаты в метрах по трем осям. Обозначим эти координаты как X_{\min} и X_{\max} , Y_{\min} и Y_{\max} , Z_{\min} и Z_{\max} . Эти значения зависят от типа и свойств присутствующих в наборе данных сцен, потому они уникальны для набора данных.

Каждый воксел имеет одинаковые измерения по трем координатным осям. Обозначим размеры каждого вокселя в метрах как S . Если число вокселей по осям x , y , z составляет N_x , N_y и N_z , то $N_x S = X_{\max} - X_{\min}$, $N_y S = Y_{\max} - Y_{\min}$, $N_z S = Z_{\max} - Z_{\min}$.

В нейросетевых методах трехмерной детекции объектов на основе облаков точек признаков описание каждого вокселя вычисляется по попавшим внутрь него точкам. В случае монокулярного или мультиракурсного входа эта операция менее тривиальна. Необходимо определить отображение из двумерных признаков каждого изображения в параллелепипед вокселей. Такое отображение является стандартным для задач компьютерного зрения; в частности, оно было описано в работах [14, 28]. Приведем далее его краткое описание.

На предыдущем этапе работы метода — этапе I — из изображений были извлечены признаки, составляющие тензор размера $W/4 \times H/4 \times c_1$. Теперь следует отобразить эти признаки в пространство так, чтобы каждому из $N_x \times N_y \times N_z$ вокселей было назначено некоторое признаковое описание. Иными словами, требуется установить соответствие между значениями тензора признаков и вокселями в пространстве. Сверточные нейросетевые слои вычисляют свертки над локальными областями. Таким образом, значение, расположенное выше и левее внутри тензора признаков, будет соответствовать областям входного изображения, также расположенным выше и левее.

Признаки, извлеченные с помощью сверточных нейросетевых моделей, обладают свойством локальности и сохраняют относительную геометрию. Таким образом, тензор признаков можно рассматривать как многоканальное изображение, между которым и входным существует сохраняющее геометрию соответствие. Следовательно, если возможно определить соответствие между пикселями исходного изображения и вокселями в трехмерном пространстве, то возможно и аналогичным образом определить соответствие между признаками изображения и вокселями.

Пусть параметры внутренней калибровки камеры \mathbf{K} имеют вид

$$\mathbf{K} = \begin{bmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix},$$

где f_x и f_y обозначают величину фокусного расстояния камеры по соответствующим осям, а u_0 и v_0 — координаты центра проекции на изображении. При уменьшении разрешения изображения f_x и f_y , u_0

и v_0 кратно уменьшаются. Таким образом, если для исходного изображения матрица параметров была равной \mathbf{K} , то для извлеченных признаков следует использовать матрицу следующего вида:

$$\begin{bmatrix} 1/4 & 0 & 0 \\ 0 & 1/4 & 0 \\ 0 & 0 & 1 \end{bmatrix} [\mathbf{K} \ 0].$$

Тогда центр произвольного вокселя (x, y, z) можно перевести в координаты (u, v) двумерного изображения по формуле

$$z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} 1/4 & 0 & 0 \\ 0 & 1/4 & 0 \\ 0 & 0 & 1 \end{bmatrix} [\mathbf{K} \ 0] \begin{bmatrix} \mathbf{R} & \mathbf{T} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix},$$

где z_c обозначает коэффициент масштаба при переводе из однородных координат; \mathbf{R} — матрица поворота камеры; \mathbf{T} — вектор смещения камеры относительно начала координат. Таким образом, для каждого трехмерного вокселя с центром (x, y, z) , для которого $0 \leq u \leq W/4$ и $0 \leq v \leq H/4$, однозначно определено искомое признаковое описание. Для оставшихся вокселей в качестве признакового описания назначается вектор, состоящий из нулей. В случае мультиракурсного входа некоторым вокселям может быть назначено несколько признаковых описаний, полученных из двумерных признаков нескольких входных изображений. Для таких вокселей итоговое признаковое описание определяется как покоординатное среднее признаковых описаний.

3.3. Получение трехмерных признаков

На предыдущем шаге каждому видимому вокселю было назначено признаковое описание, полученное из двумерных признаков исходных изображений. Однако такого описания недостаточно для решения задачи детекции объектов. Так, в результате отображения признаков изображения в трехмерное пространство всем вокселям, находящимся на одном луче из центра камеры, было назначено одинаковое признаковое описание. Чтобы разрешить эту неоднозначность и получить больше информации об окружающем трехмерном пространстве, необходимо выполнить следующий шаг — получить трехмерные признаки с помощью нейросетевой модели.

На этом этапе необходимо учитывать специфику сцен. Так, при применении метода для детекции автомобилей в уличных сценах можно учитывать, что автомобили имеют похожие размеры и располагаются строго на поверхности земли. Иными словами, вариативность объектов этой категории по пространственным размерам и расположению относительно поверхности земли довольно невелика. По этой причине достаточно отобразить трехмерные признаки [6–9] в один тензор размера $N_x \times N_y \times c_2$. Так же, как предлагается в работах [6–9], значение c_2 выбрано равным 256.

Для извлечения трехмерных признаков используется нейросетевая модель, включающая в себя

трехмерные сверточные слои. Для детекции объектов в уличных сценах используется архитектура, описанная в работе [28], в урезанном виде, представленная на рис. 2, см. вторую сторону обложки.

На рис. 2 (см. вторую сторону обложки) параллелепипедами обозначены четырехмерные выходы трехмерных сверточных слоев. Прямоугольниками, как и на рис. 1 (см. вторую сторону обложки), обозначают трехмерные выходы двумерных сверточных слоев. Каждый желтый параллелепипед содержит два трехмерных сверточных слоя и остаточную связь (*residual connection*) входа блока с его выходом. За каждым сверточным слоем на этапе получения трехмерных признаков следуют слои нормализации и активации.

3.4. Получение итоговых предсказаний

Для получения финальных предсказаний используется простая нейросетевая модель, как в работах [6–9], оценивающая координату центра объекта, пространственные размеры по трем осям и угол поворота.

Последним шагом получения итоговых предсказаний, как и при решении задачи детекции объектов на изображении [1–4], является алгоритм подавления немаксимумов. Для этого ограничивающие параллелепипеды обнаруженных объектов проецируются на опорную плоскость — поверхность земли или пола. Далее алгоритм подавления немаксимумов применяется к проекции из вида сверху.

4. Эксперименты

Экспериментальная проверка была проведена на данных, моделирующих сценарий использования метода трехмерной детекции в составе системы навигации беспилотного автомобиля, а именно городских и природных сценах, снятых одной или несколькими камерами, установленными на автомобиле.

4.1. Наборы данных

4.1.3. KITTI

Набор данных KITTI [29], предназначенный для обучения и тестирования методов трехмерной детекции объектов, представляет собой размеченные видеопоследовательности, снятые на камеру на крыше автомобиля. Таким образом, KITTI содержит городские сцены с эталонной разметкой различных участников дорожного движения: автомобилей, пешеходов и велосипедистов.

Обучающая часть состоит из 3711 отдельных изображений, валидационная — из 3768, в то время как остальные 7518 изображений набора данных предназначены для тестирования. Каждому кадру соответствует облако точек, полученное с помощью лидара. Итого весь набор данных содержит 80 256 размеченных объектов. Все объекты разбиты по трем категориям в зависимости от уровня сложности:

- в категорию *простых* определены объекты, высота которых на изображении составляет не менее 40 пикселей, между ними и камерой нет других объектов, и площадь участков объекта, попавших в кадр, составляет не менее 85 % от всей площади объекта;

• к умеренно сложным объектам относятся отчасти видимые объекты высотой не менее 25 пикселей, которые попадают в кадр на 70 % или больше;

• сложные объекты имеют высоту не менее 25 пикселей, на изображении загорожены другими объектами и попадают в кадр как минимум на 50 %.

Все методы трехмерной детекции оцениваются по точности, продемонстрированной на умеренно сложной выборке объектов.

4.1.4. nuScenes

Набор данных nuScenes [30] представляет собой специализированный набор данных для обучения и тестирования компонентов системы управления беспилотными автомобилями. Набор nuScenes содержит облака точек, полученные с помощью лидара, изображения, снятые шестью разнонаправленными камерами, расположенными на бортах автомобиля, а также измерения инерциального сенсора и GPS. Набор данных покрывает 1000 видеопоследовательностей, каждая из которых записывалась на протяжении 20 с, что составляет 1,4 млн изображений и 390 тыс. уникальных облаков точек с лидаров. Все изображения отнесены либо к обучающей части, покрывающей 28 130 сцен, либо к валидационной части, содержащей 6019 сцен соответственно. Выполненная вручную эталонная разметка содержит 1,4 млн объектов, разбитых на 23 категории. Для каждого объекта указаны не только его категория, расположение и ориентация в пространстве, но и степень видимости. Поскольку набор данных содержит изображения с шести камер, можно говорить, что он предназначен для решения задачи мультиракурсной трехмерной детекции. Точность мультиракурсной трехмерной детекции объектов измеряется только на объектах категории "автомобили".

4.2. Оценка точности предсказаний (метрики качества)

Для оценки качества работы методов трехмерной детекции объектов так же, как и для методов детекции объектов на изображении, используется метрика, называемая *средней точностью* (AP, *average precision*). Для большей ясности в случае трехмерной детекции будем обозначать ее AP_{3D} .

На наборе данных nuScenes для определения корректности найденного объекта используется расстояние между центрами этих параллелепипедов. Соответственно, на наборе данных nuScenes значение метрики AP_{3D} может быть вычислено при различных порогах на расстояние в метрах; расстояние будет указываться после символа @. Так, например, $AP_{3D}@2m$ означает, что на расстояние между центрами предсказанного и истинного ограничивающих параллелепипедов объекта установлен порог в 2 м. Все предсказанные объекты, для которых расстояние составляет более 2 м, не учитываются при вычислении метрики $AP_{3D}@2m$.

Для набора данных KITTI принято указывать значение метрики AP_{2D} . Эта метрика вычисляется как средняя точность детекции проекций объектов на опорную плоскость (вид сверху).

4.3. Протокол экспериментов

4.3.1. Предварительная обработка данных

Изображения из набора данных KITTI масштабируются до размера 1280 на 384 пикселя, как и в работах [5, 24]. Изображения набора данных nuScenes остаются в оригинальном разрешении 1600 на 900 пикселей.

На этапе тестирования не используются аугментации изображений. На этапе обучения используются две аугментации: случайное горизонтальное отражение и случайное изменение размера в пределах 25 %. Реализация протоколов экспериментов на всех наборах данных взята из библиотеки MMDetection3D.

