



# GraphiCon 2020

30-я Международная конференция  
по компьютерной графике  
и машинному зрению

## Труды конференции Том 1

# GraphiCon 2020

30th International Conference  
on Computer Graphics and Vision

## Conference Proceedings Part 1

---

Национальный исследовательский университет ИТМО

Санкт-Петербург, 22-25 сентября 2020  
<https://graphicon.itmo.ru/>

УДК 004.92  
ББК 32.973.26-018.2

**GraphiCon 2020:** труды 30-й Международной конференции по компьютерной графике и машинному зрению (Санкт-Петербург 22–25 сентября 2020 г.). Том 1 / Институт прикладной математики имени М. В. Келдыша РАН. – Санкт-Петербург, 2020. – 151 с.

ISSN 2618-8317

Настоящий сборник содержит материалы 30-й Международной конференции по компьютерной графике и машинному зрению ГрафиКон-2020. В ее подготовке и проведении участвовали специалисты ведущих организаций, отраслевых и академических научно-исследовательских институтов, а также высшей школы.

Конференция ГрафиКон ведет свою историю с 1991 года и является крупнейшей в России и странах СНГ научно-дискуссионной площадкой в области методов и технологий компьютерного анализа изображений, визуальной и когнитивной аналитики, 3D-реконструкции, визуальной навигации и человеко-машинного взаимодействия, виртуальной и дополненной реальности, распознавания образов и др.

Сборник издается в двух томах. В первый том сборника включены секционные доклады конференции, представленные на русском языке. Сборник адресуется сотрудникам научно-исследовательских и образовательных организаций, специалистам предприятий IT-индустрии, аспирантам, студентам, а также широкому кругу читателей.

Мероприятие проведено при финансовой поддержке РФФИ, проект № 20-07-22052.

**УДК 004.92**  
**ББК 32.973.26-018.2**

ISSN 2618-8317

# ОРГАНИЗАЦИОННЫЙ КОМИТЕТ

Кустарев Павел Валерьевич, Университет ИТМО	Ерёмченко Евгений Николаевич, МГУ им. М.В. Ломоносова	Палташев Тимур Турсунович, AMD
Бахолдин Алексей Валентинович, Университет ИТМО	Жданов Дмитрий Дмитриевич, Университет ИТМО	Подвесовский Александр Георгиевич, БГТУ
Берберова Мария Александровна, МФТИ	Захарова Алёна Александровна, БГТУ	Ротков Сергей Игоревич, ННГАСУ
Бондарев Александр Евгеньевич, ИПМ им. М.В. Келдыша РАН	Ильина Аглая Геннадьевна, Университет ИТМО	Смолин Артем Александрович, Университет ИТМО
Будак Владимир Павлович, МЭИ	Карпов Леонид Евгеньевич, ИСП РАН	Толок Алексей Вячеславович, ИПУ РАН, МГТУ «СТАНКИН»
Быковский Сергей Вячеславович, Университет ИТМО	Конушин Антон Сергеевич, МГУ им. М.В. Ломоносова	Точилина Татьяна Вячеславовна, Университет ИТМО
Волобой Алексей Геннадьевич, ИПМ им. М.В. Келдыша РАН	Муромцев Дмитрий Ильич, Университет ИТМО	Турлапов Вадим Евгеньевич, ННГУ им. Н.И. Лобачевского
Галактионов Владимир Александрович, ИПМ им. М.В. Келдыша РАН	Крылов Андрей Серджевич, МГУ имени М.В. Ломоносова	

# ПРОГРАММНЫЙ КОМИТЕТ

Жданов Дмитрий  
Дмитриевич, Университет  
ИТМО

Авербух Владимир  
Лазаревич,  
ИММ УрО РАН

Багдасаров Александр  
Аванесович, АО «ГОИ им.  
С.И. Вавилова»

Барладян Борис Хаимович,  
ИПМ им. М.В. Келдыша  
РАН

Бахолдин Алексей  
Валентинович,  
Университет ИТМО

Белокаменная Александра  
Александровна, ННГУ им.  
Н.И.Лобачевского

Бобков Валерий  
Александрович, ИАПУ  
ДВО РАН

Бондарев Александр  
Евгеньевич, ИПМ им. М.В.  
Келдыша РАН

Будак Владимир Павлович,  
МЭИ

Васильев Евгений  
Павлович, ННГУ им.  
Н.И.Лобачевского

Ватолин Дмитрий  
Сергеевич, МГУ им. М.В.  
Ломоносова

Вехтер Евгения  
Викторовна,  
ТПУ

Вознесенская Анна  
Олеговна,  
Университет ИТМО

Волобой Алексей  
Геннадьевич, ИПМ им.  
М.В. Келдыша РАН

Гаврилина Ольга  
Алексеевна,  
Университет ИТМО

Галактионов Владимир  
Александрович, ИПМ им.  
М.В. Келдыша РАН

Гергет Ольга Михайловна,  
ТПУ

Давыдов Сергей  
Васильевич,  
БГТУ

Дебелов Виктор  
Алексеевич,  
ИВМиМГ СО РАН

Дубнищев Юрий  
Николаевич, Институт  
теплофизики СО РАН

Ежова Ксения Викторовна,  
Университет ИТМО

Ерёмченко Евгений  
Николаевич, МГУ им. М.В.  
Ломоносова

Ершов Сергей  
Валентинович, ИПМ им.  
М.В. Келдыша РАН

Захарова Алёна  
Александровна,  
БГТУ

Знаменская Ирина  
Александровна,  
МГУ

Игнатенко Алексей  
Викторович, МГУ им. М.В.  
Ломоносова

Исаев Руслан  
Александрович,  
БГТУ

Киричек Андрей  
Викторович,  
БГТУ

Конушин Антон  
Сергеевич, МГУ им. М.В.  
Ломоносова

Коростелев Дмитрий  
Александрович,  
БГТУ

Кретинин Олег  
Васильевич, НГТУ им.  
Р.Е.Алексеева

Крылов Андрей  
Серджевич, МГУ имени  
М.В. Ломоносова

Кузьменко Александр  
Анатольевич,  
БГТУ

Лагерев Дмитрий  
Григорьевич,  
БГТУ

Макаренко Константин  
Васильевич,  
БГТУ

Мальшев Александр  
Сергеевич,  
(Нижегородское  
подразделение OPEN  
CASCADE S.A.S.,  
Cargemini Group, France)

Михайлюк Михаил  
Васильевич,  
НИИСИ РАН

Насонов Андрей  
Владимирович, МГУ  
имени М.В. Ломоносова

Небаба Степан  
Геннадьевич,  
ТПУ

Носова Светлана  
Александровна, ННГУ им.  
Н.И.Лобачевского

Пак Александр Яковлевич,  
ТПУ

Пилюгин Виктор  
Васильевич,  
МИФИ

Подвесовский Александр  
Георгиевич, БГТУ

Потемин Игорь  
Станиславович,  
Университет ИТМО

Ринкевичюс Бронюс  
Симович,  
МЭИ

Романова Галина  
Эдуардовна, Университет  
ИТМО

Ротков Сергей  
Игоревич, ННГАСУ

Рябинин Константин  
Валентинович, ПГНИУ

Сивкова Татьяна  
Александровна,  
ООО «СИАМС»

Сляднев Сергей  
Евгеньевич,  
(Нижегородское  
подразделение OPEN  
CASCADE  
S.A.S., Cargemini Group,  
France)

Смирнова Ирина  
Геннадьевна, Университет  
ИТМО

Сорокин Дмитрий  
Васильевич, МГУ имени  
М.В. Ломоносова

Спасенников Валерий  
Валентинович,  
БГТУ

Толок Алексей  
Вячеславович, ИПУ РАН,  
МГТУ «СТАНКИН»

Точилина Татьяна  
Вячеславовна,  
Университет ИТМО

Трубаков Андрей  
Олегович, БГТУ

Трубаков Евгений  
Олегович, БГТУ

Турлапов Вадим  
Евгеньевич, ННГУ им.  
Н.И. Лобачевского

Фролов Владимир  
Александрович, МГУ им.  
М.В. Ломоносова

Хацевич Татьяна  
Николаевна, СГУГиТ,  
Новосибирск

Хвостиков Александр  
Владимирович, МГУ  
имени М.В. Ломоносова

Цыганок Елена  
Анатольевна,  
Университет ИТМО

Чертов Александр  
Николаевич,  
Университет ИТМО

Чуприна Светлана  
Игоревна, ПГНИУ

Шабурова Аэлиа  
Владимировна,  
СГУГиТ, Новосибирск

Шапиро Лев Залманович,  
ИПМ им. М.В. Келдыша  
РАН

Шкляр Алексей  
Викторович, ТПУ

# Содержание

## Компьютерная графика в материаловедении, светотехнике и дизайне

*Ф.И. Маняхин, Р.В. Левченко, Л.О. Мокрецова*

Метод фотоответного изображения для диагностики локальной эффективности солнечных батарей ..... 8

*Т. Сивкова, С. Губарев, И. Каменин*

Автоматизированный анализ микроструктуры материалов. Анализ изображений с наличием дефектов пробоподготовки ..... 15

## Научная визуализация и визуальная аналитика

*А.А. Батюков, Е.В. Попов, Н. Фогт*

Визуализация многомерных поверхностей и их экстремумов ..... 28

*В. Борисов, А. Давыдов, Т. Константиновская, А. Луцкий*

Визуализация двух противоположно вращающихся сверхзвуковых вихрей ..... 40

## Геометрическое моделирование. Компьютерная графика в образовании

*С.А. Рязанов*

Геометрическая модель формообразования рабочих поверхностей элементов конической передачи ..... 48

*Е.В. Тетюев, Ю.А. Жук*

Использование WebGL для визуализации учебных материалов в образовательном процессе ..... 59

*С.А. Роменский, С.И. Ротков*

Формирование трехмерной каркасной модели в проблеме преобразования чертежно-конструкторской документации на бумажном носителе в электронную модель объекта ..... 67

*Д.В. Волошинов, К.Н. Соломонов*

Программно-аппаратная реализация конструктивных геометрических моделей ..... 83

## Компьютерное зрение

*А.П. Широков, Д.А. Купляков, А.С. Конушин*

Алгоритм подсчета автомобилей в крупномасштабных системах видеонаблюдения ..... 100

*А.И. Гайош*

Роль HD-карт в беспилотном транспорте ..... 109

*А.Л. Липкина, Л.М. Местецкий*

Метод распознавания шрифтов на основе медиального представления ..... 118

## Вычислительная оптика и оптический дизайн

*Д.Е. Трошкин, А.Н. Чертов, Е.В. Горбунова, А.А. Алёхин*

Возможности цифрового контроля и классификации зерновых культур ..... 131

*А.Н. Марковский*

Полигармоническое разложение цифрового изображения ..... 144



**КОМПЬЮТЕРНАЯ ГРАФИКА В  
МАТЕРИАЛОВЕДЕНИИ,  
СВЕТОТЕХНИКЕ И ДИЗАЙНЕ**

---

**COMPUTER GRAPHICS IN  
MATERIALS SCIENCE, LIGHTING  
AND DESIGN**

**GraphiCon 2020**

---

22-25 сентября 2020  
Санкт-Петербург, Россия

## Метод фотоответного изображения для диагностики локальной эффективности солнечных батарей\*

Ф.И. Маняхин<sup>1</sup>, Р.В. Левченко<sup>2</sup>, Л.О. Мокрецова<sup>3</sup>

<sup>1</sup> НИТУ «МИСиС», Москва, Россия  
zaomisis@yandex.ru

<sup>2</sup> НИТУ «МИСиС», Москва, Россия  
LionchenkoR@gmail.com

<sup>3</sup> НИТУ «МИСиС», Москва, Россия  
lgmisis@yandex.ru

**Аннотация.** Метод фотоответного изображения направлен на физико-математическое исследование текущего состояния солнечных элементов (СЭ). Разработано устройство для снятия показателей местной эффективности СЭ. Сканирование происходит с помощью монитора персонального компьютера. Связь с устройством происходит с помощью специального программного обеспечения. Программа позволяет настраивать диагностику, обработку и хранение данных с помощью пользовательского интерфейса. Конфигурация включает выбор алгоритма диагностики, количества циклов чтения данных и цвет светового пятна. Результаты эксперимента отображаются на экране монитора в матричном виде и визуализируются в виде изображения. Программа предназначена для работы с ОС Windows. Экспериментальные испытания проводились на монокристаллических солнечных элементах на основе кремния или перовскита.

**Ключевые слова:** метод фотоответного изображения, солнечный элемент, фотосканирующее устройство, диагностика.

### 1 Компьютерная графика в материаловедении, светотехнике и дизайне

#### 1.1 Компьютерный анализ изображений микро- и наноструктуры материалов

От начального качества солнечных батарей (СБ) в значительной степени зависит срок их службы и коэффициент полезного действия. Интегральными параметрами качества СБ являются ускоренные тесты старения (PID test) и коэффициент заполнения вольт-амперной (ВАХ) характеристики (Fill factor – FF). FF –

---

\* Работа выполнена при государственной поддержке, оказываемой в рамках Программы повышения конкурентоспособности Национального научно-технического университета «МИСиС».



это отношение реальной мощности элемента к его гипотетической мощности. Данный показатель является одним из основных параметров, по которому можно судить о качестве фотоэлектрического модуля.

Но, очевидно, что эти параметры зависят от микроскопического состояния структуры активной области солнечного элемента (СЭ), особенно на основе поликристаллического или аморфного материала: проколов, фоторезистивных включений, микро царапин и пр.

Эти дефекты возникают уже на стадии технологического процесса вследствие его нарушения. Поэтому экономически целесообразно в технологическую цепочку включить операцию экспрессконтроля дифференциальных параметров качества СЭ, таких как локальной эффективности, наличия проколов и прочее.

Исследованию локальных дефектов изделий электронной техники посвящено много работ. Разработаны разнообразные методы диагностики. Наиболее доступными и дешевыми методами являются методы, основанные на использовании воздействия света на локальные активные области полупроводниковых структур.

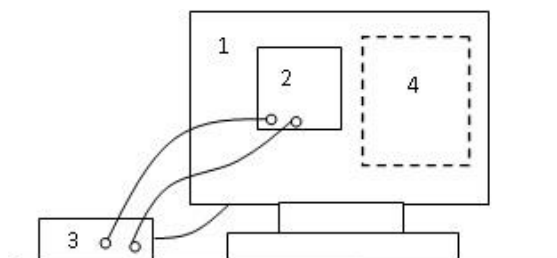
Горюновым Н.Н. в соавторстве был разработан метод, который был назван фотосканированием, на основе которого разработан ряд устройств для диагностики качества полупроводниковых изделий [1-3]. Все эти устройства содержат оптическую систему на основе излучателя, свет которого фокусируется с помощью оптической системы на локальную область поверхности полупроводниковой структуры. Микроскопическое световое пятно генерирует в активной области полупроводникового кристалла фототок, который преобразуется в видимое изображение на экране монитора в виде областей различной яркости. По этому изображению делается вывод о качестве состояния структуры. Сканирование светового пятна осуществляется различными способами: отражением телевизионной развертки, механическими устройствами. Основными недостатками указанных устройств являются большие габариты, аналоговое изображение и необходимость синхронизации сканирования с изображением. Кроме того, невозможность количественной обработки параметров изображения.

В [4] описано фотосканирующее устройство, позволяющее получать цифровой сигнал фотоответа в виде прямоугольной матрицы. Однако это устройство имеет достаточно большие габариты фокусирующей системы.

В настоящей работе представлено фотосканирующее устройство для диагностики качества СЭ и СБ на основе анализа микроскопических параметров. Устройство создано на базе персонального компьютера (ПК), в котором сканирующее световое пятно, создаваемое на мониторе ПК, засвечивает малые области СЭ, наложенного чувствительной поверхностью на экран. Сигнал фотоответа записывается в прямоугольную матрицу, формируемую в памяти ПК, рис.1. Управление процессом фотосканирования ведется контроллером Arduino Uno R3 на базе микроконтроллера ATmega328.

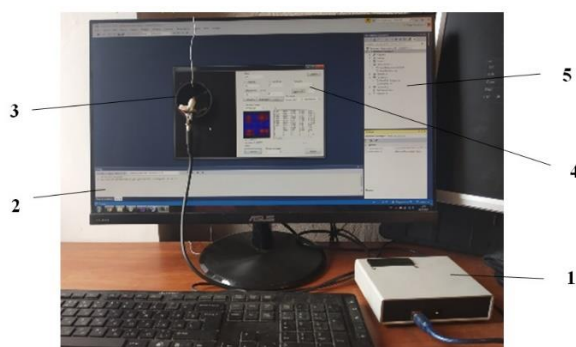
Характеристики:

- Микроконтроллер АТmega328;
- Напряжение питания 5В;
- Входное напряжение (рекомендуемое) 7-12В;
- Входное напряжение (предельное) 6-20В;
- Цифровой ввод-вывод 14 линии (6 поддерживают ШИМ);
- Аналоговый ввод 8 линий;
- Постоянный ток на линиях ввода-вывода 40мА;
- Постоянный ток на линии 3.3В 50мА;
- Flash-память 32КВ, 2 КВ из них использованы для загрузчика;
- SRAM-память 2КВ;
- EEPROM-память 1КВ;
- Тактовая частота 16МГц.



**Рис. 1.** Схематическое изображение устройства фотосканирования поверхности солнечного элемента: 1 – Экран монитора. 2 – Солнечный элемент. 3 – Контроллер. 4 – Область отображения информации

Фотография устройства представлена на рис.2.



**Рис. 2.** Макет фотосканирующего устройства. 1. Микроконтроллерный блок; 2. Монитор ПК; 3. Солнечный элемент; 4. Панель отображения фотоскана; 5. Панель отражения режимов сканирования

Контроллер подключается к ПК через COM порт кабелем USB-AB 2.0.

Управление аппаратными средствами происходит с применением программного обеспечения, которое может быть установлено на ПК с операционными системами Windows XP, Windows 7, 8 и 10. Сбор данных о локальной эффективности СЭ происходит при взаимодействии аппаратных средств с СЭ. Обработка и хранение данных происходит программно. В конце цикла тестирования СЭ дифференциальная эффективность СЭ отображается на мониторе в области информации (рис. 3).