4.3.2. Процедура обучения

Задача детекции объектов заключается в предсказании расположения объекта (область значений целевой переменной непрерывна, соответственно, решается задача регрессии) и определении категории объекта (дискретная целевая переменная принимает значение из наперед заданного ограниченного множества, следовательно, решается задача классификации). Для обучения используется составная функция потерь, представляющая собой взвешенную сумму штрафов за ошибку регрессии и ошибку классификации. В данном случае общий штраф складывается из абсолютного отклонения от истинных значений (L1) для задачи регрессии и значения фокальной функции потерь (*focal loss*) для задачи классификации. В случае работы внутри помещений метод детекции обучается оценивать расстояние от камеры до центра объекта. При обучении в качестве функции потерь используется перекрестная энтропия.

Предложенная модель ImVoxelNet на всех наборах данных обучается методом оптимизации Adam с начальной скоростью обучения 0,0001. Согласно алгоритму обучения, установленному по умолчанию в библиотеке MMDetection [31], при обучении на протяжении N эпох скорость обучения уменьшается в 10 раз после $[8N/12]$ -й эпохи и $[11N/12]$ -й эпохи. Используемые наборы данных различаются по объему данных, числу категорий и степени вариативности внешнего вида, расположения и размеров объектов. Соответственно, для разных наборов данных используется различное число эпох обучения. Так, на наборах данных nuScenes число эпох обучения равно 12, а на KITTI — 36.

4.3.3. Детали реализации

Программный код для проведения всех экспериментов по обнаружению объектов реализован на языке Python на основе библиотеки MMDetection3D [31], которая, в свою очередь, использует инструментальные средства библиотеки глубокого обучения PyTorch [32]. Библиотека MMDetection3D содержит имплементации более десятка одно- и двустадийных нейросетевых моделей обнаружения объектов, имеет модульный дизайн, позволяющий легко модифицировать отдельные компоненты нейросетевых методов детекции. Большая часть гиперпараметров инициализирована значениями по умолчанию из MMDetection3D. В качестве предварительно обучен-

Размеры рабочей области пространства для разных наборов данных

Набор данных	X_{\min} , м	X_{\max} , м	Y_{\min} , м	Y_{\max} , м	Z_{\min} , м	Z_{\max} , м	N_x	N_y	N_z	S , м
KITTI	-39,68	39,68	0	69,12	-2,92	0,92	248	216	12	0,32
nuScenes	-50	50	-50	50	-3	1	200	200	8	0,5

ной базовой нейросетевой модели для извлечения признаков используется ResNet-50.

Размеры параллелепипеда вокселей, получаемого на этапе II, задаются константными, но зависящими от специфики набора данных. В соответствии со стандартным протоколом [6–9] размеры рабочей области пространства зависят от состава записываемых лидаров, использовавшихся при сборе данных. Так, набор данных KITTI записан с помощью лидара с максимальной дальностью 70 м и единственной камеры, ориентированной по направлению движения автомобиля. Таким образом, несмотря на большое покрытие сцены лидаром, визуальная информация доступна только для участка облака точек с лидара. Соответственно, рабочая область пространства представляет собой параллелепипед, включающий в себя участок дорожного полотна впереди автомобиля. В состав записываемой установки nuScenes входит более слабый лидар с дальностью порядка 50 м и шесть камер, позволяющих получить полный обзор сцены. Соответственно, рабочая область пространства имеет схожие размеры по всем трем координатным осям. Точные значения размеров рабочей

области пространства для разных наборов данных приведены в табл. 2.

Для обучения и тестирования нейросетевых моделей были задействованы восемь графических процессоров NVidia P40.

5. Результаты

В настоящем разделе приведены результаты экспериментов в виде сводных таблиц и проведен анализ полученных результатов.

5.3. Результаты на KITTI

Набор данных KITTI содержит размеченные объекты нескольких категорий, однако для оценки точности методов трехмерной детекции объектов используются объекты категории "автомобиль". Ключевым показателем считается точность детекции, продемонстрированная на умеренно сложных объектах.

Результаты экспериментов по трехмерной детекции объектов на основе монокулярных изображений из набора данных KITTI приведены в табл. 3 и 4. Здесь и далее в таблицах лучшие результаты выделены полужирным шрифтом.

Таблица 3

Результаты экспериментов по монокулярной трехмерной детекции на наборе данных KITTI. В данной таблице приведены значения метрики $AP_{3D}@70$ для объектов категории "автомобиль". Все приведенные величины являются безразмерными

Метод	Объекты по уровню сложности					
	Простые		Умеренно сложные		Сложные	
	Валидация	Тест	Валидация	Тест	Валидация	Тест
GS3D [33]	13,46	4,47	10,97	2,90	10,38	2,47
MonoPSR [21]	12,75	10,76	11,48	7,25	8,59	5,85
MonoGRNet [18]	13,88	9,61	10,19	5,74	7,62	4,25
SS3D [34]	14,52	10,78	13,15	7,68	11,85	6,51
MonoDIS [35]	18,05	10,37	14,98	7,94	13,42	6,40
MonoPair [36]	—	13,04	—	9,99	—	8,65
SMOKE [5]	14,76	14,03	12,85	9,76	11,50	7,84
M3D-RPN [24]	20,27	14,76	17,06	9,71	15,21	7,42
RTM3D [37]	20,77	14,41	16,86	10,34	16,63	8,77
ImVoxelNet	24,54	17,15	17,80	10,97	15,67	9,15

Результаты экспериментов по монокулярной трехмерной детекции на наборе данных KITTI. В данной таблице приведены значения метрики $AP_{2D}@70$ для объектов категории "автомобиль"

Метод	Объекты по уровню сложности					
	Простые		Умеренно сложные		Сложные	
	Валидация	Тест	Валидация	Тест	Валидация	Тест
GS3D [33]	—	8,41	—	6,08	—	4,94
MonoPSR [21]	20,63	18,33	18,67	12,58	14,45	9,91
MonoGRNet [18]	—	18,19	—	11,17	—	8,73
SS3D [34]	—	16,33	—	11,52	—	9,93
MonoDIS [35]	24,26	17,32	18,43	13,19	16,95	11,12
MonoPair [36]	—	19,28	—	14,83	—	12,89
SMOKE [5]	19,99	20,83	15,61	14,49	15,28	12,75
M3D-RPN [24]	25,94	21,02	21,18	13,67	17,90	10,23
RTM3D [37]	25,56	19,17	22,12	14,20	20,91	11,99
ImVoxelNet	31,67	25,19	23,68	16,37	19,73	13,58

Согласно значениям метрики $AP_{3D}@70$ метод ImVoxelNet обнаруживает простые объекты с большей точностью по сравнению с другими методами трехмерной детекции на основе монокулярных изображений. Также он показывает лучшие результаты по детекции умеренно сложных объектов на тестовой части набора данных KITTI, что является ключевым показателем.

Согласно значениям метрики $AP_{2D}@70$ метод ImVoxelNet значительно опережает конкурирующие методы в обнаружении простых объектов. Он также показывает лучшие результаты по детекции умеренно сложных объектов на тестовой части набора данных KITTI, что является ключевым показателем. Кроме того, он превосходит конкурирующие методы по точности детекции сложных объектов на тестовой части набора данных KITTI.

На рис. 3 (см. вторую сторону обложки) визуализированы предсказания предложенного метода на изображениях из валидационной части набора данных KITTI.

5.4. Результаты на nuScenes

На наборе данных nuScenes метод ImVoxelNet лидирует среди методов трехмерной детекции на основе монокулярных изображений согласно значению ключевой метрики AP (табл. 5). Также предложенный метод превосходит конкурирующие методы по метрикам $AP@0.5m$ и $AP@1m$, при этом уступая методу MonoDIS по метрикам $AP@2m$ и $AP@4m$. Эти результаты можно интерпретировать следующим образом: предложенный метод дает меньше ошибочных предсказаний, т. е. его предсказания отличаются большей точностью и меньшей полнотой. Скорее всего объект окажется не обнару-

Таблица 5

Результаты экспериментов по монокулярной трехмерной детекции на наборе данных nuScenes. Все приведенные величины являются безразмерными

Метод	$AP@0.5m$	$AP@1m$	$AP@2m$	$AP@4m$	AP
OFTNet [25]	—	—	27,0	—	—
CenterNet [38, 39]	—	—	—	—	48,4
MonoDIS [35]	10,7	37,5	69,0	85,7	50,7
ImVoxelNet	18,0	44,2	65,6	77,0	51,2

жен предложенным методом, чем обнаружен с большой погрешностью в оценке расположения.

На рис. 4 (см. третью сторону обложки) визуализированы предсказания предложенного метода на шести изображениях из валидационной части набора данных nuScenes, составляющих мультиракурсный вход.

Заключение

Представлен новый метод монокулярной и мультиракурсной трехмерной детекции объектов ImVoxelNet. Этот метод превосходит по точности все существующие методы трехмерной детекции объектов в уличных сценах, что подтверждено результатами экспериментов на наборах данных KITTI и nuScenes. В работе впервые была поставлена задача мультиракурсной трехмерной детекции объектов внутри помещений и предложено ее решение.

Список литературы

1. **Рухович Д. Д.** Итеративный метод обнаружения объектов // Программная инженерия. — 2021. — Т. 12, № 1. — С. 31–39.
2. **Ren S., He K., Girshick R.** et al. Faster r-cnn: Towards real-time object detection with region proposal networks // Advances in neural information processing systems. — 2015. — P. 91–99.
3. **Lin T., Goyal P., Girshick R.** et al. Focal loss for dense object detection // Proceedings of the IEEE international conference on computer vision. — 2017. — P. 2980–2988.
4. **Tian Z., Shen C., Chen H.** et al. Fcos: Fully convolutional one-stage object detection // Proceedings of the IEEE international conference on computer vision. — 2019. — P. 9627–9636.
5. **Liu Z., Wu Z., Toth R.** Smoke: single-stage monocular 3d object detection via keypoint estimation // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. — 2020. — P. 996–997.
6. **Lang A. H., Vora S., Caesar H.** et al. Pointpillars: Fast encoders for object detection from point clouds // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. — 2019. — P. 12697–12705.
7. **Yan Y., Mao Y., Li B.** Second: Sparsely embedded convolutional detection // Sensors. — 2018. — Vol. 18, No. 10. — P. 3337.
8. **Zhou Y., Tuzel O.** Voxelnet: End-to-end learning for point cloud based 3d object detection // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. — 2018. — P. 4490–4499.
9. **Sindagi V. A., Zhou Y., Tuzel O.** Mvx-net: Multimodal voxelnet for 3d object detection // 2019 International Conference on Robotics and Automation (ICRA). — IEEE, 2019. — P. 7276–7282.
10. **Qi C. R., Liu W., Wu C.** et al. Frustum pointnets for 3d object detection from rgb-d data // Proceedings of the IEEE conference on computer vision and pattern recognition. — 2018. — P. 918–927.
11. **Xu D., Anguelov D., Jain A.** Pointfusion: Deep sensor fusion for 3d bounding box estimation // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. — 2018. — P. 244–253.
12. **Song S., Xiao J.** Deep sliding shapes for amodal 3d object detection in rgb-d images // Proceedings of the IEEE conference on computer vision and pattern recognition. — 2016. — P. 808–816.
13. **Chen X., Kundu K., Zhu Y.** et al. 3d object proposals using stereo imagery for accurate object class detection // IEEE transactions on pattern analysis and machine intelligence. — 2017. — Vol. 40, No. 5. — P. 1259–1272.
14. **Qin Z., Wang J., Lu Y.** Triangulation learning network: from monocular to stereo 3d object detection // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. — 2019. — P. 7615–7623.
15. **Li P., Chen X., Shen S.** Stereo r-cnn based 3d object detection for autonomous driving // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. — 2019. — P. 7644–7652.
16. **Chen X., Kundu K., Zhang Z.** et al. Monocular 3d object detection for autonomous driving // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. — 2016. — P. 2147–2156.
17. **Mousavian A., Anguelov D., Flynn J.** et al. 3d bounding box estimation using deep learning and geometry // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. — 2017. — P. 7074–7082.
18. **Qin Z., Wang J., Lu Y.** Monogrnet: A geometric reasoning network for monocular 3d object localization // Proceedings of the AAAI Conference on Artificial Intelligence. — 2019. — Vol. 33, No. 01. — P. 8851–8858.
19. **Kundu A., Li Y., Rehag J.** M3d-rcnn: Instance-level 3d object reconstruction via render-and-compare // Proceedings of the IEEE conference on computer vision and pattern recognition. — 2018. — P. 3559–3568.
20. **Ma X., Wang Z., Li H.** et al. Accurate monocular 3d object detection via color-embedded 3d reconstruction for autonomous driving // Proceedings of the IEEE/CVF International Conference on Computer Vision. — 2019. — P. 6851–6860.
21. **Ku J., Pon A. D., Waslander S. L.** Monocular 3d object detection leveraging accurate proposals and shape reconstruction // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. — 2019. — P. 11867–11876.
22. **Qi C., Su H., Mo K.** et al. Pointnet: Deep learning on point sets for 3d classification and segmentation // Proceedings of the IEEE conference on computer vision and pattern recognition. — 2017. — P. 652–660.
23. **Manhardt F., Kehl W., Gaidon A.** Roi-10d: Monocular lifting of 2d detection to 6d pose and metric shape // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. — 2019. — P. 2069–2078.
24. **Brazil G., Liu X.** M3d-rpn: Monocular 3d region proposal network for object detection // Proceedings of the IEEE/CVF International Conference on Computer Vision. — 2019. — P. 9287–9296.
25. **Roddick T., Kendall A., Cipolla R.** Orthographic feature transform for monocular 3D object detection // 30th British Machine Vision Conference 2019, BMVC 2019. — 2019. URL: <https://bmv2019.org/wp-content/uploads/papers/0328-paper.pdf>
26. **Simonyan K., Zisserman A.** Very deep convolutional networks for large-scale image recognition // arXiv preprint arXiv:1409.1556. — 2014.
27. **He K., Zhang X., Ren S.** et al. Deep residual learning for image recognition // Proceedings of the IEEE conference on computer vision and pattern recognition. — 2016. — P. 770–778.
28. **Murez Z., van As T., Bartolozzi J.** et al. Atlas: End-to-end 3d scene reconstruction from posed images // Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VII 16. — Springer International Publishing, 2020. — P. 414–431.
29. **Geiger A., Lenz P., Urtasun R.** Are we ready for autonomous driving? the kitti vision benchmark suite // 2012 IEEE Conference on Computer Vision and Pattern Recognition. — IEEE, 2012. — P. 3354–3361.
30. **Caesar H., Bankiti V., Lang A.** et al. nuscenes: A multimodal dataset for autonomous driving // Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. — 2020. — P. 11621–11631.
31. **Chen K., Wang J., Pang J.** et al. Mmdetection: Open mmlab detection toolbox and benchmark // arXiv preprint arXiv:1906.07155. — 2019.
32. **Paszke A., Gross S., Massa F.** et al. Pytorch: An imperative style, high-performance deep learning library // Advances in neural information processing systems. — 2019. — P. 8026–8037.
33. **Li B., Ouyang W., Sheng L.** et al. Gs3d: An efficient 3d object detection framework for autonomous driving // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. — 2019. — P. 1019–1028.
34. **Jørgensen E., Zach C., Kahl F.** Monocular 3d object detection and box fitting trained end-to-end using intersection-over-union loss // arXiv preprint arXiv:1906.08070. — 2019.
35. **Simonelli A., Bulò S., Porzi L.** et al. Disentangling Monocular 3D Object Detection: From Single to Multi-Class Recognition // IEEE Transactions on Pattern Analysis and Machine Intelligence. — 2020. URL: <https://ieeexplore.ieee.org/abstract/document/9200697>
36. **Chen Y., Tai L., Sun K.** et al. Monopair: Monocular 3d object detection using pairwise spatial relationships // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. — 2020. — P. 12093–12102.
37. **Li P., Zhao H., Liu P.** et al. Rtm3d: Real-time monocular 3d detection from object keypoints for autonomous driving // Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III. — Springer International Publishing, 2020. — P. 644–660.
38. **Duan K., Bai S., Xie L.** et al. Centernet: Keypoint triplets for object detection // Proceedings of the IEEE/CVF International Conference on Computer Vision. — 2019. — P. 6569–6578.
39. **Nabati R., Qi H.** Centerfusion: Center-based radar and camera fusion for 3d object detection // Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. — 2021. — P. 1527–1536.

2D-to-3D Projection for Monocular and Multi-View 3D Object Detection in Outdoor Scenes

D. D. Rukhovich, daniel-rukovich@yandex.ru, Faculty of Mechanics and Mathematics, Moscow State University, Moscow, 119234, Russian Federation

Corresponding author:

Rukhovich Danila D., Postgraduate Student, Faculty of Mechanics and Mathematics, Moscow State University, Moscow, 119234, Russian Federation
E-mail: daniel-rukovich@yandex.ru

Received on June 16, 2021

Accepted on July 07, 2021

In this article, we introduce the task of multi-view RGB-based 3D object detection as an end-to-end optimization problem. In a multi-view formulation of the 3D object detection problem, several images of a static scene are used to detect objects in the scene.

To address the 3D object detection problem in a multi-view formulation, we propose a novel 3D object detection method named ImVoxelNet. ImVoxelNet is based on a fully convolutional neural network. Unlike existing 3D object detection methods, ImVoxelNet works directly with 3D representations and does not mediate 3D object detection through 2D object detection. The proposed method accepts multi-view inputs. The number of monocular images in each multi-view input can vary during training and inference; actually, this number might be unique for each multi-view input. Moreover, we propose to treat a single RGB image as a special case of a multi-view input. Accordingly, the proposed method can also accept monocular inputs with no modifications.

Through extensive evaluation, we demonstrate that the proposed method successfully handles a variety of outdoor scenes. Specifically, it achieves state-of-the-art results in car detection on KITTI (monocular) and nuScenes (multi-view) benchmarks among all methods that accept RGB images.

The proposed method operates in real-time, which makes it possible to integrate it into the navigation systems of autonomous devices. The results of this study can be used to address tasks of navigation, path planning, and semantic scene mapping.

Keywords: machine learning, deep learning, 3D object detection

For citation:

Rukhovich D. D. 2D-to-3D Projection for Monocular and Multi-View 3D Object Detection in Outdoor Scenes, *Programmnyaya Ingeneriya*, 2017, vol. 12, no. 7, pp. 373–384.

DOI: 10.17587/prin.12.373-384

References

1. **Rukhovich D. D.** Iterative Scheme for Object Detection in Crowded Environments, *Programmnyaya Ingeneriya*, 2021, vol. 12, no. 1, pp. 31–39 (in Russian).
2. **Ren S., He K., Girshick R., Sun J.** Faster r-cnn: Towards real-time object detection with region proposal networks, *Advances in neural information processing systems*, 2015, pp. 91–99.
3. **Lin T., Goyal P., Girshick R., He K., Dollar P.** Focal loss for dense object detection *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.
4. **Tian Z., Shen C., Chen H., He T.** Fcos: Fully convolutional one-stage object detection, *Proceedings of the IEEE international conference on computer vision*, 2019, pp. 9627–9636.
5. **Liu Z., Wu Z., Toth R.** Smoke: single-stage monocular 3d object detection via keypoint estimation, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 996–997.
6. **Lang A. H., Vora S., Caesar H., Zhou L., Yang J., Beijbom O.** Pointpillars: Fast encoders for object detection from point clouds, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12697–12705.
7. **Yan Y., Mao Y., Li B.** Second: Sparsely embedded convolutional detection, *Sensors*, 2018, vol. 18, no. 10, p. 3337.
8. **Zhou Y., Tuzel O.** Voxelnet: End-to-end learning for point cloud based 3d object detection, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4490–4499.
9. **Sindagi V. A., Zhou Y., Tuzel O.** Mvx-net: Multimodal voxelnet for 3d object detection, *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 7276–7282.
10. **Qi C. R., Liu W., Wu C., Su H., Guibas L. J.** Frustum pointnets for 3d object detection from rgb-d data, *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 918–927.
11. **Xu D., Anguelov D., Jain A.** Pointfusion: Deep sensor fusion for 3d bounding box estimation, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 244–253.
12. **Song S., Xiao J.** Deep sliding shapes for amodal 3d object detection in rgb-d images, *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 808–816.
13. **Chen X., Kundu K., Zhu Y., Ma H., Fidler S., Urtasun R.** 3d object proposals using stereo imagery for accurate object class detection, *IEEE transactions on pattern analysis and machine intelligence*, 2017, vol. 40, no. 5, pp. 1259–1272.
14. **Qin Z., Wang J., Lu Y.** Triangulation learning network: from monocular to stereo 3d object detection, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7615–7623.