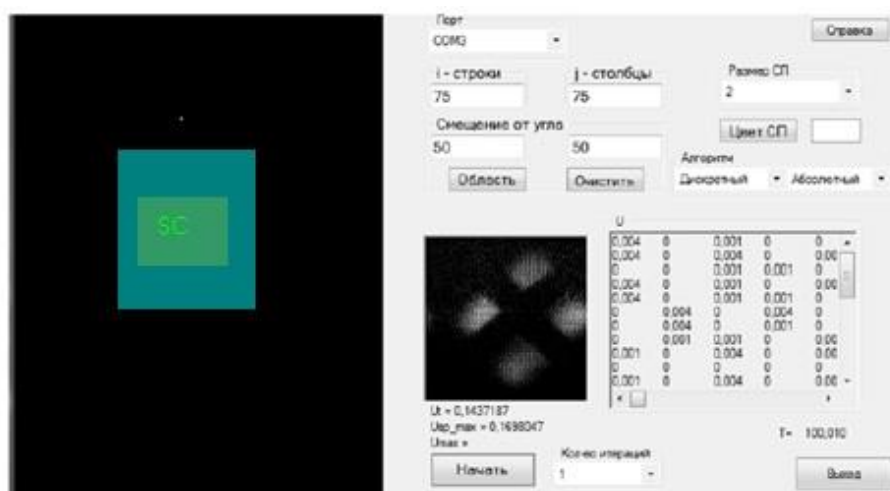


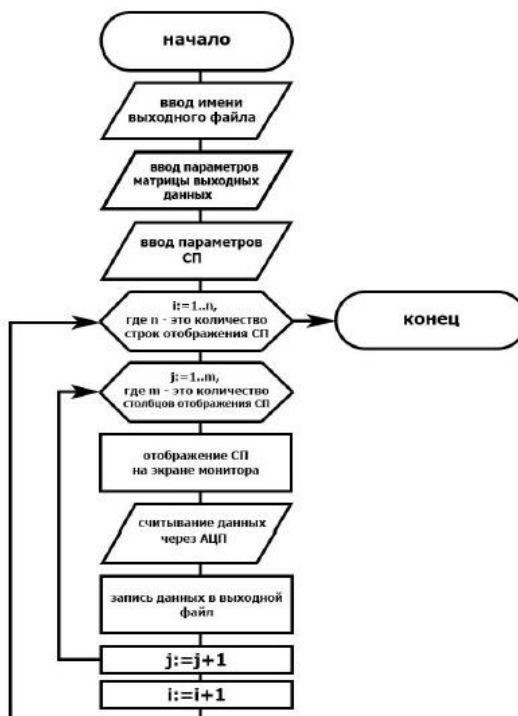
Рис. 3. Интерфейс программы диагностики СЭ

Блок-схема программы управления и обработки результатов фотосканирования приведена на рис. 4.

На этапе обработки информации использован инструментальный математический САПР программного продукта MathCad. С помощью нее осуществляется анализ матричной информации – дифференциальная диагностика эффективности СЭ: наличие локальных "проколов", утечки, шунты. Устройство реализует следующие возможности: минимизацию ошибки измерения обнулением значений зоны нечувствительности, введение контраста, измерение локальных фототоков на уровне  $1 \cdot 10^{-8}$  А.

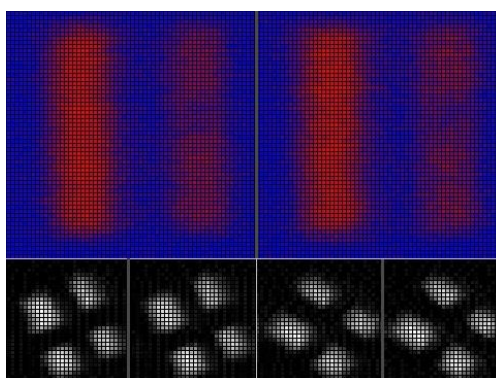
Для измерения малых фототоков СЭ большой площади и устранения влияния фоновой засветки LED – экрана применена схема модуляции интенсивности светового пятна монитора. Модуляция осуществляется программным способом с частотой 20 Гц. Фотоответный сигнал усиливается селективным усилителем и выпрямляется с помощью интегратора, выполняющего роль синхронного детектора. Оба каскада выполнены на операционных усилителях УД17.

Размер светового пятна устанавливается числом светящихся пикселей.



**Рис. 4.** Блок-схема программы управления, сбора и обработки данных фотосканирования СЭ

Результат эксперимента по получению фототока от монокристаллической солнечной панели на кремнии показывает повторяемость результатов (рис. 5). На данный момент ведется работы по устранению шума, который можно увидеть на изображении визуализации результатов.



**Рис. 5.** Визуализация результатов получения фототока из монокристаллического кремния СЭ

Сканирование принимающей поверхности можно осуществлять двумя способами.

При первом способе измеряется фототок в локальной области светового пятна. При втором – измеряется фототок в СЭ при пошаговом увеличении площади засветки на площадь единичного пятна.

В первом случае матрица фотоответного изображения отражает локальную (дифференциальную) фоточувствительность СЭ. При втором – интегрированный фотоответ. При втором способе можно выявить области так называемых «проколов». Проколы – это места в СЭ, где отсутствует р-п переход и эта область обладает свойствами фоторезистора. При засветке этой области ее сопротивление уменьшается и возрастают утечки по этому шунту. При наличии таких шунтов коэффициент полезного действия СЭ уменьшается тем больше, чем больше мощность освещения.

Другим положительным качеством предлагаемого устройства является возможность использовать засветку световым пятном с различной длиной волны: красного, зеленого и синего диапазонов спектра. Это позволяет исследовать качество многослойных плёночных структур.

Для уменьшения расходимости света от пикселя монитора необходимо использовать сетку между монитором и СЭ.

## 2 Заключение

1. Разработано устройство экспресс диагностики качества солнечных элементов, в котором в качестве фотосканирующего устройства используется монитор персонального компьютера. Персональный компьютер выполняет роль управляющего устройства и устройства обработки и хранения экспериментальных данных;
2. Предлагаемое устройство может быть использовано в технологической цепочке промежуточного или конечного контроля качества технологического процесса производства СЭ.

Работа выполнена при государственной поддержке, оказываемой в рамках Программы повышения конкурентоспособности Национального научно-технического университета «МИСиС» среди ведущих мировых научных и образовательных центров на 2013-2020 годы.

## Литература

1. Амазаспян В.Н., Саркисян В.С., Горюнов Н.Н. Устройство для выявления дефектов поверхности полупроводниковых приборов // Авторское свидетельство СССР SU 630983 A1 (1985).
2. Розинков Н.С., Лонской И.И., Дубицкий Л.Г., Горюнов Н.Н. Устройство дефектоскопического контроля планарных структур // Авторское свидетельство СССР 813202 (1981).

3. Лонской И.И., Горюнов Н.Н., Каверзнев В.А. Устройство для оптического контроля интегральных схем // Авторское свидетельство СССР 594477 (1978).
4. Григорян В.Г., Горюнов Н.Н., Маняхин Ф.И., Бессонов В.А., Кодак А.С. Исследование качества полупроводниковых структур метода фотоответственного изображения // Технологии электронной промышленности №1, 72-75 (2005).

## Автоматизированный анализ микроструктуры материалов. Анализ изображений с наличием дефектов пробоподготовки\*

Т. Сивкова<sup>1</sup>, С. Губарев<sup>2</sup>, И. Каменин<sup>3</sup>

<sup>1</sup> ООО «СИАМС», 620078, Екатеринбург, Россия  
sivkova@siams.com

<sup>2</sup> ООО «СИАМС», 620078, Екатеринбург, Россия  
mnogono@gmail.com

<sup>3</sup> ООО «СИАМС», 620078, Екатеринбург, Россия  
igor@siams.com

**Аннотация.** В продолжение темы, поднятой авторами статьи на 29-й Международной конференции по компьютерной графике и машинному зрению 23–26 сентября 2019г., в данной статье вновь рассматриваются вопросы распознавания элементов микроструктуры материалов, наиболее часто встречающиеся проблемы и ошибки распознавания, способы их решения. Показаны способы восстановления формы частиц после удаления артефактов на изображении, разделения соприкасающихся частиц. Описаны примеры практического применения разработанных алгоритмов для анализа частиц в микроструктуре графита в чугунах и частиц порошка. Показана возможность количественного анализа изображений, содержащих дефекты.

**Ключевые слова:** цифровая обработка изображений, дефекты пробоподготовки, дилатация, контур частиц, разделение частиц.

### 1 Введение

Известно, что системы анализа изображений требуют высокого качества подготовки поверхности микрошлифа для того, чтобы дефекты пробоподготовки, такие как, пятна от реактива, воды, суспензии, царапины, не были распознаны программой как объекты контроля. Но даже использование современных автоматических станков не дает полной гарантии получения идеальной поверхности. С учетом того, что в России все еще большое количество лабораторий продолжают применять ручные шлифовально-полировальные станки наряду с внедрением компьютерной обработки изображений, проблема автоматического распознавания дефектов становится все более актуальной.

---

\* Публикация выполнена при поддержке гранта РФФИ №20-01-00547

Отдельного внимания заслуживает анализ порошков. Для качественного распознавания частицы должны быть хорошо диспергированы. Однако не всегда удается получить хорошо подготовленную пробу. На изображениях часто встречаются агломераты частиц, которые приходится разделять программным способом для того, чтобы не вносить погрешность при вычислении количественных характеристик частиц.

## 2 Алгоритмы борьбы с дефектами на изображениях

Применение любого алгоритма первичной сегментации требует уточнения полученного результата, так как в большинстве случаев совместно с объектами контроля выделяются лишние объекты - «мусор», или связанная компонента, представляющая собой целый агломерат объектов контроля. Перед тем, как решать задачу классификации выделенных объектов, необходимо провести эффективное удаление заведомо ложных объектов.

Существует базовый алгоритм бинарного морфологического открытия и закрытия [1], который позволяет отфильтровать мелкие и тонкие объекты, а также разделить соприкасающиеся объекты. Описанное преобразование имеет два существенных недостатка: оно приводит к изменению формы объектов контроля, а также к разделению целых объектов вместе с объектами, подлежащими разделению.

Ниже описываются подходы, позволяющие восстановить исходную форму объектов контроля, разделить связанные компоненты, уменьшив тем самым негативные последствия проведенной морфологической фильтрации.

### 2.1 Морфологическая реконструкция через дилатацию

#### Устранение влияния дефектов в виде пятен.

Применительно к анализу частиц графита в чугуне операция морфологического открытия эффективно работает для структуры, содержащей крупные частицы шаровидного графита. При анализе структуры с тонкими частицами пластинчатого или вермикулярного графита применение операции открытия приводит к удалению вместе с «мусором» тонких фрагментов частиц, сопоставимых по размеру с применяемым морфологическим шаблоном.

Рассмотрим пример анализа «дефектного» изображения микроструктуры чугуна, которое при классическом подходе к анализу изображений считается непригодным для цифровой обработки (рис.1).



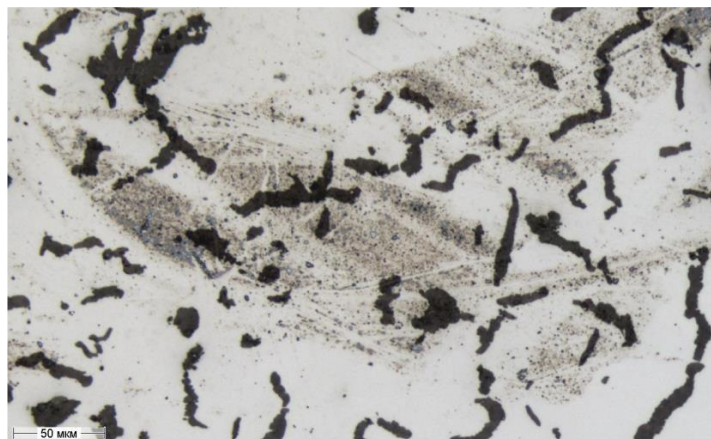


Рис. 1. Исходное изображение

При пороговой сегментации изображения вместе с частицами выделяются и загрязнения - «мусор», подлежащий последующему удалению из числа объектов контроля (рис. 2).

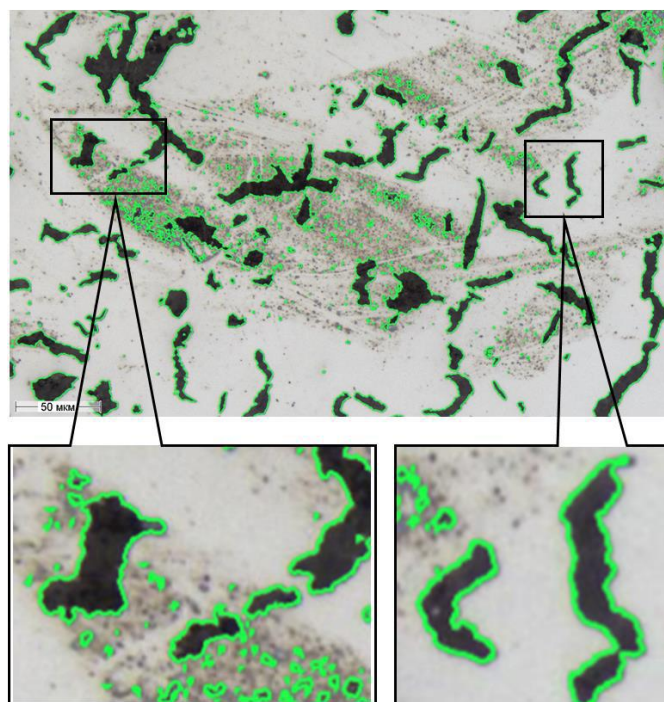
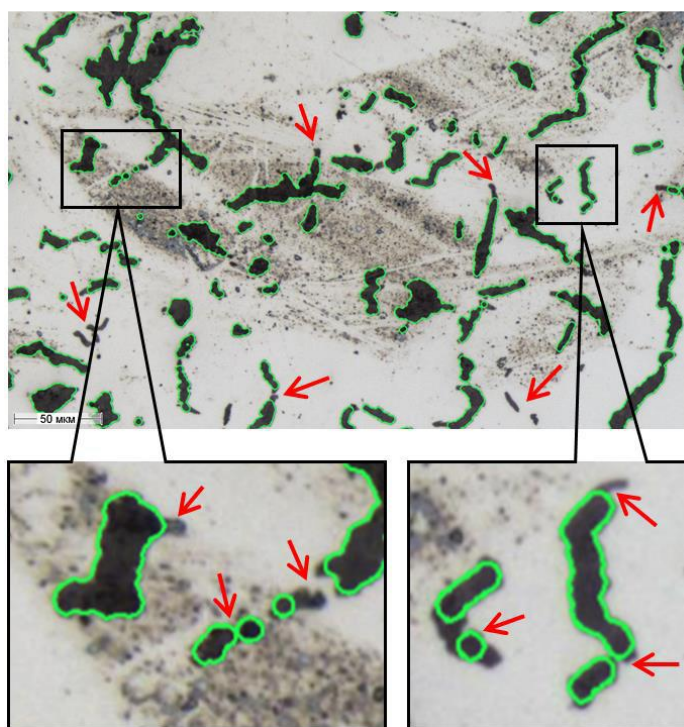


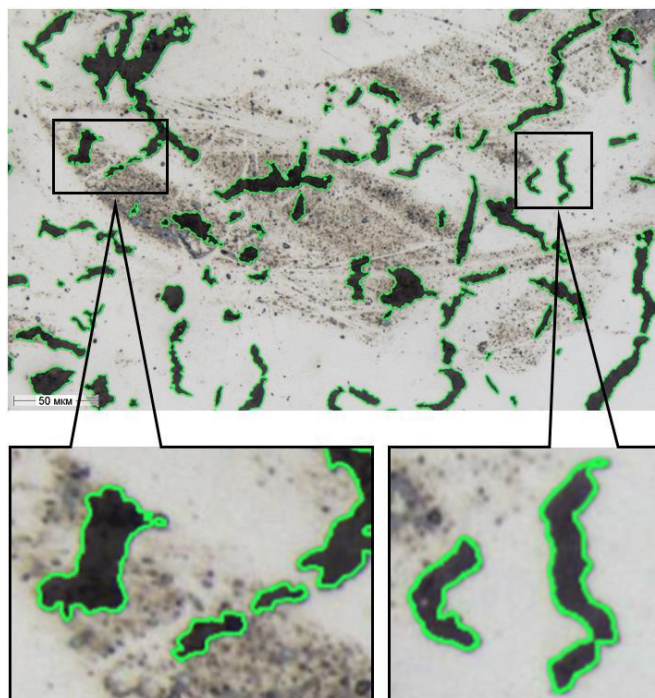
Рис. 2. Результат пороговой сегментации. Одновременное выделение частиц и дефектов изображения

Применение простой фильтрации частиц по размеру не будет эффективным для подобных загрязнений, т.к. при этом останутся загрязнения, объединенные с частицами. Применение функции морфологического открытия для исключения «мусора» одновременно приводит к нежелательному удалению тонких частиц или их тонких фрагментов и, как следствие, к дроблению целых частиц, искажая форму и размеры частиц (рис. 3).

Программное обеспечение SIAMS использует комплекс функций, включающий морфологическую реконструкцию через дилатацию, который позволяет избавиться от «мусора» при выделении частиц графита, полностью сохранив исходный контур частиц (рис. 4). В сочетании с алгоритмами сегментации, описанными авторами статьи [2], это позволяет обрабатывать изображения, содержащие дефекты в виде различных пятен с минимальными потерями в точности количественного анализа частиц графита.



**Рис. 3.** Результат морфологического открытия. Удаление дефектов из числа объектов контроля с искажением контура частиц



**Рис. 4.** Результат применения «умной» дилатации. Удаление дефектов из числа объектов контроля с сохранением контура частиц

Морфологическая реконструкция через дилатацию - операция морфологической дилатации с ограничением  $R$  [1]. Результатом такой операции является выполнение дилатации множества  $A$  на шаблон  $P$  и объединение с множеством  $B$  с учетом связности компонент (1):

$$R = (A \oplus P) \cap B, \quad (1)$$

где  $B$  – исходное множество (все выделенные объекты на изображении);  
 $A$  – множество, полученное в результате применения морфологического открытия;  
 $P$  – шаблон дилатации заданного размера в форме круга.

Таким образом, в результирующем множестве присутствуют только компоненты, пересекающиеся с множеством  $B$ .

Операция геодезической дилатации [1] с учетом связности компонент множества  $B$ , представляет собой комплекс функций, предназначенный для очистки изображения от ненужных объектов с сохранением исходного контура объектов контроля, названа разработчиками программного обеспечения SIAMS «умной дилатацией».

### Устранение влияния дефектов в виде царапин.

При анализе исцарапанных образцов программное обеспечение SIAMS также использует принцип «умной дилатации» для исключения царапин и сохранения контура частиц.

В источнике [2] описан способ поиска царапин на изображении при анализе неметаллических включений в стали. Поиск царапин базируется на анализе собственных значений Гессиан функции интенсивности. Обычно неметаллические включения расположены на изображении достаточно редко, имеют малую толщину и протяженность, что позволяет эффективно применять описанный способ поиска царапин без дополнительной обработки.

Рассмотрим случай, когда изображение содержит большое количество объектов контроля наряду с царапинами. В этом случае только обнаружения царапин недостаточно, необходимо восстановление контура частиц в местах пересечения их царапинами.

На рис. 5 приведен пример микроструктуры пластинчатого графита в чугуна. Как и в предыдущем примере, изображение априори считается не пригодным для цифровой обработки из-за наличия дефектов в виде грубых и множественных царапин и неравномерного освещения.

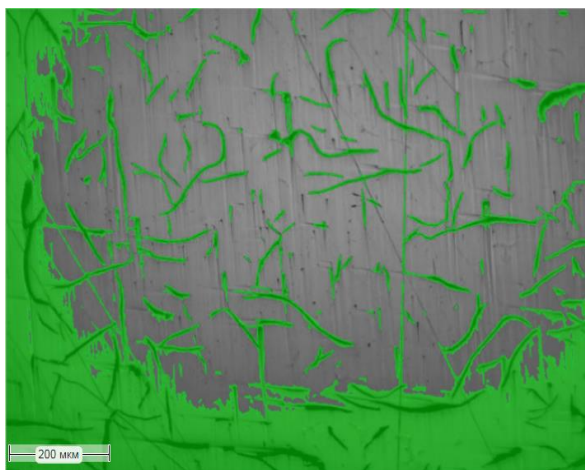


Рис. 5. Исходное изображение

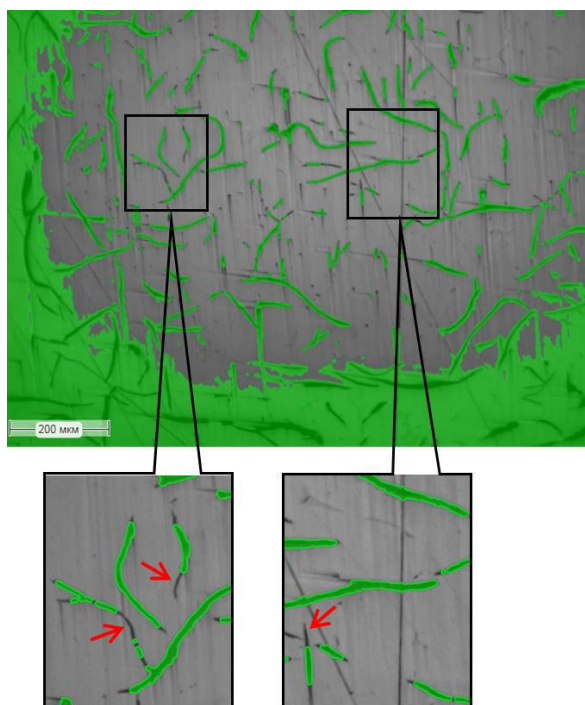
На рис. 6 показан результат выделения частиц графита после применения стандартного приема обработки изображения - пороговой сегментации (выделение частиц графита по уровню серого). Вместе с частицами графита выделяются тени и царапины.

На рис. 7 показан результат применения последующего морфологического открытия для исключения царапин. Очевидно, что простые приемы обработки изображений в данном случае не подходят. Частицы в центре изображения, которые удается отделить от фона, приобретают искаженную форму и умень-

шаются в размерах. Вырезание центра изображения в два раза уменьшит площадь анализа.



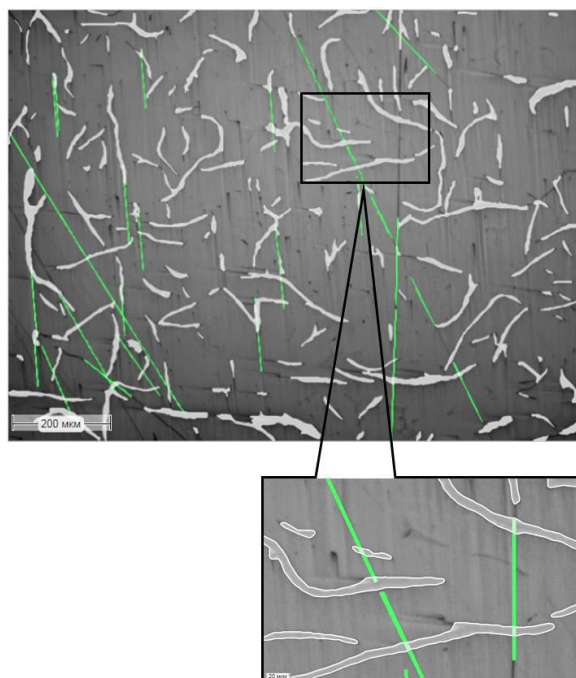
**Рис. 6.** Результат применения пороговой сегментации. Вместе с частицами графита выделяются тени и царапины



**Рис. 7.** Результат морфологического открытия. Исключение царапин из объектов контроля с искажением контура частиц

Для получения приемлемого результата цифровой обработки необходимо использование более сложного алгоритма, включающего вычисление локального порога сегментации [2], поиск царапин и «умную» дилатацию.

Результат применения комплекса функций SIAMS для обработки дефектных изображений в программном обеспечении SIAMS приведен на рис. 8.



**Рис. 8.** Результат применения комплекса функций SIAMS для анализа дефектных изображений. Раздельное распознавание царапин и частиц с сохранением контура частиц

Частицы и царапины распознаются отдельно. При этом найденные объекты «царапины», пересекая частицы, не нарушают их исходный контур.

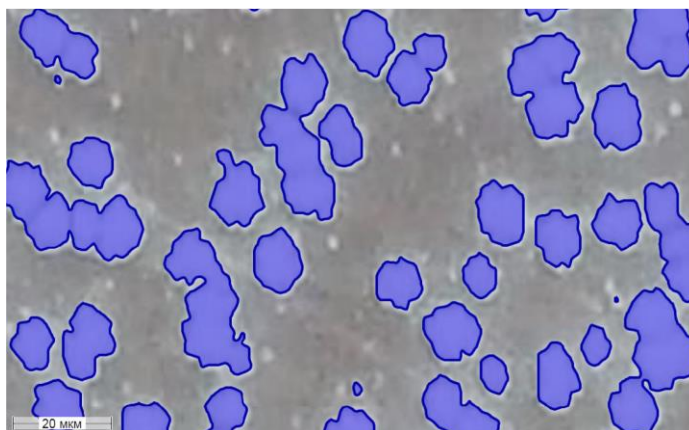
## 2.2 Разделение слипшихся частиц на изображении

Рассмотрим пример анализа гранулометрического состава микроструктуры порошка (рис. 9). Обработка изображения проведена с применением пороговой сегментации (по уровню серого) (рис. 10) и последующего морфологического открытия (рис. 11). Операция открытия позволила избавиться от мелкого «мусора» и сгладить контуры частиц, но справилась только с разделением одной частицы на приведенном фрагменте изображения (рис. 11). Очевидно, что порошок более дисперсный, чем полученный контур маски частиц. Для проведения количественного гранулометрического анализа необходимо разделение частиц другими способами.

Для эффективного разделения соприкасающихся объектов на изображении программное обеспечение SIAMS использует комплекс одновременно применяемых и взаимосвязанных функций.



**Рис. 9.** Исходное изображение



**Рис. 10.** Результат применения пороговой сегментации

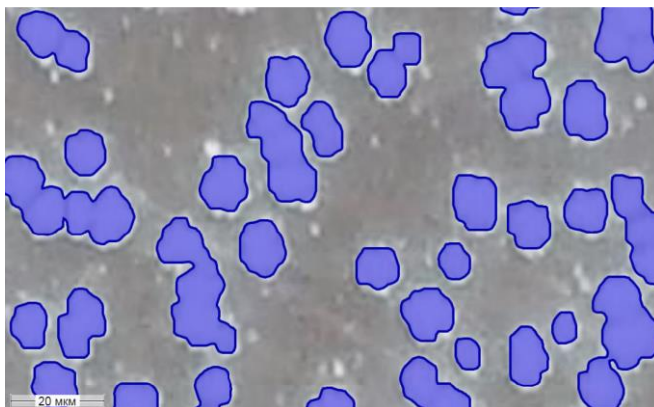


Рис. 11. Результат применения пороговой сегментации и открытия

#### Разделение с помощью анализа скелета объекта.

Алгоритм анализирует скелетные линии связанных компонент бинарной маски объектов и пиксельной карты расстояний до границ объектов [3]. Скелетные линии образуют граф связей узлов скелета Zhang-Suen [4], по которому производится поиск выраженного минимума. Из точки локализации минимума алгоритм находит две ближайшие точки границы объекта по обе стороны от ребра скелета и проводит линии разрезов в эти точки.

#### Разделение алгоритмом водораздела.

Алгоритм использует маркеры на пиках карты высот.

$D_{ij}$  – карта высот,  $d_m = \max(D_{ij})$ , где:

$d_{ij} = \begin{cases} 0, & D_{ij} < d_m \\ 1, & D_{ij} \geq d_m \end{cases}$  – множество точек, соответствующих максимальному значению карты высот;

$d_{ij}^n = \begin{cases} 0, & D_{ij} < n \\ 1, & D_{ij} \geq n \end{cases}$ ,  $n = d_m \cdot 1$  – множество точек, превышающее порог  $n$ .

Картой маркеров  $d^*$  для алгоритма водораздела является объединение множеств  $d^n$ , в котором отсутствует самопересечение (2):

$$d^* = \bigcup_{n=d_m-1}^1 (d/d^n \cup d). \quad (2)$$

Возникновение нескольких маркеров внутри одной связной компоненты разделяет ее на фрагменты [5].



### Разделение с помощью алгоритма триангуляции.

Алгоритм разделения частиц основан на принципах триангуляции Делоне с ограничениями, оценивающей толщину частицы по всему ее контуру. Разделение происходит по ребру треугольника, имеющего значение менее заданного порогового значения [6].

Большой опыт работы по разделению частиц на различных изображениях микроструктуры порошков показывает, что применение только одного из описанных приемов не позволяет эффективно разделить частицы. На рис. 12 показан результат разделения частиц с использованием триангуляции Делоне.

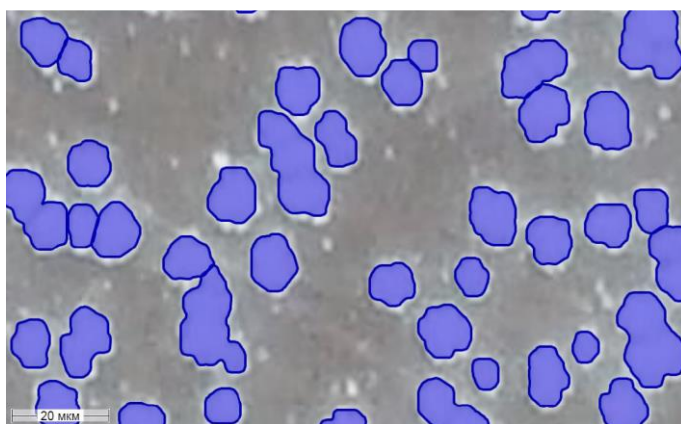


Рис. 12. Результат разделения частиц с применением триангуляции Делоне

И только использование сложного алгоритма, включающего сразу несколько способов разделения частиц, приводит к результату, близкому к ожидаемому (рис. 13).

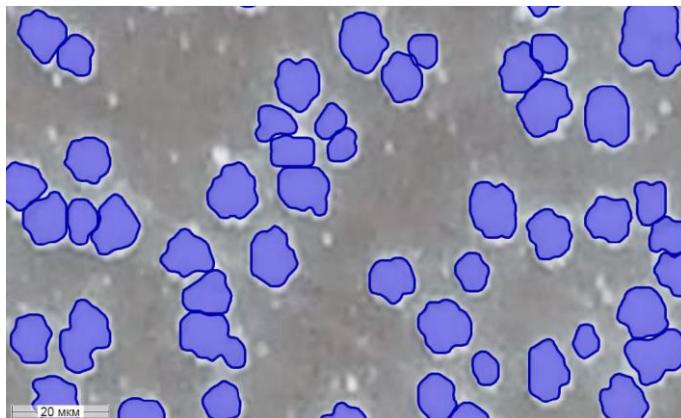


Рис. 13. Результат разделения частиц с применением комплекса функций SIAMS

### 3 Заключение

В данной работе рассмотрены примеры цифровой обработки изображений микроструктуры, предназначенной для проведения количественного анализа. В качестве примеров подобраны изображения с часто встречающимися дефектами пробоподготовки: царапины, пятна от реактива и воды, недостаточное диспергирование частиц порошка. Описаны алгоритмы SIAMS, учитывающие наличие на анализируемых изображениях перечисленных дефектов.

Авторы уверены, что разработка алгоритмов, учитывающих недостаточное качество пробоподготовки, имеет практическую значимость, так как лабораториям серийного контроля микроструктуры материалов часто приходится работать в условиях недостатка времени, навыков или специального оборудования.

Необходимо заметить, что, несмотря на положительные результаты борьбы с дефектами на изображениях микроструктуры, не следует пренебрегать требованием к качеству пробоподготовки. Количественный анализ изначально бездефектного изображения позволит получить более высокую точность результатов, чем анализ изображения с дефектами при использовании самой современной математической обработки.

### 4 Благодарности

Компания ООО «СИАМС» выражает благодарность нашим заказчикам и партнерам, которые передавая нам свои образцы или изображения микроструктуры, полученные в обычном режиме работы лаборатории, ставят перед нами новые задачи количественного анализа, тем самым побуждая нас искать новые пути их решения и развивать возможности наших программных продуктов.

### Литература

1. Гонсалес Р., Вудс.: Цифровая обработка изображений, 1104 с, (2012)
2. Сивкова Т., Губарев С.: Автоматизированный анализ микроструктуры материалов. Нестандартный подход к анализу изображений. GraphiCon 2019: труды 29-й Международной конференции по компьютерной графике и машинному зрению (Брянск, 23–26 сент. 2019 г.)/Брянский государственный технический университет. – Брянск (2019) – 286 с.
3. Borgefors, Gunilla: Distance transformations in digital images. *Comput. Vision Graph. Image Process.* 34 3, pp 344–371 (1986).
4. Zhang, T. Y., Suen, C. Y.: A Fast Parallel Algorithm for Thinning Digital Patterns. *Communications of ACM.* 27(3), 236–239 (1984)
5. Fernand Meyer: Color image segmentation. In *Image Processing and its Applications, 1992., International Conference on*, pages 303–306. IET (1992)
6. Скворцов А.В.: Триангуляция Делоне и её применение. – Томск: Изд-во Том.ун-та, (2002) 128 с.



**НАУЧНАЯ ВИЗУАЛИЗАЦИЯ И  
ВИЗУАЛЬНАЯ АНАЛИТИКА**

---

**SCIENTIFIC VISUALIZATION AND  
VISUAL ANALYTICS**

**GraphiCon 2020**

---

22-25 сентября 2020  
Санкт-Петербург, Россия

## Визуализация многомерных поверхностей и их экстремумов \*

А.А. Батюков<sup>1</sup>, Е.В. Попов<sup>1</sup>, Н. Фогт<sup>2,3</sup>

<sup>1</sup> Нижегородский государственный архитектурно-строительный университет, 603950, Нижний Новгород, Россия

<sup>2</sup> Московский государственный университет им. М.В. Ломоносова, 119234, Москва, Россия

<sup>3</sup> University of Ulm, Albert Einstein Allee 47, D-89069 Ulm, Germany

**Аннотация.** В статье описан метод многомерной визуализации гиперповерхности и ее экстремумов по модели Люмиграф. В статье показаны недостатки классического Люмиграфа и сформулированы причины этого. Чтобы преодолеть эти недостатки, авторы вводят концепцию дополнительного экрана. Описаны правила построения изображения на этом экране. Окончательное изображение делает видимой всю гиперповерхность и зоны ее экстремумов. Для возможности визуализации гиперповерхностей размерностью порядка более, чем 5D как развитие модели Люмиграф, разработана модель Exidiagraph и описан способ её построения. Введено понятие псевдо-воксела для отображения гиперповерхности, представленной многомерной таблицей. Разработан программный пакет для визуализации таблично заданных многомерных поверхностей.

**Ключевые слова:** многомерная визуализация, Люмиграф, Exidiagraph.

### 1 Введение

Основной особенностью информации является ее многомерность. В связи с этим информация часто теряет свою четкость и возможность представить данные в визуальной форме с помощью стандартных средств визуализации, таких как чертежи, графики, диаграммы и т.д. [1-3]. Эта проблема часто возникает в различных областях человеческой деятельности. Например, в работе [4] сделана попытка систематизировать и кратко описать некоторые типы проблем визуализации, возникающих при обработке результатов обобщенного математического эксперимента в вычислительной газовой динамике. Автор также организовал новые задачи визуального представления многомерных числовых выходных данных. Разработка методик анализа результатов обобщенного вычислительного эксперимента, представленного в виде многомерного объема данных, рассмотрена в [5].

---

\* Работа выполнена в рамках гранта РФФИ № 19-07-01024 А.

Физико-химическое изучение и прогнозирование свойств химических соединений имеют первостепенное значение как с технологической, так и с академической точек зрения. Квантово-химические вычислительные методы становятся полезными инструментами для решения некоторых химических задач, в том числе для изучения механизма реакции перед сложными, дорогостоящими или иногда невозможными экспериментами. Исследование структуры молекул нередко сводится к нахождению множества экстремумов на поверхности потенциальной энергии (PES), то есть к изучению так называемого «конформационного ландшафта» [6], [7]. В общем случае PES имеет многомерный характер. Строго говоря, решение задачи визуализации многомерной поверхности с ее экстремумами весьма актуально для многих прикладных задач в разных отраслях науки и техники. В этой статье рассмотрены общие принципы организации визуализации для многомерной поверхности в зависимости от четырех до шести независимых переменных. Будем называть такую поверхность 5D-7D поверхностью (или гиперповерхностью).

Проблема многомерной визуализации гиперповерхностей очень актуальна и сложна. Существует множество подходов к визуализации и большое количество визуальных таксономий [8]. В качестве подходов к многомерной визуализации можно выделить следующие: 2D и 3D диаграммы рассеяния; Матрица диаграмм рассеяния; Тепловые карты; Карты высот; Линзовые таблицы; Обзорные участки; Иконографические представления; Общие логические диаграммы; Параллельные координаты; Линейный график, Многострочный график; Пиксельные техники; Круговые сегменты; Многомерное масштабирование и графики Саммона; Полярные карты; Анализ главных компонент и главной кривой; Погоня за проекцией; Самоорганизующиеся карты Кохонена и многие другие. Однако анализ в работе [6] показал, что ни один из них не является достаточно подходящим для одновременной визуализации гиперповерхности и ее экстремумов.

## 2 Визуализация 5D поверхности с помощью Люмиграфа

Существует возможность представить пятимерную гиперповерхность с помощью функции четырех независимых переменных с уровнем энергии, представленным цветом, и применить модель под названием Люмиграф. Обычное параметризованное двухплоскостное световое поле (2PP), расширенное геометрией объекта для процесса реконструкции, было описано Гёртлером [9], который ввел термин Люмиграф. Это обычно применяется, чтобы подчеркнуть тот факт, что прокси-геометрия требуется и используется для реконструкции светового поля. Это подмножество полной пленоптической функции, которая описывает поток света во всех положениях и во всех направлениях. Двухплоскостная установка — это глобальная параметризация, которая описывает луч в пространстве с точками пересечения в двух параллельных плоскостях. Каждая точка пересечения является 2D, и это приводит к четырем параметрам выборки. В нашем случае этот вид луча можно рассматривать как четырехмерную точку. Цвет

точки, рассматриваемый как дополнительный параметр, преобразует эту точку в 5D точку.

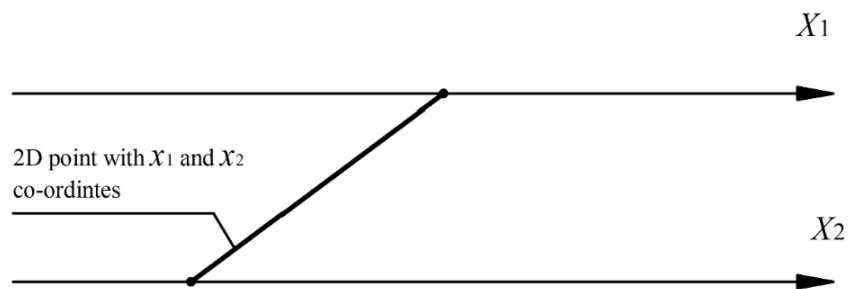


Рис. 1. Представление 2D точки с помощью Люмиграфа

Основная идея может быть объяснена на примере 2D аналога Люмиграфа (см. рис.1). Пусть есть две параллельные оси  $X_1$  и  $X_2$ , связанные с двумя независимыми переменными. Тогда прямая между двумя точками на этих осях адекватна 2D точке с координатами  $(x_1, x_2)$  в декартовом пространстве.