15. **Li P., Chen X., Shen S.** Stereo r-cnn based 3d object detection for autonomous driving, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7644–7652.
16. **Chen X., Kundu K., Zhang Z., Ma H., Fidler S., Urtasun R.** Monocular 3d object detection for autonomous driving, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2147–2156.
17. **Mousavian A., Anguelov D., Flynn J., Kosecka J.** 3d bounding box estimation using deep learning and geometry, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7074–7082.
18. **Qin Z., Wang J., Lu Y.** Monogrnet: A geometric reasoning network for monocular 3d object localization, *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019, vol. 33, no. 01, pp. 8851–8858.
19. **Kundu A., Li Y., Rehag J. M.** 3d-rcnn: Instance-level 3d object reconstruction via render-and-compare, *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3559–3568.
20. **Ma X., Wang Z., Li H., Zhang P., Ouyang W., Fan X.** Accurate monocular 3d object detection via color-embedded 3d reconstruction for autonomous driving, *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 6851–6860.
21. **Ku J., Pon A. D., Waslander S. L.** Monocular 3d object detection leveraging accurate proposals and shape reconstruction, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11867–11876.
22. **Qi C. R., Su H., Mo K., Guibas L. J.** Pointnet: Deep learning on point sets for 3d classification and segmentation, *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 652–660.
23. **Manhardt F., Kehl W., Gaidon A.** Roi-10d: Monocular lifting of 2d detection to 6d pose and metric shape, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2069–2078.
24. **Brazil G., Liu X.** M3d-rpn: Monocular 3d region proposal network for object detection, *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9287–9296.
25. **Roddick T., Kendall A., Cipolla R.** Orthographic feature transform for monocular 3D object detection, *30th British Machine Vision Conference 2019, BMVC 2019*, 2019, available at: <https://bmvc2019.org/wp-content/uploads/papers/0328-paper.pdf>
26. **Simonyan K., Zisserman A.** Very deep convolutional networks for large-scale image recognition, *arXiv preprint arXiv:1409.1556*, 2014.
27. **He K., Zhang X., Ren S., Sun J.** Deep residual learning for image recognition, *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
28. **Murez Z., van As T., Bartolozzi J., Sinha A., Badrinarayanan V., Rabinovich A.** Atlas: End-to-end 3d scene reconstruction from posed images, *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VII 16*, 2020, pp. 414–431.
29. **Geiger A., Lenz P., Urtasun R.** Are we ready for autonomous driving? the kitti vision benchmark suite, *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 3354–3361.
30. **Caesar H., Bankiti V., Lang A., Vora S., Liong V. E., Xu Q., Krishnan A., Pan Y., Baldan G., Beijbom O.** nuscenes: A multimodal dataset for autonomous driving, *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11621–11631.
31. **Chen K., Wang J., Pang J., Cao Y., Xiong Y., Li X., Sun S., Feng W., Liu Z., Xu J., Zhang Z., Cheng D., Zhu C., Cheng T., Zhao Q., Li B., Lu X., Zhu R., Wu Y., Dai J., Wang J., Shi J., Ouyang W., Loy C. C., Lin D.** Mmdetection: Open mmlab detection toolbox and benchmark, *arXiv preprint arXiv:1906.07155*, 2019.
32. **Paszke A., Gross S., Massa F., Lerer A., Bradbury J., Chanan G., Killeen T., Lin Z., Gimelshein N., Antiga L., Desmaison A., Kopf A., Yang E., DeVito Z., Raison M., Tejani A., Chilamkurthy S., Steiner B., Fang L., Bai J., Chintala S.** Pytorch: An imperative style, high-performance deep learning library, *Advances in neural information processing systems*, 2019, pp. 8026–8037.
33. **Li B., Ouyang W., Sheng L., Zeng X., Wang X.** Gs3d: An efficient 3d object detection framework for autonomous driving, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1019–1028.
34. **Jørgensen E., Zach C., Kahl F.** Monocular 3d object detection and box fitting trained end-to-end using intersection-over-union loss, *arXiv preprint arXiv:1906.08070*, 2019.
35. **Simonelli A., Bulò S., Porzi L., Antequera M. L., Kotschieder P.** Disentangling Monocular 3D Object Detection: From Single to Multi-Class Recognition, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020, available at: <https://ieeexplore.ieee.org/abstract/document/9200697>
36. **Chen Y., Tai L., Sun K., Li M.** Monopair: Monocular 3d object detection using pairwise spatial relationships, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 12093–12102.
37. **Li P., Zhao H., Liu P., Cao F.** Rtm3d: Real-time monocular 3d detection from object keypoints for autonomous driving, *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III*, 2020, pp. 644–660.
38. **Duan K., Bai S., Xie L., Qi H., Huang Q., Tian Q.** CenterNet: Keypoint triplets for object detection, *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 6569–6578.
39. **Nabati R., Qi H.** Centerfusion: Center-based radar and camera fusion for 3d object detection, *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021, pp. 1527–1536.

**Начинается подписка на журнал
"Программная инженерия" на первое полугодие 2022 г.**

Оформить подписку можно в любом отделении Почты России, через подписные агентства или непосредственно в редакции журнала.

Подписной индекс по Объединенному каталогу

"Пресса России" — 22765

Сообщаем, что с 2020 г. возможна подписка на электронную версию нашего журнала через:

ООО "ИВИС": тел. (495) 777-65-57, 777-65-58; e-mail: sales@ivis.ru,

ООО "УП Урал-Пресс". Для оформления подписки (индекс 013312)

следует обратиться в филиал по месту жительства — <http://ural-press.ru>

Адрес редакции: 107076, Москва, ул. Матросская Тишина, д. 23, с. 2, оф. 45

Издательство "Новые технологии",

редакция журнала "Программная инженерия"

Тел.: (499) 270-16-52. E-mail: prin@novtex.ru

М. А. Торгашев, канд. физ.-мат. наук, зав. сектором, mtorg@mail.ru,
Е. В. Страшнов, науч. сотр., strashnov_evlg@mail.ru, Федеральное государственное учреждение "Федеральный научный центр Научно-исследовательский институт системных исследований Российской академии наук" (ФГУ ФНЦ НИИСИ РАН), Москва

Определение коллизий аппроксимирующих цилиндров в системах виртуального окружения

Предложены методы и алгоритмы определения коллизий аппроксимирующих цилиндров, которые окружают геометрию виртуальных объектов. Предлагаемые решения основаны на теореме о разделяющей оси, аппроксимации цилиндров призмами с применением отображения Гаусса и анализе возможных случаев контактного взаимодействия двух цилиндров. В свою очередь, поиск контактных точек базируется на быстрых геометрических тестах, в которых выполняется отсечение многоугольника и отрезка призмой, а также вычисление ближайших точек на отрезках и окружностях. Апробация предлагаемых в статье решений проводилась в программном комплексе виртуального окружения, разработанном в ФГУ ФНЦ НИИСИ РАН, и показала их адекватность и эффективность.

Ключевые слова: определение коллизий, аппроксимирующий цилиндр, призма, теорема о разделяющей оси, отображение Гаусса, отсечение, система виртуального окружения

Введение

В настоящее время широкое распространение получили системы виртуального окружения и виртуальной реальности, в которых осуществляется компьютерное моделирование виртуальных объектов. Примером служат имитационно-тренажерные комплексы, которые создаются для обучения человека навыкам управления реальными устройствами (автотренажеры, авиасимуляторы, тренажеры мобильных роботов и т. д.) на виртуальных моделях с использованием моделируемой и визуализируемой на компьютере трехмерной виртуальной среды. Для повышения качества такого обучения требуется обеспечить адекватное поведение виртуальных объектов, согласно реальным законам физики. В процессе движения объекты могут сталкиваться и пересекаться друг с другом, что приводит к необходимости решения задачи определения коллизий этих объектов [1, 2]. Разработка быстрых и эффективных алгоритмов определения коллизий объектов, которые будут обеспечивать физическую правдоподобность поведения этих объектов при их взаимодействии, является важной и актуальной задачей.

Один из подходов для определения коллизий виртуальных объектов заключается в использовании аппроксимирующих контейнеров [1, с. 75] стандартной формы (сфер, цилиндров, параллелепипедов, капсул и т. д.), которые окружают геометрию виртуальных объектов. Тогда задача сводится к разработке методов и алгоритмов определения пересечений для каждой пары таких контейнеров. В настоящей работе рассматривается задача определения коллизий аппроксимирующих цилиндров, которая актуальна

для моделирования в виртуальном окружении движения колес транспортных средств, качении бочек и перемещений других объектов, форма которых близка к цилиндрической. Важное требование, которое предъявляют системы виртуального окружения, состоит в том, что разрабатываемые алгоритмы определения коллизий аппроксимирующих цилиндров должны работать в масштабе реального времени. Это означает, что все расчеты должны занимать не более 40 мс, что необходимо для формирования одного кадра визуализации.

Задаче определения коллизий виртуальных объектов посвящено много публикаций. Первый подход базируется на связке итерационных алгоритмов, которые носят название GJK (англ. *Gilbert Johnson Keerthi algorithm*) и EPA (англ. *Expanding Polytope Algorithm*) для объектов, представленных в виде выпуклой оболочки из множества точек [3–7]. Эти алгоритмы нашли широкое применение для определения коллизий многогранников. Однако они сложны в плане их математического описания и программной реализации и обладают слабой численной устойчивостью. Другой подход для определения коллизий виртуальных объектов основан на применении теоремы о разделяющей оси SAT [8–10] (англ. *Separating Axis Theorem*). Эффективность такого подхода непосредственно зависит от сложности геометрии и числа осей для тестирования на пересечение. В свою очередь, задача определения коллизий цилиндров была рассмотрена в работах [11–14]. Предлагаемые решения включают аналитические методы [11, 12] обработки столкновений двух цилиндров с применением алгебры дуальных чисел. Их идея заключается в том, что определение пересечений цилиндров базируется

ся на тестах между линиями и прямоугольниками в плоскости, включая поиск общей нормали между осями цилиндров и точек, где нормаль пересекает оси. Представленные в этих публикациях алгоритмы нашли применение в приложениях планирования движения роботов, где не требуется обеспечение реального времени. В работе [13] приведен обзор различных случаев контакта двух цилиндров с иллюстрацией вычисления контактных точек. Однако на практике требуется идентифицировать каждый из этих случаев по отдельности. Алгоритмы определения коллизий двух цилиндров с помощью SAT предложены в работе [14] для ограниченного числа направлений, некоторые из которых вычисляются путем оптимизации специальной функции на сфере, что требует существенных вычислений. Несмотря на наличие решений для определения коллизий цилиндров, остается актуальным создание быстрых и эффективных алгоритмов, работающих в масштабе реального времени.

В настоящей статье предложены оригинальные методы и алгоритмы определения коллизий двух цилиндров, в которых разделяющие оси вычисляются в зависимости от возможных случаев контакта цилиндров. Идея заключается в аппроксимации цилиндров призмами с их проецированием на единичную сферу, что позволяет сократить число разделяющих осей. В свою очередь, для вычисления контактных точек предлагается подход, в котором выполняется отсечение ближайших элементов двух цилиндров и поиск точек между линиями и окружностями. Предложенные в настоящей статье алгоритмы были реализованы в разработанном авторами в ФГУ ФНЦ НИИСИ РАН программном комплексе виртуального окружения VirSim [15] и апробированы на примерах взаимодействия объектов, имеющих форму цилиндра.

1. Постановка задачи

Рассмотрим два аппроксимирующих цилиндра (рис. 1) с радиусами r_i и высотами h_i , центры которых расположены в точках O_i , а оси задаются с помощью единичных векторов v_i , $i = 1, 2$. Основаниями цилиндров являются круги с центрами A_i и B_i . Все значения координат точек и векторов задаются в мировой системе координат.

В контексте настоящей статьи рассматриваем апостериорные алгоритмы определения коллизий.