Если расширить эту идею до случая четырех независимых переменных, точка может быть представлена линией, показанной на Рис. 2, где две ее конечные точки имеют две пары переменных  $(D_3, D_1)$  и  $(D_6, D_5)$ , связанных с соответствующей плоскостью.

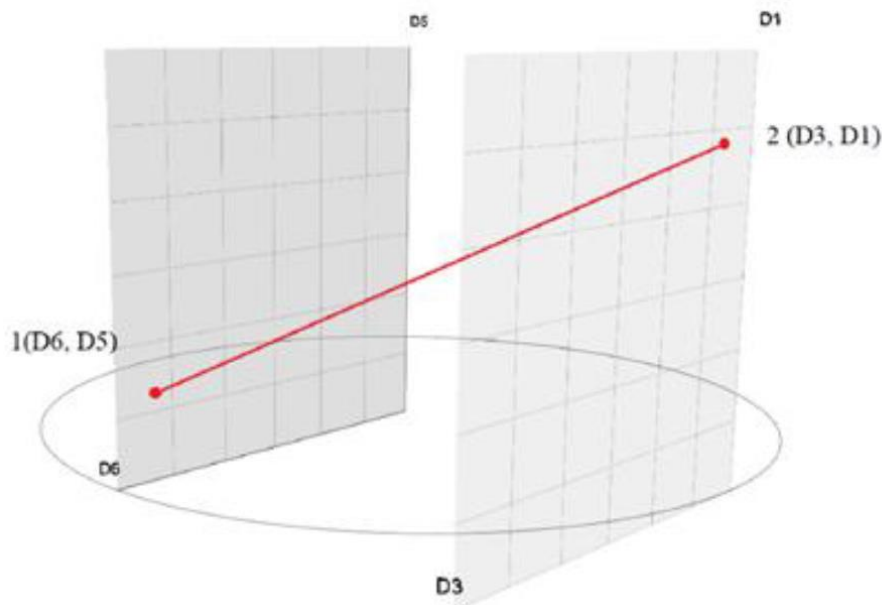


Рис. 2. Точка в зависимости от четырех переменных

В геометрии любая поверхность рассматривается как бесконечное множество точек. Однако в большинстве приложений функция многомерной поверхности представлена в виде конечных таблиц. Это означает, если такие таблицы являются четырехмерными, то их визуализация может быть осуществлена как конечный набор только четырехмерных точек. Люмиграф — это удобная и простая модель для визуализации функции четырех переменных, как показано на рис. 3, где значения функции отображаются в цветном виде. Обычно исходная таблица функций 4D имеет  $n$  строк и столбцов в каждом измерении, поэтому количество точек 4D в Люмиграфе равно  $n^4$ . Хотя такое представление очень просто, наглядно и удобно, оно недостаточно ясно для восприятия. Области экстремумов функции при этом практически не видны.

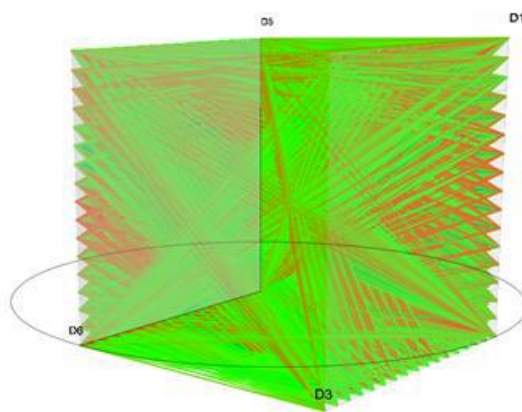


Рис. 3. Люмиграф табличной функции четырех переменных

С целью улучшения визуализации 5D поверхности использован прием отображения Люмиграфа на плоскость (см. работу [10]). Основное требование к такому отображению заключается в том, чтобы вся поверхность и ее экстремумы были видны пользователю. Основная идея здесь — разместить третью плоскость, параллельную двум начальным внутри Люмиграфа для нахождения точки пересечения с ней отрезков. Этот подход назван концепцией дополнительного экрана. Такой способ отображения позволяет осуществить полное 2D-отображение всей 5D-гиперповерхности в виде растрового изображения (см. рис. 4, где условно скрыты 4D-точки).

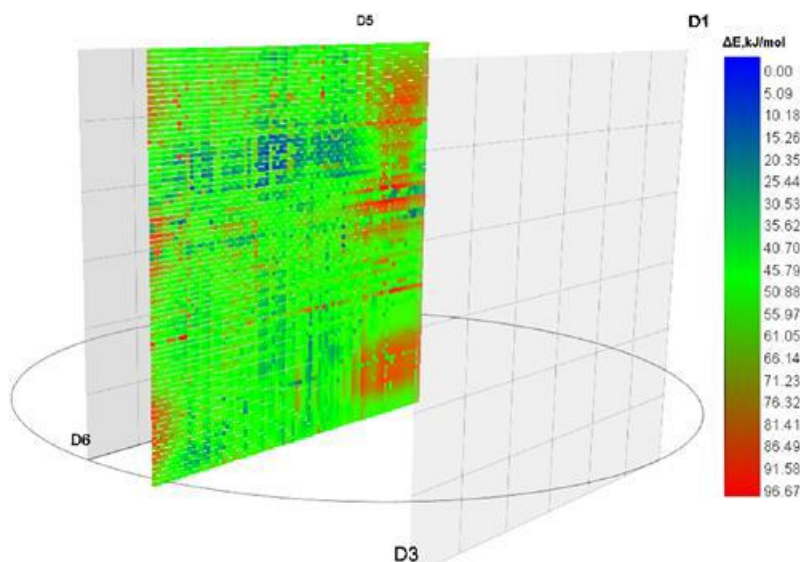


Рис. 4. Отображение табличной функции четырех переменных с помощью растрового изображения

### 3 Визуализация 6D и 7D поверхностей

Отображение точки в пространстве размерности 6 и выше осуществимо с использованием идеи, реализованной в Люмиграф модели. Как было сказано выше для визуализации точки в пространстве 4D были использованы две параллельные плоскости, в каждой из которых были заданы две двумерные системы координат. В развитие этой идеи была разработана новая модель визуализации точки в пространстве 6D. Реконструкция Люмиграф модели заключалась в добавлении к ней еще одной плоскости с двумерной системой координат, дополняющей модель до пространства размерности 6D. Таким образом была получена система координат, состоящая из трех координатных плоскостей, расположенных так, чтобы все плоскости образовали трехгранник вокруг оси Z (Рис.5). Данная модель получила название Exidiagraph (греч. *éxi diastáseon gráfima* - шестимерный график).

Для отображения в Exidiagraph точек гиперповерхности набор шести координат разбивается на пары. Для случая шести переменных можно получить три пары координат по одной паре для каждой из трех плоскостей. Последовательность выбора координат для разбиения по парам может быть любой. Аналогичный подход можно использовать для визуализации пространств большей размерности, например 8D, 10D и т.д., разбив координаты по парам и добавив координатную плоскость для каждой дополнительной пары координат. С каждой координатной плоскостью можно ассоциировать точку, соответствующую од-



ной пары координат  $(x, y)$ . Полученные на координатных плоскостях точки можно считать вершинами треугольника, изображенного на рис.5.

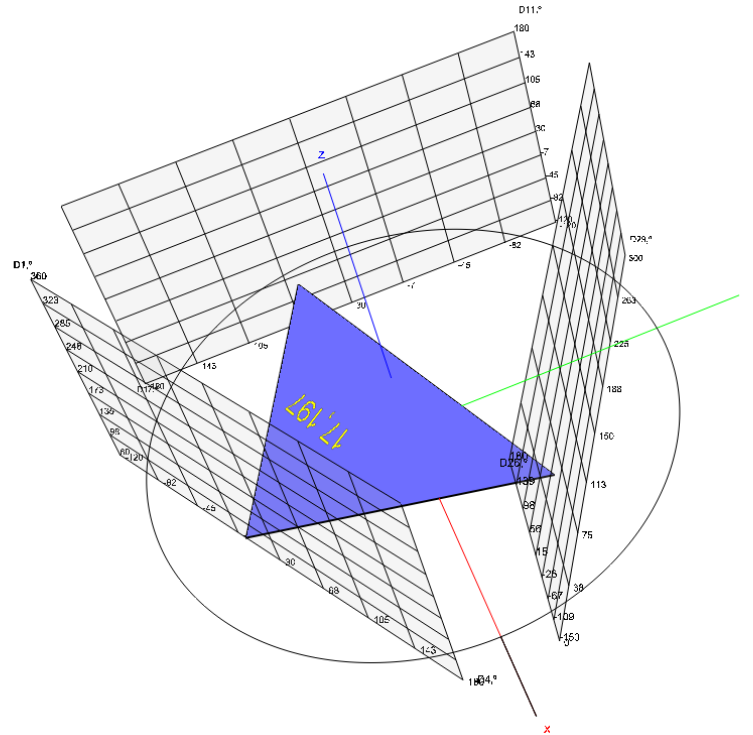


Рис. 5. Отображение точки в виде треугольника в 6D пространстве

Таким образом полученный треугольник однозначным образом соответствует точке, определяемой шестью независимыми переменными. Если данному треугольнику поставить в соответствие определенное значение энергии и отобразить его в виде цвета, ассоциированного с цветовой легендой, то можно получить отображение 7D точки, зависящей от шести независимых переменных в 6D пространстве. Однако, поскольку в геометрическом смысле поверхность любой размерности может рассматриваться как бесконечное множество точек, визуализация всей гиперповерхности с использованием описанной модели представляет собой значительные трудности.

Промежуточный путь преодоление данной трудности может быть найден следующим образом. Точка в 6D пространстве может быть ассоциирована с геометрическим центром треугольника, а соответствие этой точки конкретным координатам может быть визуализировано отрезками прямых линий, связывающих эту точку с точками на соответствующих координатных плоскостях. Эти отрезки назовем линиями связи.

С учетом положительного эффекта от использования вышеописанного подхода к визуализации многомерных точек была разработана псевдо-воксельная

модель визуализации гиперповерхности. В основе модели лежит понятие псевдо-воксел, которое в отличие от термина воксел формируется не в пространстве 3D, а в пространстве 6D. Визуализация строится при этом с учетом того, что гиперповерхность представлена в виде многомерной таблицы. С целью визуализации каждой ячейке многомерной таблицы ставится в соответствие многомерная точка. Затем каждая многомерная точка визуализируется по схеме, описанной в данном разделе выше. В результате визуализации в модели Exidiagraph можно получить картину, представленную на рис.6, где области экстремумов хорошо видны (синим цветом отображены зоны минимумов, красным – зоны максимумов).

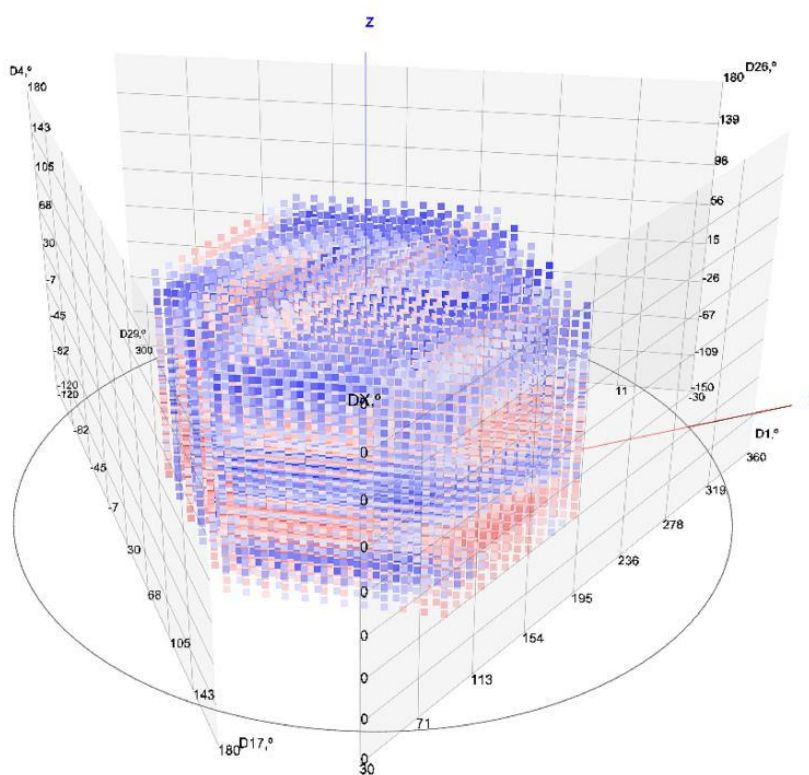


Рис. 6. Псевдо-воксельная модель 7D поверхности в Exidiagraph

Если теперь уровни энергии, соответствующие каждой точке гиперповерхности, нормировать по их истинному значению, то можно получить уточненную псевдо-воксельную визуализацию гиперповерхности, которая является наиболее адекватной визуализацией табличной многомерной гиперповерхности со всеми ее экстремумами (рис. 7).

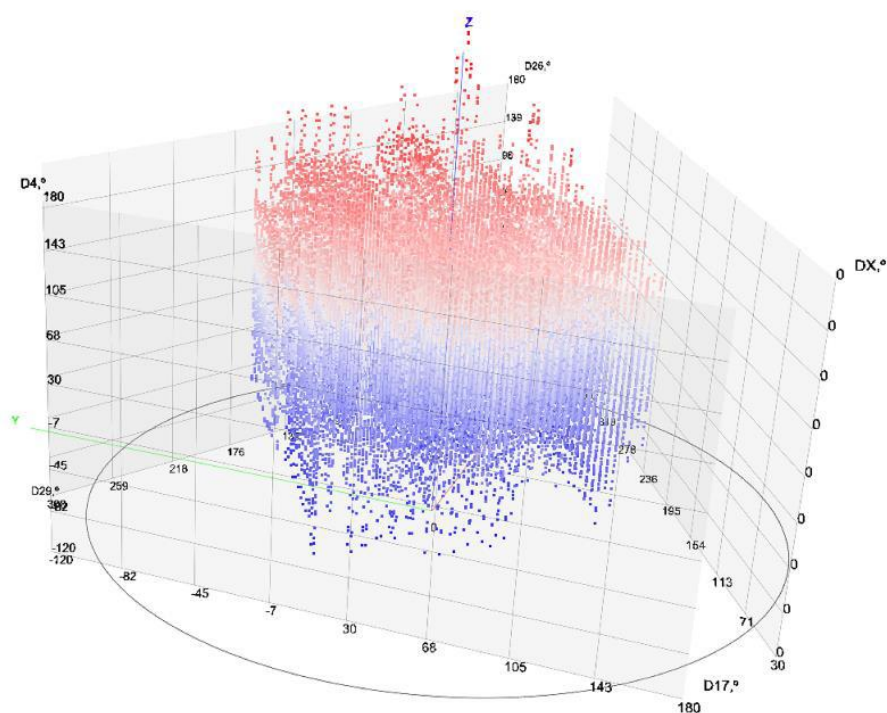


Рис. 7. Уточненная псевдо-воксельная модель 7D поверхности в Exidiagraph

#### 4 Нахождение минимумов PES в табличном виде

С математической точки зрения проблема нахождения минимумов функций связана с нахождением таких ее точек, в которых первые частные производные функций стремятся к нулю, а вторые частные производные одновременно неотрицательны. Эта проблема требует нахождения такой точки  $x$ , что скалярная функция  $f(x_1, x_2, x_3 \dots x_n)$  принимает значение, которое ниже, чем в любой соседней точке. Для гладких функций градиент  $g = \nabla f$  обращается в нуль в экстремуме. Эта проблема также известна в математике как проблема оптимизации, которая является известной областью науки, техники и технологии. При решении так называемой задачи оптимизации необходимо вычислить глобальный экстремум (или его хорошее приближение) функции с несколькими переменными. Переменные, которые определяют оптимизированную функцию, могут быть непрерывными или дискретными, и, кроме того, они часто должны удовлетворять определенным ограничениям. Задачи класса сложности NP-hard, включающие задачи оптимизации, чрезвычайно сложно поддаются решению. Поэтому традиционные алгоритмы оптимизации спуска не подходят для их решения из-за локальной природы обрабатываемой информации. В последние

десятилетия появилось много новых алгоритмов для решения задач глобальной оптимизации. Это привело к успеху в решении широкого круга задач в различных областях, таких как, например, вычислительная химия и биология, информатика, экономика, инженерное проектирование и другие. Но в целом нет эффективных методов, позволяющих минимизировать  $n$ -мерные функции. Алгоритмы исходят из исходного предположения с использованием алгоритма поиска, который пытается двигаться в наклонном направлении. Алгоритмы, использующие функцию градиента, минимизируют одномерную линию вдоль этого направления, пока не будет найдена самая низкая точка с подходящим допуском. Затем направление поиска обновляется в соответствии с локальной информацией о функции и ее производных, и весь процесс повторяется до тех пор, пока не будет найден истинный  $n$ -мерный минимум.

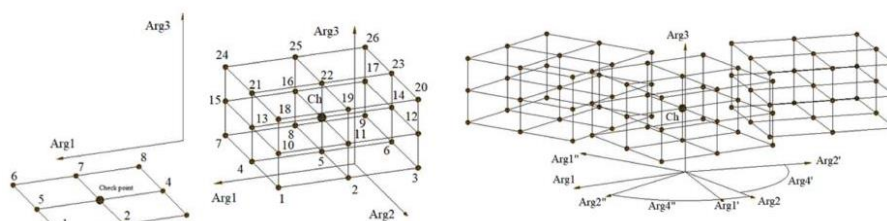
Алгоритмы, которые не требуют градиента функции, используют другой подход. Алгоритм Симплекса Нелдера-Мида поддерживает  $n + 1$  векторов тестовых параметров в качестве вершин  $n$ -мерного симплекса. Алгоритм пытается исправить вершину симплекса, имеющую наихудшие параметры, на каждой итерации, используя геометрические преобразования. Итерации продолжаются до тех пор, пока общий размер симплекса не будет достаточно уменьшен. Проблема представляется очень сложной для негладких или дискретных функций. На самом деле, пакеты, например, вычислительной химии производят значения молекулярной энергии в виде большой многомерной таблицы в зависимости от 2, 3, 4 или более переменных, которые соответствуют ряду переменных молекулы. В этом случае табличная гиперповерхность не имеет аналитического выражения, что серьезно затрудняет поиск ее экстремумов. Кроме того, время для расчета энергии критически зависит от ряда переменных, а также от шага расчета для каждой данной переменной. С большим числом переменных и более сжатым шагом время расчета может занять до нескольких месяцев непрерывных вычислений на суперкомпьютерных платформах. Таким образом, это усложняет получение экстремума гиперповерхности с приемлемой надежностью.