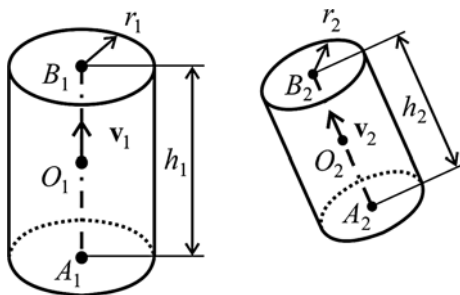


Рис. 1. Аппроксимирующие цилиндры

Такие алгоритмы обрабатывают коллизию по факту пересечения объектов. Задача определения коллизий двух цилиндров заключается в проверке их пересечения и вычислении необходимой контактной информации, что требуется для правильного расталкивания объектов на стадии разрешения коллизий. Контактная информация задается в виде множества параметров $C_P = \{P, \mathbf{n}, d\}$, где P — точка контакта; \mathbf{n} — единичный вектор нормали в точке P ; d — глубина проникновения. В общем случае контактные точки образуют целое множество (пятно контакта). Из этих точек требуется выбрать конечное число, которые наилучшим образом аппроксимируют пятно контакта. При рассмотрении пересечения двух цилиндров требуется найти точки P_i на поверхности первого цилиндра, их глубины проникновения d_i , $i = 1, M$, где M — число точек контакта, а также общую нормаль \mathbf{n} , которая направлена из первого цилиндра во второй. Далее более подробно опишем предлагаемые методы и алгоритмы для решения этой задачи.

2. Определение коллизий двух цилиндров

Для определения коллизий двух цилиндров воспользуемся теоремой о разделяющей оси [8–10]. Идея этой теоремы заключается в том, что два выпуклых объекта не пересекаются, если найдется такая плоскость, которая их разделяет. Другими словами, это означает, что не пересекаются проекции объектов на ось, которая является перпендикуляром к этой плоскости. Если такая ось не найдена, то объекты пересекаются. Важное следствие этой теоремы состоит в том, что глубина проникновения объектов равна минимальной длине их проекций на некоторую ось. Этот факт позволяет не только определить пересечение объектов, но также вычислить нормаль \mathbf{n} и глубину проникновения d .

Покажем, как применяется теорема о разделяющей оси для определения коллизий двух аппроксимирующих цилиндров. Для этого спроецируем цилиндры на разделяющую ось, которая задается вектором \mathbf{d}_a (рис. 2). В силу симметрии значение величины

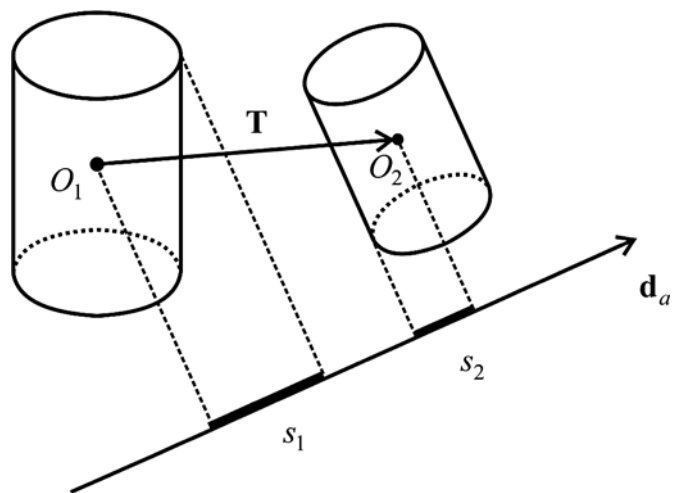


Рис. 2. Проекция цилиндров на разделяющую ось

$$s_i = \frac{h_i}{2} |\mathbf{v}_i \cdot \mathbf{d}_a| + r_i \sqrt{1 - (\mathbf{v}_i \cdot \mathbf{d}_a)^2}, \quad i = 1, 2 \quad (1)$$

равно половине проекции цилиндра на разделяющую ось \mathbf{d}_a [16].

Рассмотрим вектор $\mathbf{T} = O_1O_2$, соединяющий центры двух цилиндров и направленный из первого цилиндра во второй. Если найдется такая ось \mathbf{d}_a , для которой будет выполнено неравенство

$$|\mathbf{T} \cdot \mathbf{d}_a| > s_1 + s_2, \quad (2)$$

то два цилиндра не пересекаются.

В противном случае, среди всех возможных разделяющих осей выполняется поиск такой оси \mathbf{d}_a , для которой величина

$$d = \min(s_1 + s_2 - |\mathbf{T} \cdot \mathbf{d}_a|)$$

является глубиной проникновения, а \mathbf{d}_a — нормалью, т. е. $\mathbf{n} = \mathbf{d}_a$.

Непосредственное применение этой теоремы к аппроксимирующим цилиндрам затруднительно, так как в этом случае получим бесконечное число разделяющих осей. Поэтому в данной работе предлагается решение, в котором цилиндр аппроксимируется с помощью правильной n -угольной призмы, нормали к граням которой вычисляются как

$$\mathbf{e}_{k,i} = \mathbf{u}_i \cos \varphi_k + \mathbf{w}_i \sin \varphi_k, \quad \varphi_k = \frac{2\pi k}{n} - \frac{\pi}{n},$$

$$k = \overline{1, n}, \quad i = 1, 2,$$

где \mathbf{u}_i и \mathbf{w}_i — единичные орты цилиндров такие, что $\mathbf{v}_i \perp \mathbf{u}_i$ и $\mathbf{w}_i \perp \mathbf{u}_i$.

При таком подходе в качестве разделяющих осей могут быть выбраны перпендикуляры к граням двух призм и попарные векторные произведения векторов их ребер. В связи с этим обстоятельством рассматриваются следующие случаи контактного взаимодействия двух цилиндров:

- а) основание первого цилиндра со вторым цилиндром;
- б) основание второго цилиндра с первым цилиндром;
- в) боковые поверхности двух цилиндров;

г) боковая поверхность первого цилиндра со вторым цилиндром;

д) боковая поверхность второго цилиндра с первым цилиндром;

е) боковая поверхность первого цилиндра с окружностью основания второго цилиндра;

ж) боковая поверхность второго цилиндра с окружностью основания первого цилиндра;

з) окружности оснований двух цилиндров.

При контакте оснований цилиндра (случаи а и б) получим, что разделяющими являются оси цилиндра, т. е. $\mathbf{d}_a = \mathbf{v}_i$. Для контакта боковых поверхностей цилиндров (случай в) разделяющая ось вычисляется через векторное произведение осей двух цилиндров: $\mathbf{d}_a = \mathbf{v}_1 \times \mathbf{v}_2$. При контакте боковых поверхностей цилиндра (случаи г и д) разделяющими осями являются перпендикуляры к граням призмы, т. е. $\mathbf{d}_a = \mathbf{e}_{k,i}$. Для случая е разделяющая ось образована векторным произведением оси первого цилиндра с касательной к окружности основания второго цилиндра и вычисляется как $\mathbf{d}_a = \mathbf{v}_1 \times (\mathbf{v}_2 \times \mathbf{e}_{k,2})$, $k = \overline{1, n}$. Аналогично, для случая ж разделяющей осью является вектор $\mathbf{d}_a = (\mathbf{v}_1 \times \mathbf{e}_{j,1}) \times \mathbf{v}_2$, $j = \overline{1, n}$. Наконец, случай з соответствует векторному произведению касательных двух окружностей: $\mathbf{d}_a = (\mathbf{v}_1 \times \mathbf{e}_{j,1}) \times (\mathbf{v}_2 \times \mathbf{e}_{k,2})$, $j = \overline{1, n}$, $k = \overline{1, n}$.

Однако в последнем случае число разделяющих осей равно n^2 , что является достаточно большим числом при увеличении n . Для решения вопроса о сокращении перебора возможных разделяющих осей предлагается следующий подход. Рассмотрим отображение Гаусса (сферическое отображение), которое переводит призму в единичную сферу [17, с. 75]. Тогда грани призмы переходят в вершины окружности, а ребра — в дуги (на рис. 3, а показано на примере аппроксимации цилиндра восьмиугольной призмой). Согласно теории разности Минковского [10] разделяющими осями могут быть только вершины на единичной сфере, которые получены при проецировании двух призм. Это означает, что надо рассматривать только те ребра призм, дуги которых пересекаются (рис. 3, б). Пусть \mathbf{a} и \mathbf{b} — векторы вершин первой дуги, а \mathbf{c} и \mathbf{d} — векторы вершин второй. Тогда тестом на пересечение дуг является проверка условий

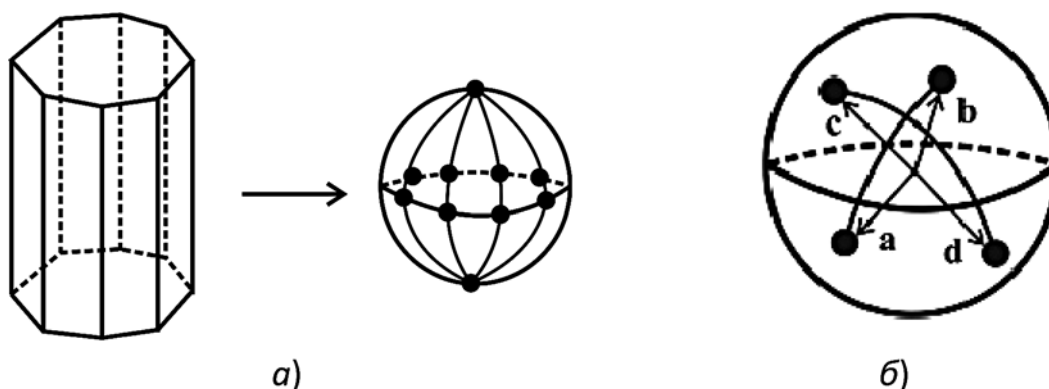


Рис. 3. Отображение Гаусса из призмы в сферу (а) и разность Минковского (б)

$$\begin{aligned}
[\mathbf{c} \cdot (\mathbf{b} \times \mathbf{a})] \cdot [\mathbf{d} \cdot (\mathbf{b} \times \mathbf{a})] &< 0; \\
[\mathbf{a} \cdot (\mathbf{d} \times \mathbf{c})] \cdot [\mathbf{b} \cdot (\mathbf{d} \times \mathbf{c})] &< 0; \\
[\mathbf{c} \cdot (\mathbf{b} \times \mathbf{a})] \cdot [\mathbf{b} \cdot (\mathbf{d} \times \mathbf{c})] &> 0.
\end{aligned} \quad (3)$$

Первое условие (3) означает, что вершины второй дуги располагаются по разные стороны от плоскости первой дуги. Второе условие формулируется аналогичным образом для вершин первой дуги. Третье условие является проверкой того факта, что дуги располагаются в одной и той же полусфере. Для двух призм в качестве вершин дуг выбираются $\mathbf{a} = \pm \mathbf{v}_1$, $\mathbf{b} = \mathbf{e}_j$, $\mathbf{c} = \pm \mathbf{v}_2$, $\mathbf{d} = \mathbf{e}_k$.