Для решения этой проблемы был разработан подход для нахождения локальных минимумов гиперповерхности. Он состоит из двух этапов:

1. Чтобы улучшить поиск минимумов, необходимо заменить исходную многомерную функцию, определенную как регулярная сетка вычисленных точек, на более сжатую регулярную сетку. Значения функций в узлах сжатой сетки рассчитываются путем многомерной сплайн-интерполяции. Существует несколько подходов к многомерной интерполяции. В описанной версии гиперповерхностного процессора авторы применяют процедуру интерполяции многомерного сплайна, разработанную M. Flanagan [11], которая позволяет интерполировать значения функции, заданной таблицей любой размерности. Эта процедура содержит метод для выполнения многомерной кубической сплайн-интерполяции, то есть интерполяции в  $n$ -мерных массивах точек данных,  $y = f(x_1, x_2, x_3 \dots x_n)$ , используя естественные кубические сплайны и где  $n$  может принимать любые целочисленные значения. Данные представле-

ны в виде таблицы для массивов  $x_1, x_2, x_3 \dots x_n$  и для  $y = f(x_1, x_2, x_3 \dots x_n)$ . Размерность массивов  $x_i$  и этапов и  $x$  разбиения могут быть разными. Процедура интерполяции является рекурсивной.

- Следующий этап – последовательное сравнение значений функций в каждом узле сетки со значениями функций в соседних узлах, окружающих его. Если это значение является наименьшим или наибольшим, то это значение можно считать локальным экстремумом. Количество соседних узлов зависит от размерности решаемой задачи. В трехмерном случае таких узлов 8, в четырехмерном случае 26 узлов, в пятимерном случае таких узлов 80 (рис.8). Чтобы сделать экстремумы более точными, сетка должна быть максимально сжатой. Этот подход позволяет находить экстремумы энергетической функции на достаточном уровне надежности, даже если расчет является относительно грубым. В отличие от классического решения по оптимизации, этот подход позволяет очень быстро находить как глобальный экстремум, так и локальные экстремумы. Тем не менее, следует отметить, что конденсация сетки фактически не повышает точность исходного решения. Это может привести к появлению некоторых фиктивных минимумов, которые на самом деле не существуют, в то же время некоторые реальные минимумы могут быть потеряны. Следовательно, весь минимум должен быть подтвержден соответствующим экспериментом.



**Рис. 8.** Схема нахождения минимумов. а) 3D; б) 3D + переменная для цвета = 4D; в) 4D + переменная для цвета = 5D

## 5 Заключение

В статье описан метод визуализации многомерной гиперповерхности с ее экстремумами на основе модели Люмиграф. Классический Люмиграф недостаточно удобен для восприятия, поскольку он может быть представлен только набором прямых отрезков, как описано в этой статье. Зоны экстремумов функции практически не видны, даже если Люмиграф окрашен в разные цвета. Дополнительный экран, параллельный плоскостям Люмиграфа, содержит растровое изображение всех зон гиперповерхности. Кроме того, такое изображение окрашено в соответствии с его цветовой легендой. Этот механизм отображения многомерной поверхности назван концепцией дополнительного экрана. Благодаря такому механизму мы можем получить более визуально эффективное растровое

изображение функции вместе со всеми ее экстремумами. Этот подход применим для визуализации 5-мерной поверхности, зависящей от четырех переменных. Дальнейшая оценка этого подхода направлена на увеличение размерности отображаемой гиперповерхности. Как развитие идеи, заложенной в модели Люмиграф, была получена система координат, состоящая из трех координатных плоскостей, расположенных так, чтобы все плоскости образовали трехгранник вокруг оси Z. Данная модель получила название Exidiagraph. На базе этой модели разработан программный пакет Exidiagraph для визуализации многомерных поверхностей вместе с их экстремумами на основе языка Java [12], библиотеки Java 3D и библиотеки многомерной сплайн интерполяции Flanagan M. [11]. Программный пакет состоит из трех основных модулей: модуля считывания численных таблиц с представлением их в виде многомерной матрицы, модуля интерполяции матричных значений, модуля отображения многомерной поверхности. Программное обеспечение позволяет визуализировать многомерные поверхности размером до 7D.

## Литература

1. Wehrend S. and Lewis C. "A Problem-Oriented Classification of Visualization Techniques", *Proceedings of the 1st IEEE Conference on Visualization '90*, 1990, pp.139-143.
2. Etemadpour R. et al. Choosing Visualization Techniques for Multidimensional Data Projection Tasks: A Guideline with Examples. (2016, 166-186 pp 598. 166-186. 10.1007/978-3-319-29971-6\_9.
3. Keim D. A. and Kriegel H.-P. "Visualization Techniques for Mining Large Databases: A Comparison", *IEEE Transactions on Knowledge and Data Engineering*, vol.8, no.6, 1996 pp.923-938.
4. Bondarev A.E. On visualization problems in a generalized computational experiment / *Scientific Visualization*, 2019, volume 11, number 2, pages 156-162, <http://doi.org/10.26583/sv.11.2.12>.
5. Bondarev A.E., Galaktionov V.A., Shapiro L.Z. Processing and visual analysis of multidimensional data / *Scientific Visualization*. V.9, № 5, с.86-104, 2017, DOI: <http://doi.org/10.26583/sv.9.5.08>
6. E. Popov, A. Batiukov, N. Vogt, T. Popova, J. Vogt. Visualization and Analysis of Molecular Potential Energy Surface (Pes) and Its Minima. *IADIS International Conference Interfaces and Human Computer Interaction 2019* (part of MCCSIS 2019), Porto, 2019. pp. 411-415.
7. Popov, E. V., Batiukov, A. A., Vogt, N., Popova, T. P., & Vogt, J. (2020). Visualization and Minima Finding of Multidimensional Hypersurface. In P. Isaias, & K. Blashki (Eds.), *Interactivity and the Future of the Human-Computer Interface* (pp. 282-309). Hershey, PA: IGI Global. doi:10.4018/978-1-7998-2637-8.ch015
8. B. Shneiderman, "The Eyes Have It: A Task by Data Type Taxonomy of Information Visualization," presented at IEEE Symposium on Visual Languages '96, Boulder, CO, 1996.
9. Götler S. et al, The Lumigraph. In *Computer Graphics, Annual Conference Series* (Proc. SIGGRAPH '96): 1996. pp. 43-54.
10. Eugene Popov, Tatyana Popova, Anatoly Batiukov, Natalja Vogt. Visualization of Multidimensional Hypersurface Extrema by Lumigraph (2020). *Scientific Visualization* 12.2: 1 - 8, DOI: 10.26583/sv.12.2.01.

11. Flanagan, M. T. (2014). *JavaScientific Library*  
<http://www.ee.ucl.ac.uk/~mflanaga/java/index.html>.
12. Popov E.V., Batiukov A.A. Exi Diastaseon Grafima (ExiDiaGraph). Certificate of a computer program RU 2020612284, 19/02/2020.Application No. 2020611368/69 of 10/02/2020.

## Визуализация двух противоположно вращающихся сверхзвуковых вихрей<sup>†</sup>

В. Борисов<sup>1</sup>, А. Давыдов<sup>1</sup>, Т. Константиновская<sup>1</sup>, А. Луцкий<sup>1</sup>

<sup>1</sup> Институт Прикладной Математики им.М.В. Келдыша РАН, 125047, Москва, Россия  
allutsky@yandex.ru  
konstantinovskaya.t.v@gmail.com

**Аннотация.** В работе выполнена визуализация пары сверхзвуковых противоположно вращающихся вихрей с использованием методов максимальной завихренности и  $\lambda_2$ -критерия. Численные данные были получены на суперкомпьютерной системе К-60 в ИПМ им. М.В. Келдыша РАН с использованием параллельного алгоритма для моделирования турбулентных течений.

**Ключевые слова:** идентификация вихря, концевой вихрь, противоположно вращающиеся вихри.

### 1 Введение

Полет любого летательного аппарата (ЛА) сопровождается образованием вихревых структур за ним, в частности, концевых вихрей на кромках крыльев и прочих деталей.

Изучение сверхзвуковых концевых вихрей, а также их взаимодействия представляется актуальным в виду возросшего интереса к сверхзвуковым ЛА как в военной промышленности, так и в гражданской отрасли. Однако наряду с вопросом получения корректных данных о течении, существует вопрос постобработки и визуализации полученных данных, зачастую оказывающихся весьма массивными, особенно в трехмерном случае. Для этого используются, в частности, специальные методы визуализации вихревых течений. Обширные обзоры были сделаны по данной тематике различными авторами, в том числе [1–3].

Авторами представлена визуализация результатов численного исследования пары противоположно вращающихся сверхзвуковых концевых вихрей с помощью метода максимальной завихренности и  $\lambda_2$ -критерия визуализации вихревых структур.

---

<sup>†</sup> Работа выполнена при поддержке гранта РФФИ № 19-01-00765.



## 2 Конфигурация модели

Методы визуализации применены к данным, полученным при моделировании пары сверхзвуковых противоположно вращающихся вихрей при числе Маха набегающего потока  $M = 3$ .

Численно исследовалось течение за двумя соосными крыльями с острыми передней, задней и боковой кромками и ромбовидным основанием (рис.1). Крылья располагались под углом  $10^\circ$  к набегающему потоку и крепились основанием к стенкам, параллельным его направлению. Расчеты проводились в безразмерных величинах [4], за единицу длины бралось  $L = 1$  м. Хорда каждого крыла была равна  $b = 0.03$ , полуразмах первого крыла  $l_1 = 0.075$ , второго -  $l_2 = 0.095$ . Толщина ромбовидного основания обоих крыльев равнялась  $h = 0.004$ . Расстояние между концевыми хордами крыльев составило  $l_3 = 0.03$ , таким образом, ширина области между стенками равнялась  $H = 0.2$ . Ось  $x$  была сонаправлена набегающему потоку. Ось  $z$  совпадала с общей осью крыльев. Ось  $y$  была направлена с подветренной стороны крыльев в наветренную сторону. Длина рассматриваемой области составляла 10 хорд крыла вниз по потоку от общей оси крыльев. Число Рейнольдса в расчетах задавалось равным  $Re_L = 1 \times 10^7$ .

## 3 Численный метод

Численное моделирование проводилось в ИПМ им. М.В. Келдыша РАН на суперкомпьютере К-60 с помощью разработанного авторами программного комплекса ARES [5] для расчета трехмерных турбулентных течений вязкого сжимаемого газа на высокопроизводительных вычислительных системах. Численный метод основан на решении нестационарных осредненных по Рейнольдсу уравнений Навье-Стокса (URANS) с моделью турбулентности Спаларта-Аллмараса [6,7]. Для дискретизации уравнений по пространству использовался метод конечных объемов с реконструкциями 2-го (TVD) и 3-го (WENO) порядков. Временная аппроксимация была выполнена на основе явной схемы. Расчеты проводились на неструктурированной гексагональной сетке, состоящей из 25 774 200 ячеек.

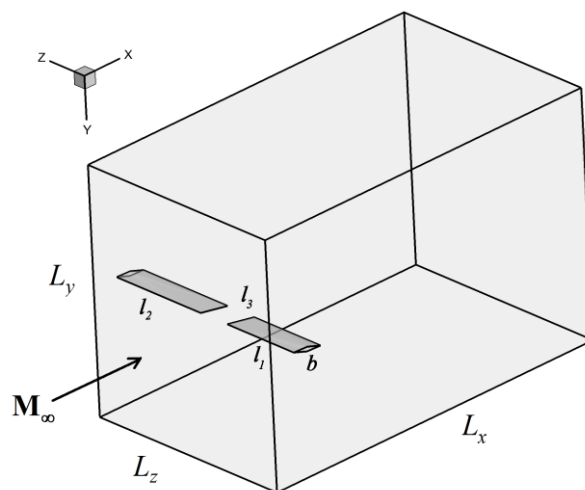


Рис. 1. Расчетная область, соосные крылья-генераторы концевых вихрей

## 4 Визуализация вихрей

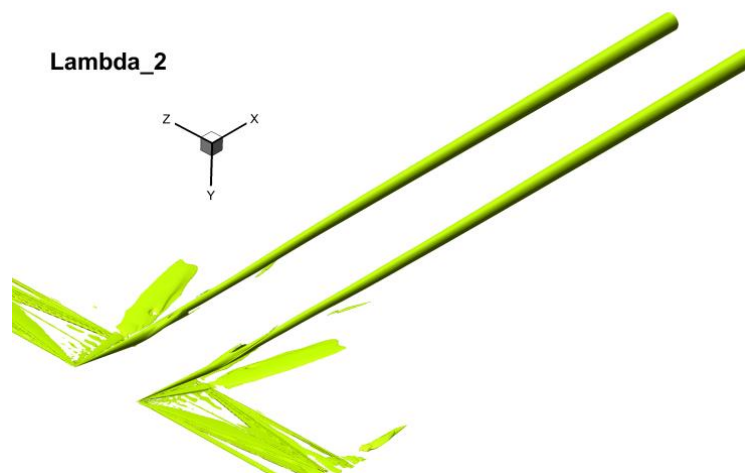
Внутри программного комплекса ARES был разработан отдельный постпроцессинговый модуль расчета данных для определения вихревых структур для гексагональных сеток. Метод максимальной завихренности в нем реализован полностью, а для применения  $\lambda_2$ -метода вычислены необходимые матрицы. Модуль формирует данные вихревых структур в формате пакета программ Tecplot, с помощью которой в дальнейшем осуществляется визуализация результатов расчетов.

### 4.1 $\lambda_2$ -метод

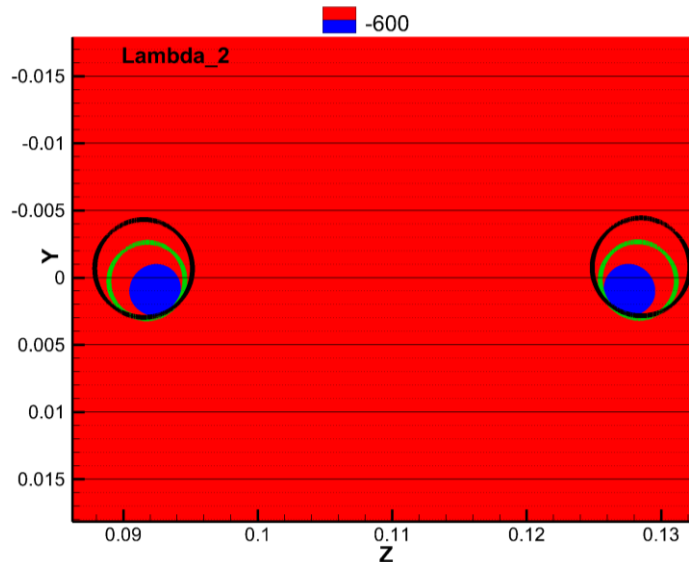
$\lambda_2$ -метод (или критерий) для идентификации вихрей был предложен в работе [8]. Согласно данному критерию область вихревого течения определяется, исходя из анализа собственных чисел симметричной матрицы  $S^2 + \Omega^2$ , которые всегда вещественны (здесь  $S$  и  $\Omega$  соответственно тензоры деформации и завихренности течения). Вихревой областью при этом считается часть пространства, в которой второе собственное число  $\lambda_2 < 0$  ( $\lambda_1 \geq \lambda_2 \geq \lambda_3$ ). Этот метод является достаточно широко распространенным и часто применяется при обработке данных.

На рис. 2 представлены результаты применения  $\lambda_2$ -метода для визуализации результатов расчетов описанной выше задачи. Этот метод позволяет найти область вихревого течения, ограниченную изолиниями отрицательного значения собственного числа  $\lambda_2$ . На рис. 3 изображены линии уровня  $\lambda_2 = -600$  в сечениях, перпендикулярных направлению набегающего потока:  $x = 0.1$ ,  $x = 0.2$ ,  $x = 0.3$ . В сечении  $x = 0.1$  значения  $\lambda_2$  изображены заполненным красным и синим цветом, в сечении  $x = 0.2$  – зеленым кругом, в сечении  $x = 0.3$  – черным кругом. Отмече-

но смещение противоположно вращающихся вихрей вверх (в подветренную сторону), что соотносится с данными других авторов [9] и их расхождение на данных дистанциях вниз по потоку от оси крыльев. Отмечено расширение диаметра зоны вихря по мере удаления вниз по потоку от оси крыльев.



**Рис. 2.** Пара противоположно вращающихся сверхзвуковых концевых вихрей, визуализированная с помощью  $\lambda_2$ -метода (изолинии уровня  $\lambda_2 = -600$ )



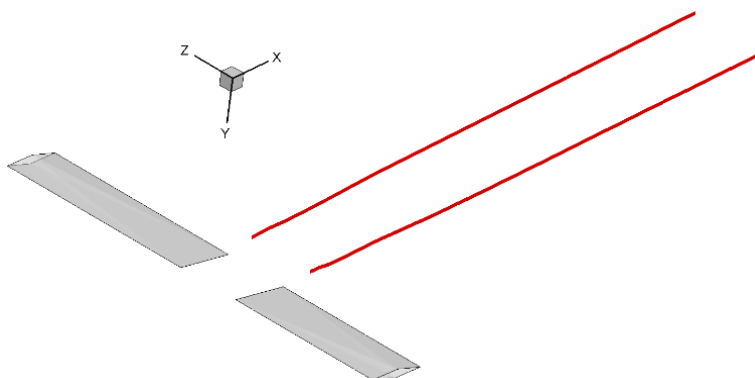
**Рис. 3.** Линии уровня  $\lambda_2 = -600$  в сечениях  $x = 0.1$  (разделение красного и синего цветов),  $x = 0.2$  (зеленый круг),  $x = 0.3$  (черный круг)

#### 4.2 Метод максимальной завихренности

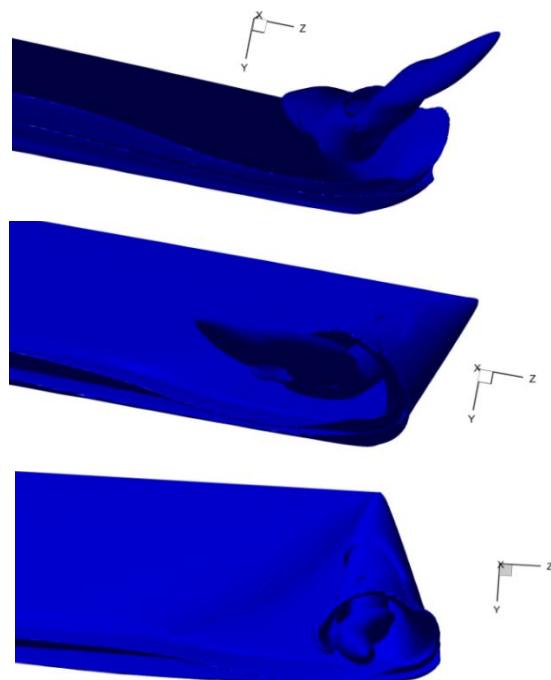
Метод максимальной завихренности был предложен в работе [10]. Он основан на одном из определений вихревого течения и заключается в нахождении локального максимума модуля вектора завихренности в плоскости, перпендикулярной направлению этого вектора. Этот метод позволяет определить, например, точную ось продольного вихря в случае достаточного разрешения расчетной сетки.

На рис. 4 показаны оси концевых вихрей, найденные методом максимальной завихренности для рассмотренной выше задачи (красные линии).

Отдельный интерес представляет собой визуализация области формирования вихря (зона сворачивания жгута концевой вихрей), более детальное представление о которой позволяют получить изолинии завихренности. На рис. 5 видно, как пелена с концевой кромки крыла сворачивается в жгут.