Проверка на пересечение двух цилиндров реализуется с помощью следующего алгоритма, который определяет коллизию двух цилиндров.

Алгоритм. Определение коллизий двух цилиндров.

1. Инициализация: $\mathbf{n} = 0$, $d = d_{\max}$, где $d_{\max} > 0$ — большое значение.

2. Формируем потенциальные разделяющие оси \mathbf{d}_a^i , $i = 1, N$ с учетом разности Минковского и условий (3).

3. Цикл $i = 1, \dots, N$ по разделяющим осям:

3.1) вычисляем вектор \mathbf{T} , а также проекции s_1 и s_2 по формуле (1);

3.2) если условие (2) выполнено, то цилиндры не пересекаются; досрочный выход из цикла;

3.3) вычисляем $\Delta d = s_1 + s_2 - |\mathbf{T} \cdot \mathbf{d}_a^i|$;

3.4) если $\Delta d < d$, то $d = \Delta d$, $\mathbf{n} = \mathbf{d}_a^i$.

Конец алгоритма.

Приведенный алгоритм позволяет определить, пересекаются ли два цилиндра, а также вычислить нормаль \mathbf{n} и глубину проникновения d . В случае, если цилиндры пересекаются, то требуется найти точки контакта.

3. Вычисление контактных точек

Точки контакта вычисляются в зависимости от взаимного расположения двух цилиндров. В данной работе предлагаются решения, основанные на отсечении многоугольника и отрезка призмой, на поиске общего перпендикуляра к двум отрезкам и на вычислении точки на окружности основания цилиндра. Рассмотрим предлагаемые решения более подробно.

Отсечение многоугольника призмой. При контакте оснований двух цилиндров разделяющей является ось первого цилиндра. В этом случае поиск контактных точек осуществляется на окружности основания второго цилиндра путем отсечения $n + 2$ плоскостями призмы первого цилиндра (рис. 4). Плоскость отсечения задается в следующем виде:

$$s(P) = P \cdot \mathbf{n}_p - k_p, \quad (4)$$

где P — точка, принадлежащая плоскости; \mathbf{n}_p — нормаль к плоскости; k_p — постоянная плоскости.

Пусть многоугольник с вершинами $Q_1^{(i)} \dots Q_n^{(i)}$ относится к окружности второго цилиндра, которая наиболее удалена в противоположном направлении относительно нормали \mathbf{n} . Покажем, как выполняется

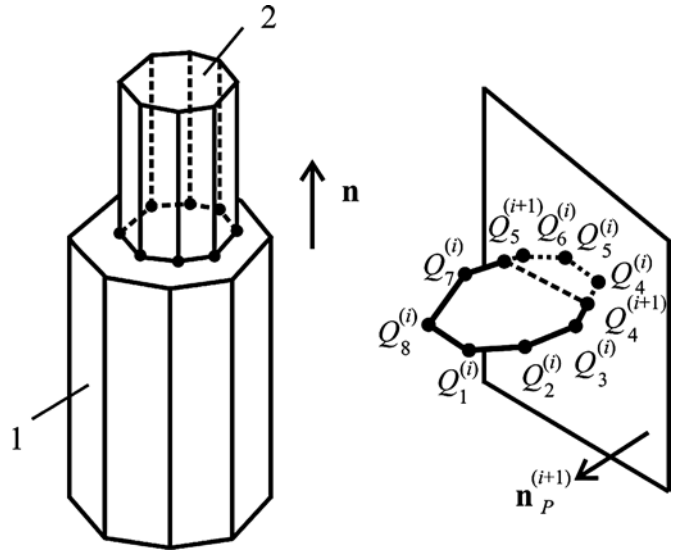


Рис. 4. Отсечение многоугольника призмой

отсечение этого многоугольника $n + 2$ плоскостями призмы первого цилиндра, где плоскость нижнего основания имеет нормаль $\mathbf{n}_p^{(1)} = \mathbf{v}_1$ с постоянной $k_p^{(1)} = h_1/2$, плоскость верхнего основания — $\mathbf{n}_p^{(2)} = -\mathbf{v}_1$ и $k_p^{(2)} = h_1/2$, а плоскости призмы — $\mathbf{n}_p^{(j+2)} = -\mathbf{e}_{j,1}$ и $k_p^{(j+2)} = r_1$, $j = \overline{1, n}$. Для этого рассмотрим очередное отсечение, как показано на рис. 4. На этом шаге имеем многоугольник с вершинами $Q_1^{(i)} \dots Q_K^{(i)}$, который был получен при выполнении предыдущих отсечений, где K — число вершин многоугольника. Для каждого ребра $Q_k^{(i)} Q_{k+1}^{(i)}$ этого многоугольника проверяем расположение вершин относительно плоскости. Если из уравнения (4) выполнено $s_{i+1}(Q_k^{(i)}) \geq 0$ и $s_{i+1}(Q_{k+1}^{(i)}) \geq 0$, то ребро находится перед плоскостью, а его вершины $Q_k^{(i)}$ и $Q_{k+1}^{(i)}$ добавляются в новый многоугольник. При $s_{i+1}(Q_k^{(i)}) < 0$ и $s_{i+1}(Q_{k+1}^{(i)}) < 0$ ребро находится позади плоскости, а его вершины не участвуют в формировании нового многоугольника. В том случае, если $s_{i+1}(Q_k^{(i)})$ и $s_{i+1}(Q_{k+1}^{(i)})$ имеют разные знаки, то требуется вычислить точку $Q_l^{(i+1)}$ пересечения ребра $Q_k^{(i)} Q_{k+1}^{(i)}$ с плоскостью. Для этого подставим $Q_l^{(i+1)} = Q_k^{(i)} + \alpha Q_k^{(i)} Q_{k+1}^{(i)}$ в $s_{i+1}(Q_l^{(i+1)}) = 0$:

$$s_{i+1}(Q_k^{(i)}) + \alpha (s_{i+1}(Q_{k+1}^{(i)}) - s_{i+1}(Q_k^{(i)})) = 0.$$

Решая это уравнение относительно параметра α , получим, что точка пересечения ребра с плоскостью вычисляется как

$$Q_l^{(i+1)} = Q_k^{(i)} + \frac{s_{i+1}(Q_k^{(i)})}{s_{i+1}(Q_k^{(i)}) - s_{i+1}(Q_{k+1}^{(i)})} Q_k^{(i)} Q_{k+1}^{(i)}.$$

На рис. 4 показан пример, в котором после отсечения удаляются вершины $Q_4^{(i)}$, $Q_5^{(i)}$, $Q_6^{(i)}$ и добав-

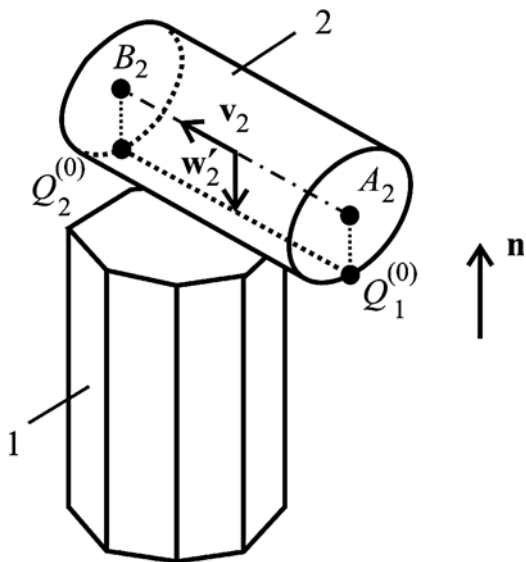


Рис. 5. Отсечение отрезка призмой

ляются новые вершины $Q_4^{(i+1)}$, $Q_5^{(i+1)}$. В результате всех отсечений получим многоугольник $Q_1 \dots Q_M$, вершины которого находятся на поверхности второго цилиндра. В качестве контактных точек рассматриваются проекции вершин многоугольника на поверхность первого цилиндра:

$$P_i = Q_i + d_i \mathbf{n}_i, \quad d_i = h_1/2 - |O_1 Q_i \cdot \mathbf{n}|, \quad i = \overline{1, M}, \quad (5)$$

где d_i — глубины проникновения точек.

Отсечение отрезка призмой. Рассмотрим контакт основания первого цилиндра с боковой поверхностью второго цилиндра. В этом случае выполняется отсечение отрезка $Q_1^{(0)} Q_2^{(0)}$ второго цилиндра $n + 2$ плоскостями первого цилиндра (рис. 5). Этот отрезок образуется путем проецирования оси второго цилиндра в противоположном направлении относительно нормали:

$$Q_1^{(0)} = A_2 + \mathbf{w}'_2, \quad Q_2^{(0)} = B_2 + \mathbf{w}'_2,$$

$$\text{где } \mathbf{w}'_2 = r_2 \frac{(\mathbf{v}_2 \cdot \mathbf{n})\mathbf{v}_2 - \mathbf{n}}{\|(\mathbf{v}_2 \cdot \mathbf{n})\mathbf{v}_2 - \mathbf{n}\|}.$$

После выполнения всех отсечений получим отрезок $Q_1 Q_2$. В рассматриваемом случае точками контакта являются проекции вершин этого отрезка на окружность основания первого цилиндра, вычисляемые по формуле (5). Аналогичным образом выполняется отсечение для случая, когда нормаль является перпендикуляром к боковой поверхности цилиндра.

Общий перпендикуляр к двум отрезкам. Здесь рассматривается случай контакта боковых поверхностей двух цилиндров с нормалью $\mathbf{n} = \mathbf{v}_1 \times \mathbf{v}_2$. Точка контакта P находится на общем перпендикуляре PQ к двум отрезкам $D_1 D_2$ и $E_1 E_2$ боковых поверхностей первого и второго цилиндра (рис. 6). С использованием параметрического задания отрезка получим, что

$$P = D_1 + \alpha \mathbf{v}_1, \quad Q = E_1 + \beta \mathbf{v}_2, \quad (6)$$

где α и β — неизвестные параметры.