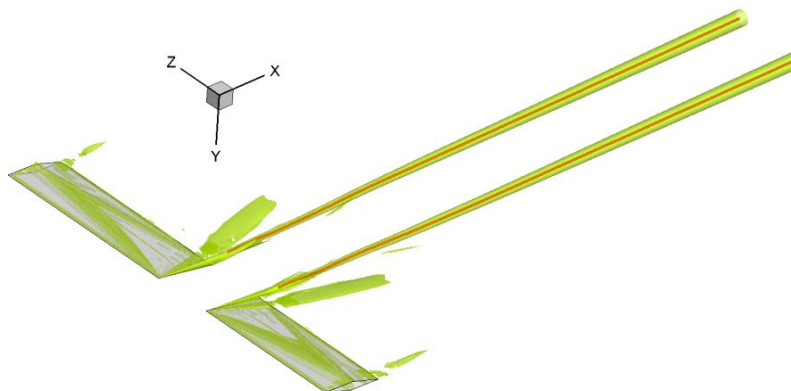


**Рис. 4.** Оси пары противоположно вращающихся сверхзвуковых концевых вихрей (красные линии)



**Рис. 5.** Зона сворачивания жгута концевых вихря: изолинии завихренности равной 2000 для крыла с полуразмахом 0.075 с разных ракурсов

На рис. 6 показано совместное положение осей пары вихрей, определенных методом максимальной завихренности (красные линии) в области сформированного вихря, и зоны вихрей, найденной  $\lambda_2$ -методом (зеленые изоповерхности),  $\lambda_2 = -600$ .



**Рис. 6.** Суперпозиция осей вихрей (красные линии), найденных методом максимальной завихренности, и области вихрей, полученной применением  $\lambda_2$ -метода,  $\lambda_2 = -600$

## 5 Заключение

В работе приведены результаты применения двух методов визуализации вихревых структур для пары сверхзвуковых противоположно вращающихся концевых вихрей:  $\lambda_2$ -критерий и метод максимальной завихренности.  $\lambda_2$ -критерий позволяет получить вихревую область вокруг моделируемых концевых вихрей, в то время как метод максимальной завихренности обеспечивает определение положения оси вихрей, а также наглядно демонстрирует сворачивание вихревого жгута. В целом можно отметить, что данные методы хорошо дополняют друг друга, позволяя получить подробную картину течения, которая может быть использована в дальнейшем для анализа процессов в вихревой зоне.

## Литература

1. Chakraborty, P., Balachandar, S., Adrian, R. G.: On the relationships between local vortex identification schemes. *J. Fluid Mech.* 535, 189-214 (2005)
2. Волков, К. Н.: Методы визуализации вихревых течений в вычислительной газовой динамике и их применение при решении прикладных задач. *Научно-технический вестник информационных технологий, механики и оптики* 14(3) (2014)
3. Jiang, M., Machiraju, R., Thompson, D.: Detection and visualization of vortices. *Visualization Handbook*: Elsevier, Burlington, pp. 295–309 (2005)
4. Быков, Л.В., Молчанов, А.М., Щербаков, М.А., Яньшев, Д.С.: Вычислительная механика сплошных сред в задачах авиационной и космической техники. ЛЕНАНД, Москва, 688 (2015)
5. Борисов В.Е., Давыдов А.А., Кудряшов И.Ю., Луцкий А.Е. Программный комплекс ARES для расчета трехмерных турбулентных течений вязкого сжимаемого газа на высокопроизводительных вычислительных системах. Свидетельство о регистрации программы для ЭВМ RU 2019667338, 23.12.2019.
6. Борисов В.Е., Луцкий А.Е. Моделирование перехода между регулярным и маховским отражением ударных волн с помощью неявной схемы на основе методов LU-SGS и BiCGStab. *Препринты ИПМ им. М.В. Келдыша*. 2016. № 68. 36 с.
7. Allmaras, S. R., Johnson, F. T., Spalart, P. R.: Modifications and Clarifications for the Implementation of the Spalart-Allmaras Turbulence Model. 7th Int. Conf. on CFD (ICCFD7) 2012, Big Island, Hawaii (9-13 July 2012)
8. Jeong, J., Hussain, F.: On the identification of a vortex. *Journal of Fluid Mechanics* 285, 69–94 (1995)
9. Forster, K. J., Barber, T. J., Diasinos, S., Doig, G.: Interaction of a counter-rotating vortex pair at multiple offsets. *Experimental Thermal and Fluid Science* J. 86, 63-74 (2017)
10. Strawn, R.C., Kenwright, D.N., Ahmad, J: Computer visualization of vortex wake systems. *AIAA Journal* 37(4), 511–512 (1999)



**ГЕОМЕТРИЧЕСКОЕ  
МОДЕЛИРОВАНИЕ.  
КОМПЬЮТЕРНАЯ ГРАФИКА В  
ОБРАЗОВАНИИ**

---

**GEOMETRIC MODELING.  
COMPUTER GRAPHICS IN  
EDUCATION**

**GraphiCon 2020**

---

22-25 сентября 2020  
Санкт-Петербург, Россия

## Геометрическая модель формообразования рабочих поверхностей элементов конической передачи\*

С.А. Рязанов<sup>1</sup>[0000-0001-7469-1180]

<sup>1</sup> Саратовский государственный технический университет им. Гагарина Ю.А.,  
410054, Саратов, Россия

**Аннотация.** Предложен метод твердотельного геометрического моделирования технологических процессов формообразования пространственных зубчатых передач в 3D пространстве. Рабочие и переходные поверхности зубьев образуются как огибающие производящей поверхности инструмента. В результате образуются конечные формы поверхностей зубьев в виде точных компьютерных твердотельных моделей. Приведены блок-схема и пример «компьютерного нарезания» конической передачи.

**Ключевые слова:** компьютерное моделирование, зубчатое зацепление, коническая передача, твердотельное моделирование.

### 1 Введение

Зубчатые передачи являются ответственными элементами многих современных машин и механизмов. К их основным качественным показателям относятся точность передачи вращения, нагрузочная способность, малошумность, ресурс. Перечисленные критерии обеспечиваются технологическими средствами на стадии операций зубопрофилирования, где формируются рабочие поверхности зубьев. Технология изготовления в большинстве случаев построена на реализации метода обкатки, как на традиционных, так и на многокоординатных станках различных компоновок.

При подготовке производства каждого нового вида изделий выполняется расчет наладочных установок зуборезного оборудования. Для этого в настоящее время используются расчетные методики и компьютерные программы, основанные на математических зависимостях теории зацепления [1, 4, 5]. Уточнение расчетных параметров и величины корректирующих поправок в наладки определяются путем нарезания нескольких опытных образцов.

Использование современных методов компьютерной 3D графики позволяют усовершенствовать и ускорить процесс проектирования технологических операций зубопрофилирования, обеспечивая получение конечных форм поверхностей зубьев в виде наглядных и точных компьютерных твердотельных моделей

---

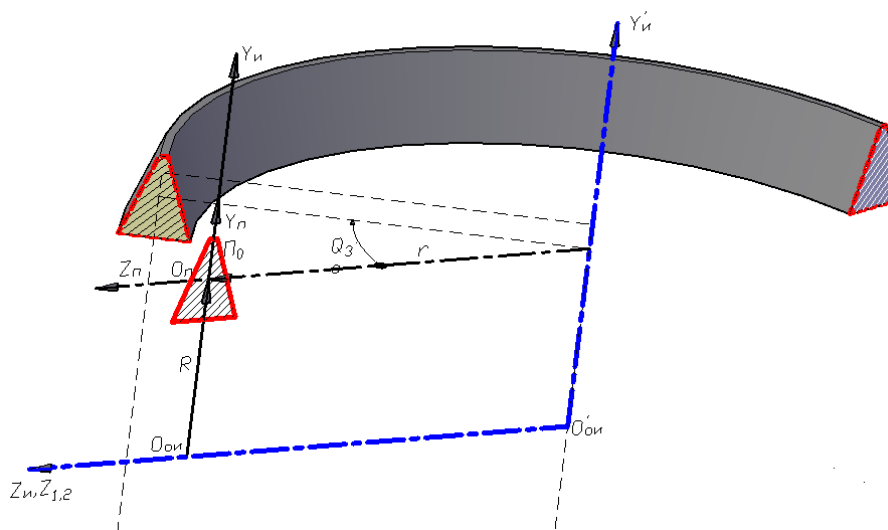
\* Публикация выполнена при поддержке гранта РФФИ №19-07-00844



[2, 3, 10]. Способ основан на виртуальном представлении процесса формообразования в виде взаимопересечения твердотельных 3D моделей двух объектов (инструмента и заготовки), совершающих в общем случае винтовое относительное движение [8]. В результате рабочие поверхности зубьев образуются как огибающие производящей поверхности инструмента.

## 2 Геометрическая модель производящей поверхности

В векторно-матричном виде поворот сечения исходного профиля производящей поверхности происходит на угол  $\theta_3$  вокруг оси  $Y_3$ . При этом, если величина угла  $\theta_3$  меняет свое значение от  $0^\circ$  до  $360^\circ$ , то получаем имитацию вращения производящего осевого профиля инструмента  $\Pi_O$  как на зуборезном станке [14-16]. Это приведет к получению компьютерной модели инструмента эквивалентного зуборезному инструменту «зуборезная головка» [9, 11] (рис. 1).



**Рис. 1.** Геометрическая модель образования производящей поверхности, эквивалентной производящей поверхности зуборезного инструмента «зуборезная головка»

В общем виде вращение исходного осевого профиля  $\Pi_O$  вокруг оси  $Y_3$  описывается с помощью матрицы поворота сечения (1):

$$\bar{R}_m^{(Su)} = H_{S_{oi}S_{o\Pi}} \times \bar{R}_m^{(So\Pi)}, \quad (1)$$

где

$$H_{S_{ou}S_{oII}} = \begin{vmatrix} A_{S_{ou}S_{oII}} & r_{oII}^{(s_{ou})} \\ 0 & 1 \end{vmatrix}, \quad (2)$$

$$r_{oII}^{(s_{ou})} = [-r \cdot \sin(\theta_3) \quad R \quad r \cdot \cos(\theta_3)]^T = \begin{bmatrix} -r \cdot \sin(\theta_3) \\ R \\ r \cdot \cos(\theta_3) \end{bmatrix}. \quad (3)$$

Для описания координат точек, принадлежащих производящей поверхности инструмента «зуборезная головка», в формуле (2) необходимо задать следующие параметры:

$$A_{S_{ou}S_{oII}} = \begin{vmatrix} \cos(\theta_3) & 0 & -\sin(\theta_3) \\ 0 & 1 & 0 \\ \sin(\theta_3) & 0 & \cos(\theta_3) \end{vmatrix}. \quad (4)$$

Подставляя значения блочной матрицы  $A_{S_{ou}S_{oII}}$  (4) и вектора  $r_{oII}^{(s_{ou})}$  (3) в формулу (2), получим окончательный вид матрицы, описывающей радиус-вектор формообразующей поверхности инструмента:

$$\bar{R}_m^{(Su)} = \begin{vmatrix} \cos(\theta_3) & 0 & -\sin(\theta_3) & -r \cdot \sin(\theta_3) \\ 0 & 1 & 0 & R \\ \sin(\theta_3) & 0 & \cos(\theta_3) & r \cdot \cos(\theta_3) \\ 0 & 0 & 0 & 1 \end{vmatrix} \times \bar{R}_m^{(SoII)}, \quad (5)$$

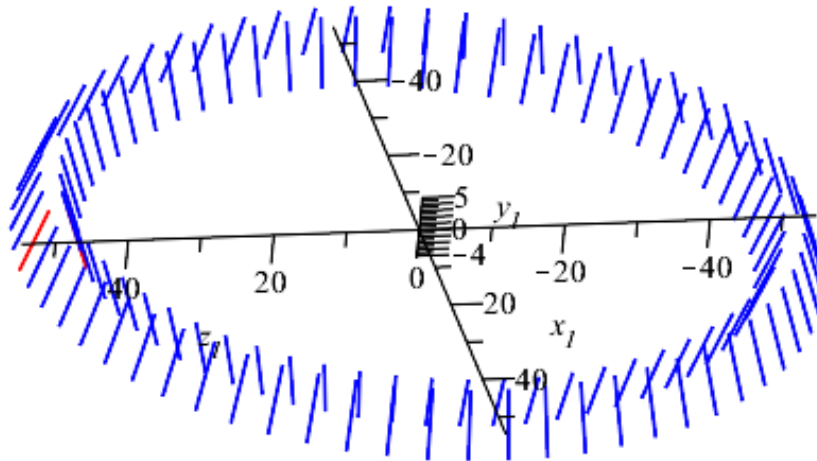
где  $\bar{R}_m^{(SoII)}$  – радиус-вектор осевого профиля производящей поверхности в подвижной системе координат  $S_{oII}(O_{II}x_{II}y_{II}z_{II})$ ;  
 $r$  – радиус зуборезной головки, расстояние от оси вращения  $Y_3$  до начала системы координат исходного профиля производящей поверхности  $S_{oII}(O_{II}x_{II}y_{II}z_{II})$ ;  
 $\theta_3$  – угол поворота радиус-вектора  $\bar{R}_m^{(SoII)}$ ;  
 $R$  – расстояние между началом системы координат осевого профиля производящей поверхности и осью  $Z_u$ .

Выполнив умножение матриц (5) получаем систему уравнений, которая описывает преобразование координат осевого профиля производящей поверхности из системы координат  $S_{oII}(O_{II}x_{II}y_{II}z_{II})$  в систему координат инструмента «зуборезная головка»  $(X_3Y_3Z_3)$  в координатной форме:

$$\begin{cases} R_{mX}^{(Su)} = x_m \cdot \cos(\theta_3) - z_m \cdot \sin(\theta_3) - r \cdot \sin(\theta_3) \\ R_{mY}^{(Su)} = y_m + R \\ R_{mZ}^{(Su)} = x_m \cdot \sin(\theta_3) + z_m \cdot \cos(\theta_3) + r \cdot \cos(\theta_3) \end{cases}, \quad (6)$$

где  $R_{mX}^{(Su)}$ ,  $R_{mY}^{(Su)}$ ,  $R_{mZ}^{(Su)}$  – координаты точки, принадлежащей производящей поверхности, которая задана радиус-вектором  $\vec{R}_m^{(So\Pi)}$ ;  $x_m$ ,  $y_m$ ,  $z_m$  – координаты точек осевого профиля производящей поверхности заданных в системе координат  $S_{o\Pi}(O_{\Pi}x_{\Pi}y_{\Pi}z_{\Pi})$ .

Система уравнений (6) позволяет рассчитать и получить положение точки сечения производящей поверхности в виде зуборезного инструмента «дисковая фреза» (рис. 2).



**Рис. 2.** Каркасная модель производящей поверхности, эквивалентная производящей поверхности зуборезного инструмента «зуборезная головка»

Подставив в систему уравнений (6) значение  $x_m = 0$ , т. к. сечение в подвижной системе координат находится в плоскости  $Y_{\Pi}Z_{\Pi}$ , получим системы уравнений производящей поверхности в параметрическом виде для левого (7) и правого (8) профилей производящей поверхности:

$$\begin{cases} R_{mX}^{(Su)} = -\sin(\theta_3) \cdot z_{left} - r \cdot \sin(\theta_3) \\ R_{mY}^{(Su)} = -\frac{z_{left}}{\operatorname{tg}(\alpha_{left})} + \frac{1}{4} \cdot \frac{\pi \cdot m}{\operatorname{tg}(\alpha_{left})} + R, \\ R_{mZ}^{(Su)} = \cos(\theta_3) \cdot z_{left} + r \cdot \cos(\theta_3) \end{cases}, \quad (7)$$

$$\begin{cases} R_{MX}^{(Su)} = -\sin(\theta_3) \cdot z_{right} - r \cdot \sin(\theta_3) \\ R_{MY}^{(Su)} = \frac{z_{right}}{\operatorname{tg}(\alpha_{right})} + \frac{1}{4} \cdot \frac{\pi \cdot m}{\operatorname{tg}(\alpha_{right})} + R \\ R_{MZ}^{(Su)} = \cos(\theta_3) \cdot z_{right} + r \cdot \cos(\theta_3) \end{cases} \quad (8)$$

Выполнив расчеты, получаем производящую поверхность заданную системами параметрических уравнений (7) и (8) (рис. 3). Эта поверхность является эквивалентной производящей поверхности зуборезного инструмента «зуборезная головка» [13].

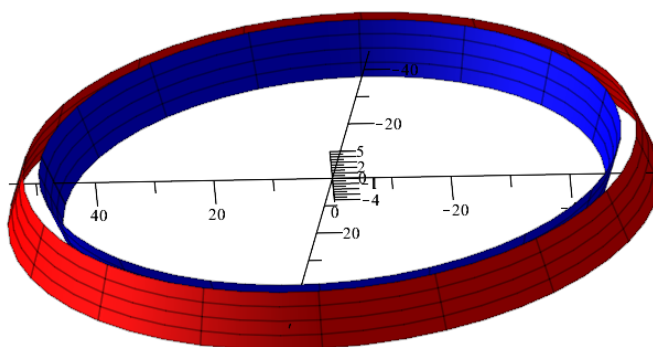


Рис. 3. Параметрическая модель производящей поверхности, эквивалентная производящей поверхности зуборезного инструмента «зуборезная головка»

Для получения модифицированной производящей поверхности необходимо выполнить изменение исходного (линейного) профиля сечения производящей поверхности, т. е. получить модифицированный осевой профиль производящей поверхности  $\Pi_O$ .

Осевой профиль  $\Pi_O$  модифицированной производящей поверхности задается отклонениями  $\Delta\mu_i$  от исходного прямолинейного профиля производящей поверхности в направлении нормали  $\bar{N}$  в средней расчетной точке профиля  $M_P$  в табличной форме или аналитически  $\Delta\mu_i = f(L)$  в виде функции линейного параметра  $L$  (рис. 4).

Для обеспечения дополнительной модификации поверхностей обрабатываемых зубьев можно предусмотреть дополнительный отвод осевого профиля производящей поверхности относительно уже модифицированного осевого профиля производящей в виде характеристики  $\Delta h$ , которую можно задавать в табличной форме.

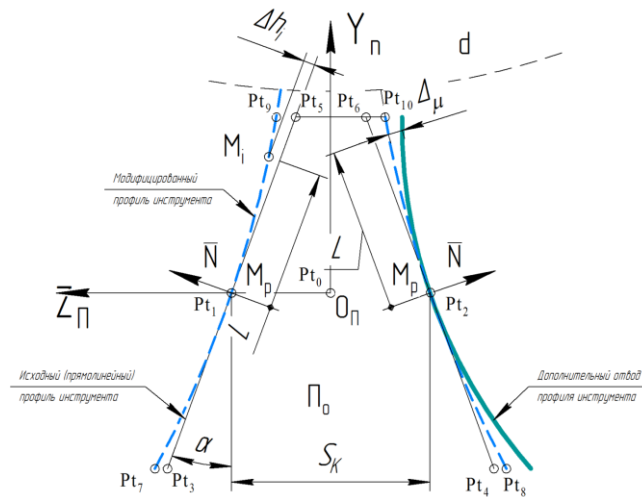


Рис. 4. Исходный осевой профиль обобщенной производящей поверхности

Для обеспечения дополнительной модификации поверхностей обрабатываемых зубьев можно предусмотреть дополнительный отвод осевого профиля производящей поверхности относительно уже модифицированного осевого профиля производящей в виде характеристики  $\Delta h$ , которую можно задавать в табличной форме.