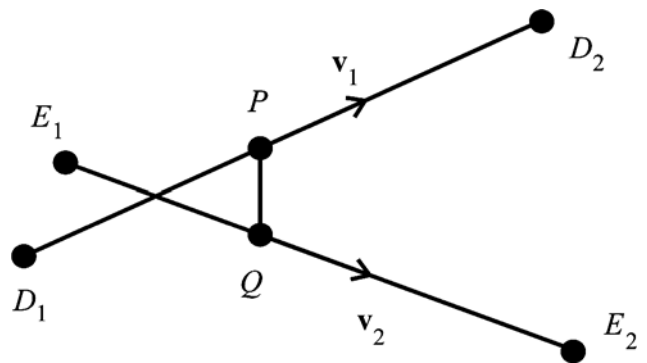


Рис. 6. Общий перпендикуляр к двум отрезкам

Так как PQ является общим перпендикуляром к двум отрезкам, то должны быть выполнены условия

$$PQ \cdot \mathbf{v}_1 = 0, \quad PQ \cdot \mathbf{v}_2 = 0. \quad (7)$$

Подставим соотношения (6) в (7). Тогда получим систему линейных уравнений относительно неизвестных параметров α и β . Если \mathbf{v}_1 и \mathbf{v}_2 не параллельны (будет верно, так как в этом случае вектор $\mathbf{v}_1 \times \mathbf{v}_2$ является невырожденным), то решением этой системы будет

$$\alpha = \frac{D_1 E_1 \cdot (\mathbf{v}_1 - (\mathbf{v}_1 \cdot \mathbf{v}_2)\mathbf{v}_2)}{1 - (\mathbf{v}_1 \cdot \mathbf{v}_2)^2}.$$

Полученный параметр α используется для вычисления точки контакта P по формуле (6).

Точка на окружности основания цилиндра. Рассмотрим контакт основания первого цилиндра с боковой поверхностью второго цилиндра. В этом случае нормалью является векторное произведение $\mathbf{n} = \mathbf{t} \times \mathbf{v}_2$, где \mathbf{t} — касательный вектор к окружности основания первого цилиндра. Задача заключается в том, чтобы найти такую точку P на окружности, касательная в которой совпадает с вектором \mathbf{t} . Для этого в качестве примера рассмотрим окружность верхнего основания цилиндра, локальная система координат (ЛСК) которой имеет центр B_1 , а оси x и y имеют направление \mathbf{u}_1 и \mathbf{w}_1 соответственно (рис. 7).

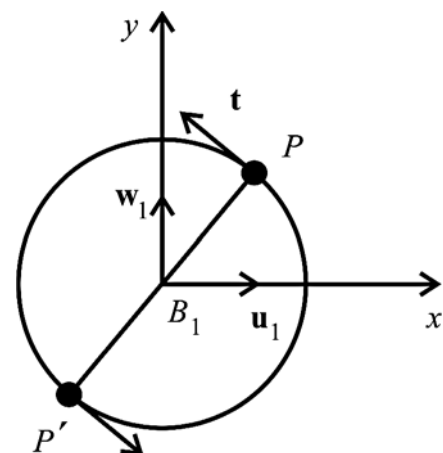


Рис. 7. Точка на окружности основания цилиндра

В ЛСК окружности вектор \mathbf{t} будет иметь координаты $t_1 = \mathbf{t} \cdot \mathbf{u}_1$ и $t_2 = \mathbf{t} \cdot \mathbf{w}_1$.

Пусть в ЛСК окружности точка P имеет координаты p_1 и p_2 . Тогда вектор \mathbf{t} будет касательным в точке P при выполнении условия $B_1 P \cdot \mathbf{t} = 0$. Так как точка P находится на окружности, то это условие приводит к системе уравнений:

$$p_1 t_1 + p_2 t_2 = 0;$$

$$p_1^2 + p_2^2 = r_1^2.$$

Эта система уравнений имеет два решения:

$$p_1 = \pm \frac{r_1}{\sqrt{1 + (t_1/t_2)^2}} \text{ и } p_2 = \mp \frac{r_1(t_1/t_2)}{\sqrt{1 + (t_1/t_2)^2}},$$

что соответствует точкам P и P' на рис. 7. Среди этих решений выбираем то, для которого выполнено условие $B_1 P \cdot \mathbf{n} > 0$.

Преобразованием в мировую систему координат точка контакта вычисляется следующим образом:

$$P = B_1 + p_1 \mathbf{u}_1 + p_2 \mathbf{w}_1.$$

Остальные случаи контакта окружности основания второго цилиндра с боковой поверхностью первого цилиндра, а также окружностей оснований двух цилиндров обрабатываются аналогичным образом.

4. Результаты

Предложенные в статье методы и алгоритмы определения коллизий аппроксимирующих цилиндров были реализованы в виде программных модулей для системы виртуального окружения VirSim [15], разработанной авторами в ФГУ ФНЦ НИИСИ РАН. Эти модули позволяют выполнять расчет коллизий виртуальных объектов, куда входит широкая и узкая фаза, построение контактного многообразия и разрешение коллизий на основе импульсного подхода. Виртуальные модели сцен и роботов создаются в системе трехмерного моделирования 3ds Max. Для этой системы был разработан плагин, с помощью которого задаются специальные динамические параметры объектов, а также создаются аппроксимирующие цилиндры, окружающие геометрию виртуальных объектов.

Апробация предложенных в статье решений проводилась на персональном компьютере с процессором Intel Core i7 (2,6 ГГц, 6 ядер) и видеоадаптером NVIDIA GeForce GTX 2070. В качестве первого примера были рассмотрены различные случаи контактного взаимодействия двух цилиндров (рис. 8). Для этого была создана виртуальная сцена, в которой первый цилиндр имеет радиус $r_1 = 30$ см и высоту $h_1 = 80$ см, а второй цилиндр — радиус $r_2 = 10$ см и высоту $h_2 = 40$ см. На рис. 8 показан результат моделирования определения коллизий этих цилиндров с отображением точек контакта. Приведены следующие случаи контакта: I — контакт основа-

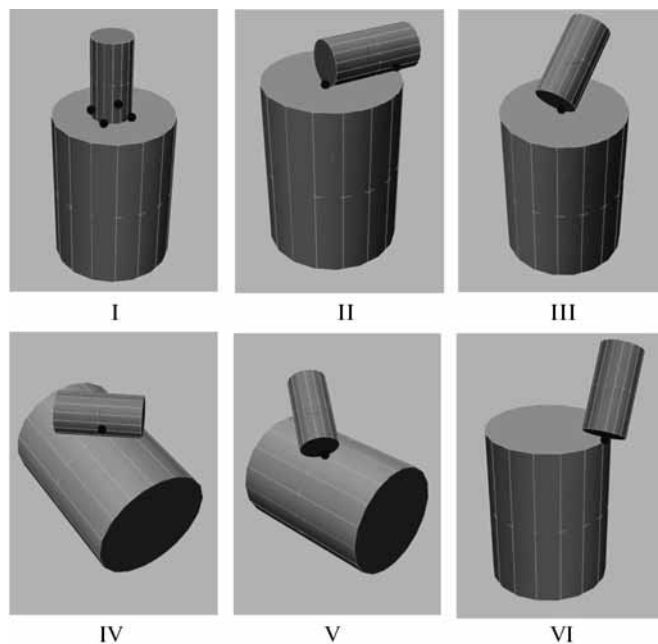


Рис. 8. Моделирование контакта двух цилиндров

ний двух цилиндров; II — основание первого цилиндра с боковой поверхностью второго цилиндра; III — основание первого цилиндра с окружностью основания второго цилиндра; IV — контакт боковых поверхностей двух цилиндров; V — боковая поверхность первого цилиндра с окружностью основания второго цилиндра; VI — контакт окружностей оснований двух цилиндров. При контакте оснований двух цилиндров среди точек контакта, вычисляемых из соотношений (5), выбираются четыре точки, которые наилучшим образом аппроксимируют пятно контакта. В остальных случаях получим одну или две точки контакта.

Во втором примере была рассмотрена тестовая сцена контактного взаимодействия нескольких цилиндров одновременно. На рис. 9 показан случай

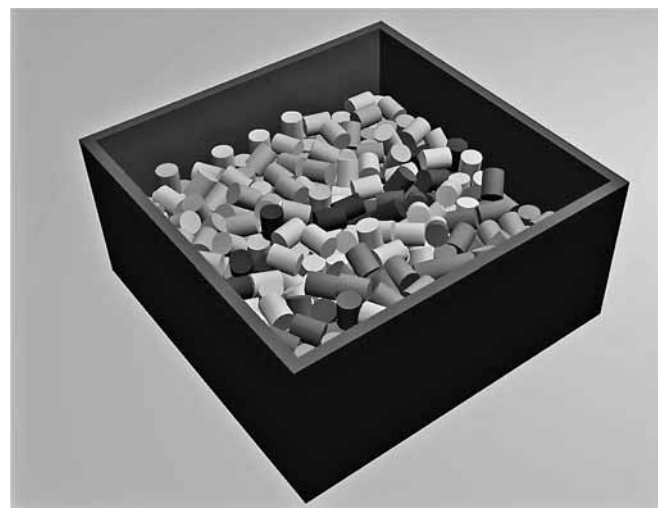


Рис. 9. Моделирование взаимодействия нескольких цилиндров

Время расчетов определения коллизий цилиндров

Число цилиндров K	Время расчетов, мс		
	$n = 8$	$n = 16$	$n = 32$
50	1	4	6
100	4	7	10
200	6	10	18
500	15	27	70
700	32	65	192
1000	57	92	256



Рис. 10. Моделирование колесного робота КРТ-100МП

взаимодействия $K = 500$ цилиндров. Результаты моделирования приведены в таблице, в которой указано максимальное время расчетов определения коллизий цилиндров в зависимости от числа цилиндров и степени n аппроксимации цилиндра призмой. Они показывают, что предложенные алгоритмы позволяют в масштабе реального времени моделировать взаимодействие около 500 цилиндров с $n = 16$.

Предлагаемые в статье решения были реализованы для моделирования виртуальных колесных роботов, где для определения коллизий колес с объектами виртуального окружения задействованы аппроксимирующие цилиндры. На рис. 10 показана виртуальная сцена, в которой рассматривается сложная динамическая ситуация, когда при движении мобильного робота КРТ-100МП его колеса сталкиваются с объектами цилиндрической формы (бочками). При этом расчет определения коллизий выполняется в масштабе реального времени, что показывает адекватность и эффективность предложенных в статье методов и алгоритмов.

Заключение

Рассмотрена задача определения коллизий аппроксимирующих цилиндров. Для ее решения разработаны методы и алгоритмы, которые основаны на применении теоремы о разделяющей оси, отбражении Гаусса и эффективных геометрических тестах для вычисления контактных точек. Полученные в работе результаты могут быть использованы для решения многих практических задач в системах виртуального окружения, имитационно-тренажерных комплексах, образовательных приложениях, анимации, компьютерных играх и т. д.

Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 19-07-00387.