Основную модификацию осевого сечения исходного профиля производящей поверхности зададим в квадратичной форме:

$$\Delta\mu = (\mu \cdot L^2) / 2, \quad (9)$$

где  $\Delta\mu$  – отклонение от линейного осевого сечения в текущей точке линейного профиля;

$\mu$  – коэффициент, который позволяет изменять величину отвода модифицированного сечения относительно исходного линейного сечения производящей поверхности;

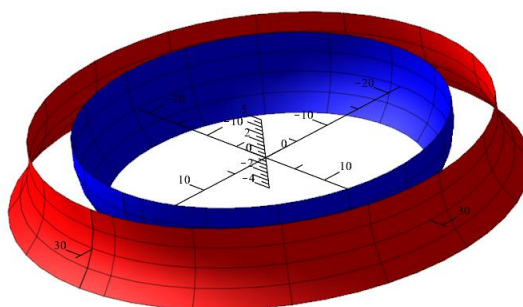
$L$  – величина шага вдоль исходного линейного сечения теоретической производящей поверхности.

Подставив уравнение (9) в уравнение (6), получаем параметрические зависимости, которые позволяют сформировать модифицированную производящую поверхность инструмента «резцовая головка». При этом система уравнений (10) позволяет сформировать левый, а система уравнений (11) правый модифицированные профили этой производящей поверхности:

$$\begin{cases} R_{MX}^{(Su)} = -\sin(\theta_3) \cdot (\mu \cdot z^2 \cdot \sin(\theta_x) + z \cdot \cos(\theta_x) + \Delta z) - r \cdot \sin(\theta_3) \\ R_{MY}^{(Su)} = \mu \cdot z^2 \cdot \cos(\theta_x) - z \cdot \sin(\theta_x) + R \\ R_{MZ}^{(Su)} = \cos(\theta_3) \cdot (\mu \cdot z^2 \cdot \sin(\theta_x) + z \cdot \cos(\theta_x) + \Delta z) + r \cdot \sin(\theta_3) \end{cases}, \quad (10)$$

$$\begin{cases} R_{MX}^{(Su)} = -\sin(\theta_3) \cdot (\mu \cdot z^2 \cdot \sin(\theta_x) - z \cdot \cos(\theta_x) - \Delta z) - r \cdot \sin(\theta_3) \\ R_{MY}^{(Su)} = \mu \cdot z^2 \cdot \cos(\theta_x) + z \cdot \sin(\theta_x) + R \\ R_{MZ}^{(Su)} = \cos(\theta_3) \cdot (\mu \cdot z^2 \cdot \sin(\theta_x) - z \cdot \cos(\theta_x) - \Delta z) + r \cdot \sin(\theta_3) \end{cases}. \quad (11)$$

Системы параметрических уравнений (10) и (11) позволяют сформировать компьютерную геометрическую модель модифицированной производящей поверхности, которая является эквивалентной производящей поверхности зуборезного инструмента «резцовая головка» с измененным формообразующим профилем (рис. 5).



**Рис. 5.** Производящая поверхность, эквивалентная производящей поверхности зуборезного инструмента «резцовая головка» с модифицированным исходным профилем

### 3 Компьютерное моделирование формообразования рабочих поверхностей конической передачи

В блоке 1 (рис. 6) осуществляется формирование твердотельных моделей заготовок и инструмента на основе применения стандартных процедур геометрического моделирования. Для этого, исходя из данных чертежа и предварительных расчетов, определяются геометрические характеристики осевых сечений заготовок (блок 1.1) и инструмента (блок 1.2), которым сообщается формообразующее вращательное движение, описываемое векторно-матричным преобразованием (1).

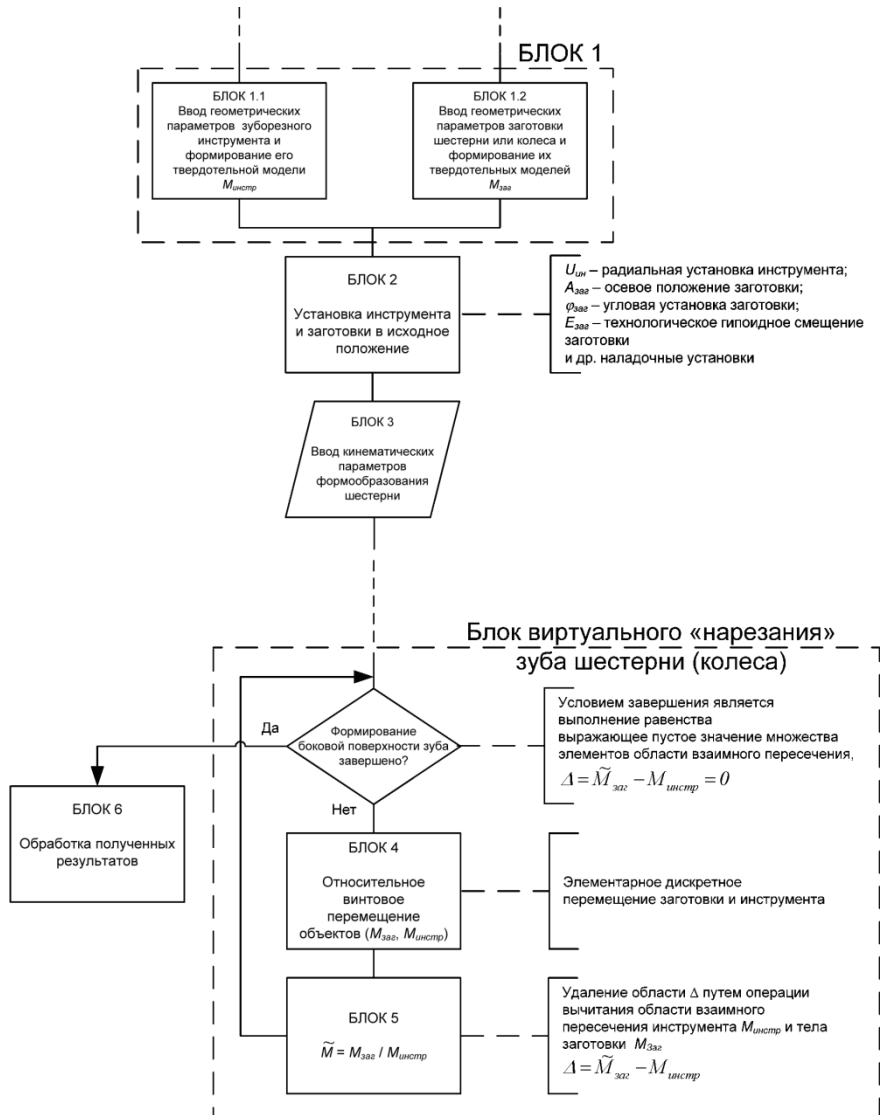


Рис. 6. Схема алгоритма «компьютерного нарезания» зубчатых передач

Сформированные твердотельные модели резцовой головки и заготовки, рассматриваемые далее как числовые множества, устанавливаются в номинальное исходное положение, определяемое радиальной установкой инструмента, технологическим гипоидным смещением, угловой и осевой установками заготовки (блок 2).

В блоке 3 вводятся кинематические зависимости, выражающие относительное движение объектов в станочном зацеплении. При моделировании традиционной схемы нарезания методом обкатки процесс формообразования задается

функциональной взаимосвязью угловых перемещений люльки станка с установленным инструментом и нарезаемой заготовки.

В блоках 4 и 5 выполняется виртуальное «нарезание» зубьев, представляющее собой последовательное удаление частей объема заготовки, отсекаемых инструментом, на основе применения булевых операций вычитания тел (в виде геометрического вычитания трехмерных числовых множеств):

$$M = M_{\text{заг}} \setminus M_{\text{инстр}}.$$

В блоке 6 выполняется анализ макро- и микрохарактеристик рабочих и переходных поверхностей сформированных зубьев методами визуализационной и метрологической оценки.

На рис. 7 и 8 показаны примеры формообразования виртуальных моделей шестерни и колеса конической зубчатой передачи.

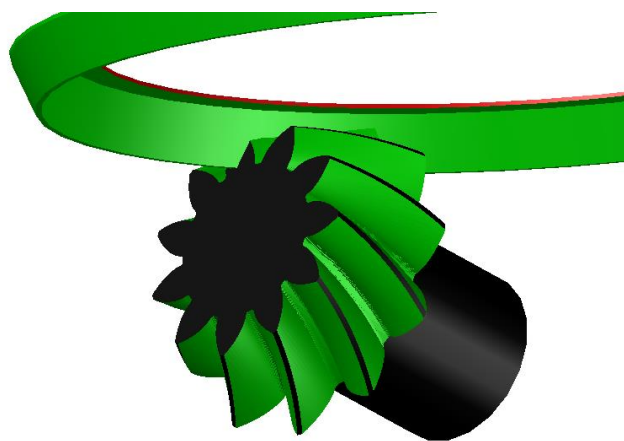


Рис. 7. Компьютерная модель шестерни

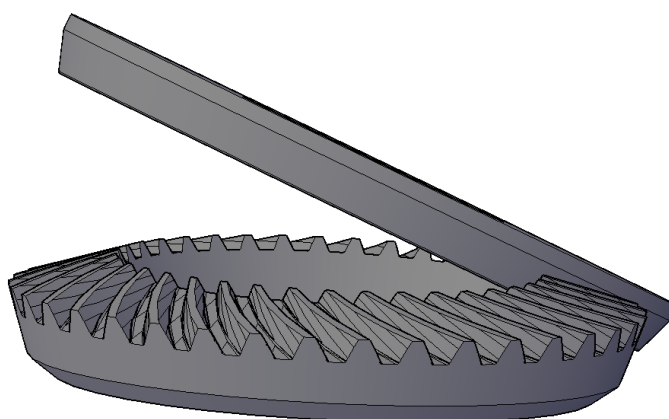


Fig. 8. Компьютерная модель колеса



## 4 Заключение

Особую актуальность разработанная методика приобретает в условиях ускоренного промышленного освоения современных технологических процессов изготовления сложных видов зубчатых передач на многокоординатном оборудовании, а также при синтезе новых разновидностей пространственных зубчатых передач [6, 7, 12].

## Литература

1. Вейц, В. Л. Геометрия зацепления зубчатых передач [Текст] / В. Л. Вейц, А. М. Волженская, Н. И. Колчин. – Л.: Машиностроение. 1978. – 136 с.
2. Волков А.Э. Компьютерное моделирование процессов формообразования поверхностей резанием // IV Международный конгресс: Конструкторско-технологическая информатика 2000: в 2 т. Москва: «СТАНКИН», 2000, Т. 1. С. 122-126.
3. Голованов, Н. Н. Геометрическое моделирование : учебник [Текст] / Н. Н. Голованов. – М. : Изд-во Физико-математической литературы, 2002. – 472 с.
4. Калашников С.Н. Зубчатые колеса и их изготовление [Текст] / С.Н. Калашников, А.С. Калашников. — М.: Машиностроение, 1983. — 264 с.
5. Калашников С.Н. Производство зубчатых колес: справочник / С.Н. Калашников [и др.]; под общ. ред. Б.А. Тайца. — 3-е изд., перераб. и дополн. — М.: Машиностроение, 1990. — 464 с.
6. Карачаровский В.Ю. Визуализационная оценка геометрических характеристик контактирующей поверхности в червячных передачах на основе применения методов 3D компьютерной графики [Текст] / В.Ю. Карачаровский, М.К. Решетников, С.А. Рязанов // Вестник Саратовского государственного технического университета. — 2010. — № 4. — Вып. 1. — С. 55–60.
7. Карачаровский В.Ю. Визуальная оценка параметров локализованного контакта в червячных передачах на основе применения методов 3D компьютерной графики [Текст] / В.Ю. Карачаровский, С.А. Рязанов // Вестник Саратовского государственного технического университета. — 2011. — № 2. — Вып. 2. — С. 73–77.
8. Карачаровский В.Ю. Геометрическое моделирование формообразования пространственных поверхностей при винтовом относительном движении [Текст] / В.Ю. Карачаровский, С.А. Рязанов // Проблемы геометрического моделирования в автоматизированном проектировании и производстве: 1-я Междунар. науч. конф. — М.: Изд-во МГИУ, 2008. — С. 143–146.
9. Карачаровский В.Ю. Обобщенная модель твердотельного зуборезного инструмента с изменяемой кривизной производящей поверхности [Текст] / В.Ю. Карачаровский, С.А. Рязанов // Совершенствование подготовки учащихся и студентов в области графики, конструирования и стандартизации: межвуз. науч.-метод. сб. — Саратов: Изд-во СГТУ, 2010. — С. 152–156.
10. Карачаровский В.Ю. Применение методов компьютерной 3D графики и твердотельного моделирования при разработке технологических процессов зубонарезания [Текст] / В.Ю. Карачаровский, С.А. Рязанов // Вестник Саратовского государственного технического университета. — 2010. — № 4. — Вып. 1. — С. 55–60.
11. Карачаровский В.Ю. Разработка цифровых технологий твердотельного моделирования процессов формообразования пространственных зубчатых передач для реализа-

- ции на многокоординатных станках [Текст] / В.Ю. Карачаровский, С.А. Рязанов // Пятый Саратовский салон изобретений, инноваций и инвестиций: в 2 ч. — Саратов: Изд-во СГТУ, 2010. Ч. 1. — С. 150–151.
12. Карачаровский, В. Ю. Геометрическое моделирование и визуальная оценка винтовой поверхности зуба [Текст] / В. Ю. Карачаровский, М. К. Решетников, С. А. Рязанов // Проблемы графической подготовки студентов технических вузов в условиях современного компетентного подхода к процессу обучения : сб. тр. междунар. науч.-метод. конф. – Дивноморское : ДГТУ, 2015. – С. 59-66.
  13. Карачаровский, В. Ю. Определение координат точек эвольвентной винтовой поверхности червячной передачи [Текст] / В. Ю. Карачаровский, М. К. Решетников, С. А. Рязанов // Проблемы графической подготовки студентов технических вузов в условиях современного компетентного подхода к процессу обучения : сб. тр. междунар. науч.-метод. конф. – Дивноморское : ДГТУ, 2015. – С. 66-73.
  14. Колчин Н.И. Аналитический расчет плоских и пространственных зацеплений [Текст] / Н.И. Колчин. — М.: Машгиз, 1949. — 210 с.
  15. Лопато Г.А. Конические и гипоидные передачи с круговыми зубьями. / Г.А. Лопато, Н.Ф. Кабатов, М.Г. Сегаль Изд. 2-е, перераб. и доп. – М.:Машиностроение , 1977. – 423с.
  16. Медведев В.И. Комплекс программ для диалогового подбора технологических и конструктивных параметров гипоидных и конических пар с круговыми зубьями // IV Международный конгресс: Конструкторско-технологическая информатика 2000: в 2 т. Москва: «СТАНКИН», 2000, Т. 2. С. 51-54.

# Использование WebGL для визуализации учебных материалов в образовательном процессе

Е.В.Тетюев<sup>1</sup>, Ю.А. Жук<sup>1</sup>

<sup>1</sup> СПбГЛТУ им. С.М. Кирова, 194021, Санкт-Петербург, Россия

**Аннотация.** Цифровизация современного образования подразумевает оцифровку учебных материалов. Оцифровка текста или рисунков не представляет сложности, однако, создание и визуализация трёхмерной цифровой копии физического объекта – нетривиальная задача. Эта статья рассказывает о WebGL – браузерной технологии для работы с графикой, которая может служить для визуализации оцифрованных учебных материалов.

**Ключевые слова:** WebGL, Three.js, 3D графика, обучение, цифровая образовательная среда, дистанционное обучение.

## 1 Тенденции в современном образовании

### 1.1 Вступление

Появление компьютеров и глобальной сети Интернет значительно отразилось на жизни современного общества. В первую очередь это выражается в доступности информации, которую теперь можно получить в считанные секунды из любой точки земного шара. Это стало возможным благодаря оцифровке уже имеющихся текстов, изображений и других данных. Более того, большинство современных текстов и статей в первую очередь издаются в электронном виде и уже затем, при необходимости, печатаются на бумаге.

Само собой, тенденция к цифровизации не обошла и сферу образования [1]. Преподаватели всё чаще используют в своих лекциях презентации и проекторы, многие учебные материалы были оцифрованы. Появилось множество электронных учебников, небывалое развитие получило дистанционное образование. Сегодня для учебного процесса разрабатываются специальные программы и целые операционные системы. Образование впитало последние достижения прогресса и снова оказалось перед вопросом: «Какие технологии будут использоваться завтра?».

### 1.2 Учебные материалы и их оцифровка

Классический учебный процесс построен из теоретической и практической части. Студенты слушают лекции, изучают учебники и на основе полученных

знаний составляют конспекты. Также студенты посещают практические занятия, выполняют лабораторные и иные работы.

На данный момент оцифровка большинства учебных материалов не составляет труда. Существует множество алгоритмов для распознавания текста, обучающие фильмы с лёгкостью кодируются в современные цифровые форматы, а изображения не только переводятся в цифру, но, при необходимости, могут быть улучшены с помощью простых алгоритмов или даже с использованием нейросетей [2].

Однако, когда речь заходит об оцифровке каких-либо учебных материалов, представляющих собой физические объекты, возникают определённые трудности, как в выборе средства для оцифровки, так и в выборе технологии для визуализации. Макеты, натурные образцы, детали машин, приборы, инструменты можно отсканировать на фото или видео, но можно представить и в виде 3D модели. Необходимость оцифровки таких материалов может быть обоснована следующими причинами:

- Уникальность и ценность исходного образца, которая подразумевает ограничение к нему доступа для широкой публики.
- Слишком большой или малый размер объекта.
- Улучшение качества наглядного пособия путём добавления к нему дополнительной информации, деталей или функционала.
- Необходимость использования цифровой копии объекта в электронном пособии или дистанционном образовании.

Данная статья не рассматривает способы создания 3D модели, но фокусируется на технологии визуализации, которую разработчики обучающих материалов часто незаслуженно обходят стороной.

## 2 Технология WebGL

### 2.1 Краткая характеристика

WebGL (англ. Web Graphics Library или Библиотека Веб-Графики) – API<sup>†</sup> для визуализации интерактивной 2D и 3D графики в любом совместимом браузере без использования дополнительных плагинов. WebGL основывается на OpenGL ES и выполняется в рамках элемента <Canvas> структуры HTML<sup>‡</sup> DOM<sup>§</sup> [3].

Технология WebGL вышла в свет в 2011 году и присутствует во многих современных браузерах, в том числе: Firefox, Chrome, Opera, Safari, Internet Explorer и

---

<sup>†</sup> Application Program Interface (интерфейс прикладного программирования) – описание методов и средств, которыми компьютерная программа взаимодействует с другими программами.

<sup>‡</sup> HyperText Markup Language (язык гипертекстовой разметки) – стандартизированный язык разметки документов в глобальной сети.

<sup>§</sup> Document Object Model (объектная модель документа) – программный интерфейс, позволяющий программам и скриптам получать доступ к элементам HTML-документа, а также менять их содержимое и структуру.

Microsoft Edge. Разработкой занимается некоммерческая организация Khronos Group в рамках стандартизации W3C\*\*.