Список литературы

1. Ericson C. Real-time collision detection. CRC Press, 2004. — 633 p.
2. Bergen G. Collision detection in interactive 3D environments. San Francisco, Morgan Kaufmann Publishers, 2004. — 195 p.
3. Olvang L. Real-time collision detection with implicit objects. — Department of Information Technology. — Uppsala University, Sweden, 2010. — 64 p.
4. Newth J. A. Minkowski portal refinement and speculative contacts in Box2D. Master's Projects, 2013. — 54 p.
5. Bergen G. A Fast and robust GJK Implementation for collision detection of convex objects // Journal of Graphics Tools. — 1999. — Vol. 4, No. 2. — P. 7–25.
6. Санников А. В. Численное моделирование динамики систем твердых деформируемых и жестких тел: дис. ... канд. физ.-мат. наук. Москва, 2015. — 122 с.
7. Gregorius D. Implementing quickhull. Game Developers Conference (Valve Software), San Francisco, 2014. — 119 p.
8. Eberly D. Intersection of Convex Objects: The method of separating axes. URL: <https://notendur.hi.is/jbs2/MethodOfSeparatingAxes.pdf> (дата обращения: 05.08.2021).
9. Boyd S., Vandenberghe L. Convex optimization. — Cambridge University Press, 2004. — 716 p.
10. Gregorius D. Physics for Game Programmers: The Separating Axis Test Between Convex Polyhedra // Game Developers Conference, 2013. — 52 p.
11. Chittawadigi R. G., Saha S. K. An analytical method to detect collision between cylinders using dual number algebra // Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, January 2013. — P. 5534–5538.
12. Ketchel J., Larochelle P. Collision detection of cylindrical rigid bodies for motion planning // Proceedings of the IEEE International Conference on Robotics and Automation. — 2006. — P. 1530–1535.
13. Sunkel M. Collision detection for cylinder-shaped rigid bodies. Technical report, Dept. of Informatics, University of Erlangen-Nuremberg, Erlangen, Germany, 2010. — 30 p.
14. Eberly D. Intersection of cylinders. Technical report, Magic software, 2000. — 13 p.
15. Михайлюк М. В., Мальцев А. В., Тимохин П. Ю., Страшнов Е. В., Крючков Б. И., Усов В. М. Система виртуального окружения Virsim для имитационно-тренажерных комплексов подготовки космонавтов // Пилотируемые полеты в космос. — 2020. — Т. 37, № 4. — С. 72–95.
16. Страшнов Е. В., Торгашев М. А., Мальцев А. В. Определение коллизий аппроксимирующих параллелепипедов и цилиндров // Труды НИИСИ РАН. — 2020. — Т. 10, № 5–6. — С. 139–146.
17. Bergen G., Gregorius D. Game physics pearls. AK Peters/CRC Press, 2010. — 340 p.

Collision Detection of Bounding Cylinders in Virtual Environment Systems

M. A. Torgashev, mtorg@mail.ru, E. V. Strashnov, strashnov_evlg@mail.ru, Scientific Research Institute for System Analysis of the Russian Academy of Sciences, Moscow, 117218, Russian Federation

Corresponding author:

Strashnov Evgeny V., Researcher, Scientific Research Institute for System Analysis of the Russian Academy of Sciences, Moscow, 117218, Russian Federation
E-mail: strashnov_evlg@mail.ru

Received on July 16, 2021
Accepted on August 10, 2021

The paper proposes methods and algorithms for collision detection of bounding cylinders that surround the geometry of virtual objects. The proposed solutions are based on the separating axis theorem and analysis of possible contact interaction cases between two cylinders. The idea is to approximate the cylinders by prisms with their Gauss map to the unit sphere, which reduces the number of separating axes. Also, to compute contact points, an approach was developed based on fast geometric tests, in which polygon and segment are clipping by a prism, as a common perpendicular between line segments and a point on the circle are found. Based on methods and algorithms proposed in this work, software modules were implemented. Approbation of these modules in the virtual environment system VirSim, which was developed in Scientific Research Institute for System Analysis of the Russian Academy of Sciences, showed the adequacy and effectiveness of created methods and algorithms for real-time simulation of virtual cylindrical objects. The results obtained in this work can be used to solve many practical problems in virtual environment systems, training complexes, educational applications, animation, computer games, etc.

Keywords: collision detection, bounding cylinder, prism, separating axis theorem, Gauss map, clipping, virtual environment system

Acknowledgements: This research was carried out with the financial support of the Russian Foundation for Basic Research in the framework of the scientific project № 19-07-00387.

For citation:

Torgashev M. A., Strashnov E. V. Collision Detection of Bounding Cylinders in Virtual Environment Systems, *Programmnaya Ingeneria*, 2021, vol. 12, no. 7, pp. 385–392.

DOI: 10.17587/prin.12.385-392

References

1. **Ericson C.** *Real-time collision detection*, CRC Press, 2004, 633 p.
2. **Bergen G.** *Collision detection in interactive 3D environments*, Morgan Kaufmann Publishers, San Francisco, 2004, 195 p.
3. **Olvang L.** *Real-time collision detection with implicit objects*, Department of Information Technology, Uppsala University, Sweden, 2010, 64 p.
4. **Newth J. A.** *Minkowski portal refinement and speculative contacts in Box2D*, Master's Projects, 2013, 54 p.
5. **Bergen G.** A Fast and robust GJK Implementation for collision detection of convex objects, *Journal of Graphics Tools*, 1999, vol. 4, no. 2, pp. 7–25.
6. **Sannikov A. V.** *Numerical simulation of rigid, deformable and stiff bodies*, Ph.D. dissertation, Moscow, 2015, 122 p. (in Russian).
7. **Gregorius D.** Implementing quickhull, *Game Developers Conference (Valve Software)*, San Francisco, 2014, 119 p.
8. **Eberly D.** Intersection of Convex Objects: The method of separating axes, available at: <https://notendur.hi.is/jbs2/Method-OfSeparatingAxes.pdf> (accessed 05.08.2021).
9. **Boyd S., Vandenberghe L.** *Convex optimization*, Cambridge University Press, 2004, 716 p.
10. **Gregorius D.** Physics for Game Programmers: The Separating Axis Test Between Convex Polyhedra, *Game Developers Conference*, 2013, 52 p.
11. **Chittawadigi R. G., Saha S. K.** An analytical method to detect collision between cylinders using dual number algebra, *In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, January 2013, pp. 5534–5538.
12. **Ketchel J., Laroche P.** Collision detection of cylindrical rigid bodies for motion planning, *In Proceedings of the IEEE International Conference on Robotics and Automation*, 2006, pp. 1530–1535.
13. **Sunkel M.** *Collision detection for cylinder-shaped rigid bodies*, Technical report, Dept. of Informatics, University of Erlangen-Nuremberg, Erlangen, Germany, 2010, 30 p.
14. **Eberly D.** *Intersection of cylinders*, Technical report, Magic software, 2000, 13 p.
15. **Mikaylyuk M. V., Maltsev A. V., Timokhin P. Yu., Strashnov E. V., Kryuchlov B. I., Usov V. M.** The VirSim virtual environment system for the simulation complexes of cosmonaut training, *Pilotiruemye polety v kosmos*, 2020, vol. 37, no. 4, pp. 72–95 (in Russian).
16. **Strashnov E. V., Torgashev M. A., Maltsev A. V.** Collision detection of bounding boxes and cylinders, *Trudy NIISI RAN*, 2020, vol. 10, no. 5–6, pp. 139–146 (in Russian).
17. **Bergen G., Gregorius D.** *Game physics pearls*, AK Peters/CRC Press, 2010, 340 p.

ООО "Издательство "Новые технологии". 107076, Москва, ул. Матросская Тишина, 23, с. 2, оф. 45
Технический редактор *Е. М. Патрушева*. Корректор *Е. В. Комиссарова*

Сдано в набор 11.08.2021 г. Подписано в печать 00.09.2021 г. Формат 60×88 1/8. Заказ PI721
Цена свободная.

Оригинал-макет ООО "Авансд солишнз". Отпечатано в ООО "Авансд солишнз".
119071, г. Москва, Ленинский пр-т, д. 19, стр. 1. Сайт: www.aov.ru

Рисунок к статье Д. Д. Руховича
«ПРОЕКЦИЯ ИЗОБРАЖЕНИЙ В ПРОСТРАНСТВО
ДЛЯ МОНОКУЛЯРНОЙ И МУЛЬТИРАКУРСНОЙ
ТРЕХМЕРНОЙ ДЕТЕКЦИИ ОБЪЕКТОВ В УЛИЧНЫХ СЦЕНАХ»



Рис. 4. Объекты, обнаруженные с помощью предложенного метода по мультиракурсному входу с шести камер из валидационной части набора данных nuScenes.

Для каждого изображения указана ориентация камеры:

a – вперед; *б* – вперед и вправо; *в* – назад и вправо; *г* – назад; *д* – назад и влево; *е* – вперед и влево

Издательство «НОВЫЕ ТЕХНОЛОГИИ»

выпускает научно-технические журналы



Теоретический и прикладной научно-технический журнал

ПРОГРАММНАЯ ИНЖЕНЕРИЯ

В журнале освещаются состояние и тенденции развития основных направлений индустрии программного обеспечения, связанных с проектированием, конструированием, архитектурой, обеспечением качества и сопровождением жизненного цикла программного обеспечения, а также рассматриваются достижения в области создания и эксплуатации прикладных программно-информационных систем во всех областях человеческой деятельности.

Подписной индекс по Объединенному каталогу
«Пресса России» – 22765

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ



Ежемесячный теоретический и прикладной научно-технический журнал

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

В журнале освещаются современное состояние, тенденции и перспективы развития основных направлений в области разработки, производства и применения информационных технологий.

Подписной индекс по
Объединенному каталогу
«Пресса России» – 72656

Междисциплинарный теоретический и прикладной научно-технический журнал

НАНО- и МИКРОСИСТЕМНАЯ ТЕХНИКА

В журнале освещаются современное состояние, тенденции и перспективы развития нано- и микросистемной техники, рассматриваются вопросы разработки и внедрения нано микросистем в различные области науки, технологии и производства.



Подписной индекс по
Объединенному каталогу
«Пресса России» – 79493



Ежемесячный теоретический и прикладной научно-технический журнал

МЕХАТРОНИКА, АВТОМАТИЗАЦИЯ, УПРАВЛЕНИЕ

В журнале освещаются достижения в области мехатроники, интегрирующей механику, электронику, автоматику и информатику в целях совершенствования технологий производства и создания техники новых поколений. Рассматриваются актуальные проблемы теории и практики автоматического и автоматизированного управления техническими объектами и технологическими процессами в промышленности, энергетике и на транспорте.

Подписной индекс по
Объединенному каталогу
«Пресса России» – 79492

Научно-практический и учебно-методический журнал

БЕЗОПАСНОСТЬ ЖИЗНЕДЕЯТЕЛЬНОСТИ

В журнале освещаются достижения и перспективы в области исследований, обеспечения и совершенствования защиты человека от всех видов опасностей производственной и природной среды, их контроля, мониторинга, предотвращения, ликвидации последствий аварий и катастроф, образования в сфере безопасности жизнедеятельности.



Подписной индекс по
Объединенному каталогу
«Пресса России» – 79963

Адрес редакции журналов для авторов и подписчиков:

107076, Москва, ул. Матросская Тишина, д. 23, стр. 2, оф. 45. Издательство "НОВЫЕ ТЕХНОЛОГИИ".

Тел.: (499) 270-16-52. E-mail: antonov@novtex.ru