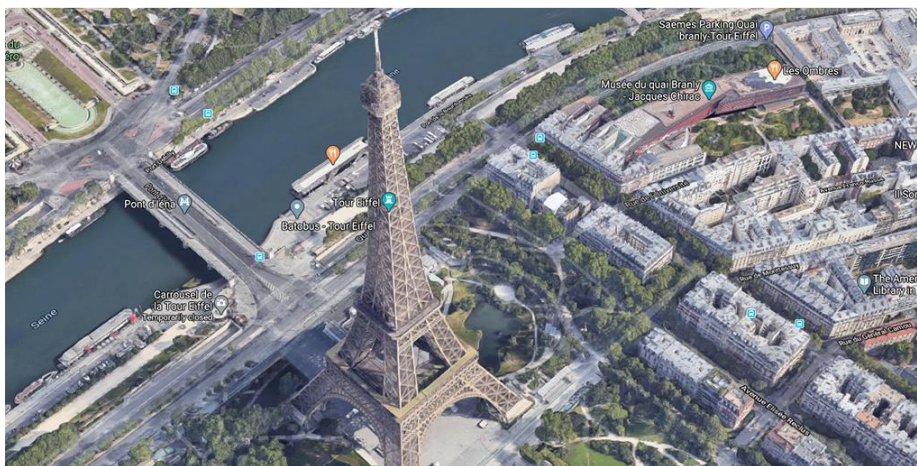
Говоря простыми словами, WebGL – это модуль для отображения 3D графики внутри обыкновенной Интернет страницы. Создание сцены, наполнение её объектами и интерактивная составляющая осуществляется средствами JavaScript и позволяет создавать различные сценарии поведения, как для самой сцены, так и для наполняющих её объектов.

Далее в статье приложением на базе WebGL будем называть совокупность из HTML-документа, его стилей на базе CSS, а также интерактивной составляющей, реализованной на языке JavaScript.

## 2.2 Примеры использования

Ранее в статье упоминалось, что WebGL не пользуется большой известностью. Тем не менее, многие из нас видят работу этой технологии практически каждый день.

Например, на картах компании Google в режиме «Спутник» при достаточном приближении можно увидеть, как объекты из плоских изображений становятся объёмными (рис. 1). Если же достаточно отдалить камеру, можно увидеть весь земной шар и даже переключиться на другие объекты Солнечной системы. Также WebGL используется и для отображения панорамных снимков, например, в сервисе Яндекс.Карты.



**Рис. 9.** Эйфелева башня в Париже на картах Google Maps. Помимо самой башни на изображении отчётливо видны возвышающиеся над плоскостью дома, деревья и другие объекты на карте.

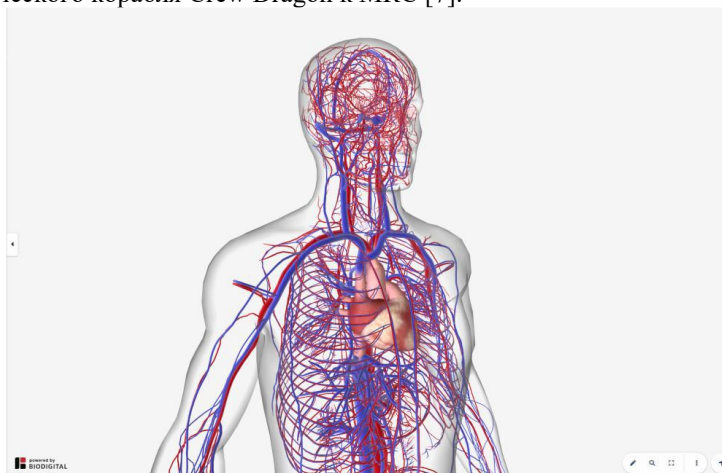
\*\* World Wide Web Consortium – организация, разрабатывающая и внедряющая стандарты для всемирной паутины.

Одним из примеров оцифровки и визуализации объектов физического мира может служить виртуальная модель Театра-Музея Дали [4], в которой пользователь может путешествовать по музею и детально рассматривать его интерьеры и экспонаты. Несмотря на то, что визуальный пример почти не отличить от «панорамы в шаре», поведение объектов в кадре, положение указателя и общая схема помещений однозначно говорят в пользу того, что это именно интерактивная сцена с множеством объектов.

Специалисты, работающие с 3D графикой, наверняка пользовались или хотя бы слышали о SketchFab [5] – облачном хранилище для моделей, одной из отличительных особенностей которого является возможность просмотреть загруженную модель без необходимости сохранять её на локальный диск.

Те, кто, так или иначе, связан с медициной, скорее всего, также сталкивались с визуализацией анатомических материалов на базе WebGL. Например, проект BioDigital [6] предоставляет электронные анатомические пособия по анатомии и здоровью человека (рис. 2).

В качестве заключительного примера приведём проект компании SpaceX, которая в рамках рекламной кампании создала на базе WebGL симулятор стыковки космического корабля Crew Dragon к МКС [7].



**Рис. 10.** Интерактивная схема сердечно-сосудистой системы человека на сайте BioDigital [5]. Модель позволяет управлять камерой и получить информацию о выбранном объекте.

### 2.3 Преимущества и недостатки

Использование WebGL имеет ряд преимуществ и недостатков, которые большей частью основываются на том, что это – браузерная технология. К преимуществам следует отнести:

- Кроссбраузерность и кроссплатформенность. Согласно стандартам W3C приложения WebGL выполняются одинаково на всех совместимых браузерах и устройствах.
- Интерактивность достигается использованием языка JavaScript и создания различных сценариев, событий и зависимостей.
- Доступность. Будучи загруженным на сервер, приложение WebGL будет доступно из любой точки земного шара.
- Самым главным преимуществом является то, что, являясь по сути веб-страницей и набором скриптов для интерактивности, приложения на базе WebGL не требуют установки. Для их работы достаточно браузера и доступа в сеть.

Само собой, не обошлось и без недостатков. Как и преимущества, они проистекают из браузерной природы интерфейса:

- Быстродействие WebGL хуже по сравнению со специализированными приложениями для работы с графикой. Причина в том, что между исполняемым кодом и графическим ядром добавился интерпретатор JavaScript и непосредственно браузер.
- Ограничения при работе с локальными файлами. Поскольку загрузка из локальных файлов в Web-браузерах заблокирована [8], для загрузки моделей популярных форматов вроде FBX или OBJ необходимо использование физического или виртуального сервера.
- Расходование трафика при использовании сервера.

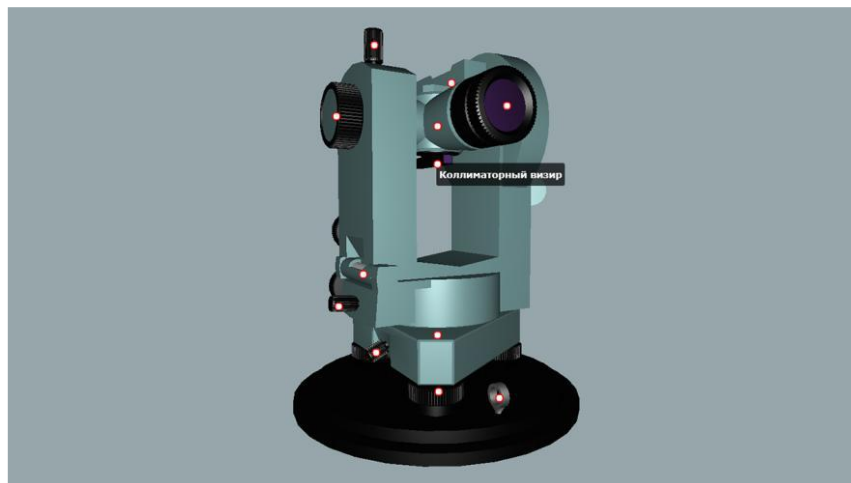
Несмотря на упомянутые недостатки, преимущества WebGL неоспоримы, и эта технология активно применяется в сферах развлечений и бизнеса. Мы же надеемся, что она найдёт своё применение и в образовательной среде.

### 3 Возможности применения WebGL в образовании

#### 3.1 Примеры использования

Примеры для этой главы взяты из выпускной квалификационной работы бакалавра Тетюева Е.В. [9], в рамках которой автор разработал несколько WebGL-приложений для решения проблем, с которыми сталкивались студенты при обучении в университете.

Для упрощения разработки была использована библиотека Three.js, которая расширяет базовый функционал WebGL. Также стоит отметить, что целью работы была демонстрация возможностей и концепции использования, поэтому приложения не обладают обширным функционалом или лоском и проработанным интерфейсом, присущим бизнес-решениям. Тем не менее, подобные приложения обладают определённым потенциалом для использования и дальнейшего развития. Из представленных в работе примеров мы отобрали два наиболее интересных.



**Рис. 11.** Интерактивная модель теодолита Т30. На модели отчётливо видны точки, соответствующие деталям, а также всплывающая подсказка при наведении на точку.

Первый пример – интерактивная схема физического объекта, в данном случае это модель теодолита Т30 (рис. 3). В отличие от статичного изображения в классическом или электронном учебнике, подобную модель можно повернуть и рассмотреть под разными углами, а информация о деталях прибора вместо списка сносок высвечивается метками прямо на экране. В дальнейшем для данного приложения планируется доработка интерфейса и добавление функционала для имитации работы закрепляющих винтов, наведения зрительной трубы и снятия измерений.

Второй пример представляет собой интерактивную карту учебных корпусов. Предвосхищая вопрос читателя ответим, что проблема весьма актуальна, поскольку, несмотря на наличие указателей и схем, абитуриенты и даже студенты с завидным постоянством теряют ориентацию в учебном пространстве. Для помощи заблудшим душам была разработана интерактивная схема (рис. 4), которая визуализирует профиль этажа и помещений на нём, а также позволяет искать нужную аудиторию и прокладывать к ней путь. При этом пользователь может перемещать камеру и переключаться между этажами схемы.

В дальнейшем планируется составление полноценной схемы учебных зданий СПбГЛТУ, работа по усовершенствованию дизайна и функционала приложения, а также его оптимизация для комфортной работы на мобильных устройствах. Предположительно, доступ к интерактивной карте будет осуществляться по ссылке, зашифрованной в QR-коде.





**Рис. 12.** Интерактивная карта абстрактного учебного здания. На рисунке видны полупрозрачные профили неактивных этажей, а найденная аудитория подсвечена.

### 3.2 Перспективы использования

Некоторые скептики считают, что WebGL – бесперспективная технология. Действительно, упомянутые в предыдущей главе разработки не дают фотореалистичной картинке, не предоставляют прорывных алгоритмов оцифровки или работы с информацией и в принципе не претендуют на какие-либо номинации. Их интерфейс примитивен, а графика схематична. Более того, существует множество «классических» приложений, которые предоставляют аналогичный или гораздо более обширный функционал.

Использование специализированного программного обеспечения однозначно необходимо при обучении специалистов. Однако имеет ли смысл устанавливать такое приложение ради использования минимальной доли его функционала? Особенно этот вопрос актуален при изучении студентами непрофильных предметов, когда обучение работе с приложением занимает гораздо больше времени, чем непосредственная работа в нём.

Основная же идея приведённых примеров и самого использования WebGL в любой сфере человеческой деятельности заключается в том, чтобы сделать обращение к приложению максимально быстрым, простым и удобным, исключив использование специализированного программного обеспечения и, тем самым, избавив пользователя от необходимости его установки и настройки. Примеры и материалы, которые встроены в электронный документ или открываются по ссылке – это и есть основное преимущество и перспективное направление использования технологии.

Такие мини-приложения могут быть полезны во многих направлениях, как технических, так и гуманитарных. Будь то визуализация деталей машин и меха-

низмов, химическая формула, экспонат в коллекции музея или схема расположения артефактов в культурном слое – всё это можно реализовать с помощью WebGL. Если же прибавить к этому соответствующий интерфейс и сценарии на базе JavaScript, вполне возможно создать приложение, которое по функционалу сможет сравниться с классическим программным обеспечением.

## 4 Заключение

По спектру решаемых задач современный браузер сравним с операционной системой. Его можно использовать для общения, чтения новостей, прослушивания музыки, редактирования файлов, работы с видео и даже для администрирования серверов. Браузеры активно используются в образовании, преимущественно в дистанционном обучении, с каждым годом утилизируя всё больше и больше встроенных технологий.

Одна из таких технологий – WebGL – может быть полезна в создании обучающих материалов за счёт простой и удобной визуализации объектов. При этом, в отличие от видео и изображений, у этой технологии есть большие возможности для создания интерактивной среды.

Стандартной ситуацией в системе образования является её консервативное отношение к новым методам и технологиям обучения несмотря на то, что суть образования – в познании нового на базе уже имеющихся знаний. И, если сегодня WebGL активно используется в сферах развлечений, культуры и бизнеса, вполне вероятно, что завтра она найдёт применение и в образовании.

## Литература

1. Никулина Т.В., Стариченко Е.Б.: Информатизация и цифровизация образования: понятия, технологии, управление, Педагогическое образование в России, УрГПУ, Екатеринбург (2018).
2. Yan, Zhicheng & Zhang, Hao & Wang, Baoyuan & Paris, Sylvain & Yu, Yizhou: Automatic Photo Adjustment Using Deep Neural Networks. ACM Transactions on Graphics. 35. 10.1145/2790296 (2016).
3. WebGL API, документация разработчика, [https://developer.mozilla.org/en-US/docs/Web/API/WebGL\\_API](https://developer.mozilla.org/en-US/docs/Web/API/WebGL_API).
4. Виртуальный Театр-Музей Дали, <https://my.matterport.com/show/?m=K5MKrKcfyRW>.
5. Облачное хранилище Sketchfab, <https://www.sketchfab.com/>.
6. Образовательная платформа BioDigital, <https://www.biodigital.com/>.
7. Симулятор стыковки SpaceX Crew Dragon, <https://iss-sim.spacex.com/>.
8. Прохоренок Н.А., Дронов В.А.: HTML, JavaScript, PHP и MySQL. Джентльменский набор Web-мастера, БХВ-Петербург, СПб (2019).
9. Тетюев Е.В.: ВКР «Использование трёхмерной графики на основе WebGL для создания обучающих и демонстрационных материалов» (научный руководитель к.п.н., доцент Жук Ю.А.), СПбГЛТУ им. С.М. Кирова, Санкт-Петербург (2020).

## Формирование трехмерной каркасной модели в проблеме преобразования чертежно- конструкторской документации на бумажном носителе в электронную модель объекта\*

С.А. Роменский<sup>1</sup>, С.И. Ротков<sup>2</sup>

<sup>1</sup> Нижегородский государственный архитектурно-строительный университет  
romensky@nngasu.ru

<sup>2</sup> Нижегородский государственный архитектурно-строительный университет  
rotkov@nngasu.ru

**Аннотация.** Данная работа посвящена различным этапам формирования трехмерной каркасной модели при решении проблемы преобразования бумажных чертежей в чертежно-конструкторскую документацию. Подробно рассматриваются этапы получения геометро-графической информации прикладной программой и формирования трехмерной каркасной модели. Так же приводится исследование временных характеристик разработанного алгоритма.

**Ключевые слова:** каркасная модель, геометро-графическая информация, проекционные изображения.

### 1 Введение

Любой объект производства в течение долгого времени должен иметь информационную поддержку жизненного цикла изделия (ЖЦИ). Данная поддержка осуществляется при помощи совокупности различных систем, объединенных единой базой данных. CALS (англ. Continuous Acquisition and Lifecycle Support) — концепция, объединяющая принципы и технологии информационной поддержки жизненного цикла продукции на всех его стадиях, основанная на использовании интегрированной информационной среды (единого информационного пространства), обеспечивающая единообразные способы управления процессами и взаимодействия всех участников этого цикла: заказчиков продукции (включая государственные учреждения, и ведомства), поставщиков (производителей) продукции, эксплуатационного и ремонтного персонала, реализованная в соответствии с требованиями системы международных стандартов, регламентирующих правила указанного взаимодействия преимущественно посредством электронного обмена данными. Возможность автоматически восстанавливать

---

\* Работа выполнена при поддержке гранта РФФИ №19-07-01024.

образы объектов в процессе автоматического проектирования полностью соответствует современной идеологии CALS-технологий. Современные системы геометрии и графики, используемые на различных этапах поддержки жизненного цикла изделия, должны работать одновременно в двух пространствах измерений таким образом, чтобы между моделью объекта (3-х мерное пространство, далее 3D) и его изображением (2-мерное пространство, далее 2D) была двухсторонняя взаимно однозначная информационная связь (далее 3D <-> 2D). Однако подавляющее большинство систем предоставляют только связь 3D->2D, что существенно ограничивает возможности проектирования и не даёт конструктору пользоваться традиционными методами проектирования. Так же необходимо учитывать тот факт, что технический чертёж содержит всю информацию о проектируемом объекте в том виде, который естественен для конструктора, полностью соответствует государственному стандарту и легко может быть создан в электронном виде.

Создание системы, выполняющей автоматическую реконструкцию трехмерных моделей по техническому чертежу, позволило бы во много раз сократить время проектирования объектов различного назначения. Поэтому необходимо разрабатывать математическое и программное обеспечение системы автоматической реконструкции трехмерных моделей объектов по техническому чертежу.

Первые исследования в области восстановления 3D моделей по проекционным изображениям были проведены Х.И. Тани (1964) [1], В.А. Щеколдиным (1966) [2], и В.С. Полозовым в его докторской диссертации (1983) [3]. Аналогичные исследования были проведены профессором Masanori Idesawa в 1973 году и были представлены в его работе «A system to generate a solid figure from three views» [4]. Ими была поставлена проблема автоматизации процесса преобразования двумерной геометрической модели в трехмерную конструктивную модель.

Балабан О.М. [5] в своей кандидатской диссертации описал метод синтеза из двумерных проекционных изображений каркасных и граничных трехмерных моделей в рамках двух классов трехмерных объектов: тела вращения и многогранники.

Тюрина В.А. [6] предложила методы и алгоритмы автоматизации процесса преобразования трехмерной каркасной модели в граничную для объектов класса «деталь», причем проблема удаления ложных ребер каркасной модели решалась путем применения теории графов и фундаментальных циклов.

Существует система автоматизированного проектирования (САПР) «Китеж», созданная в 1985 году НИИ механики ННГУ им. Лобачевского группой разработчиков (Аристов Е.В., Зудин А.А., Лабутин С.Е., Митин С.В., Шубин В.П., Дергунов В.И., Лапшин М.Л., Полозов В.С., Ротков С.И.) [7], включающая в себя реализованный алгоритм преобразования технического чертежа в трехмерную каркасную модель, но данная САПР имеет существенные ограничения по количеству вершин в каркасной модели (256), осуществление строгого контроля входных данных, необходимость удаления всех элементов оформления чертежа перед восстановлением, отсутствие поддержки современных операционных систем.

## 2 Проблема передачи геометро-графической информации в прикладную программу

Для решения проблемы передачи данных между различными системами хранения и обработки геометро-графической информации (ГГИ) в рамках поддержки жизненного цикла изделия существуют международные стандарты передачи данных IGES и STEP (STandard for the Exchange of Product model data – стандарт для обмена данными о модели изделия), но для оперативной работы с геометро-графической информацией данный способ не эффективен из-за больших объемов побочных данных и необходимости двойной перекодировки, ведущей, в свою очередь, к возникновению ошибок. Для чтения же внутреннего формата хранения данных любой системы автоматизированного проектирования необходимо либо изучение открытой спецификации (если таковая предоставляется фирмой-разработчиком данной САПР) либо незаконное вскрытие внутренней структуры хранения данных при закрытой для сторонних разработчиков спецификации.

Электронная модель любого изделия, используемая в CALS-технологиях, на 80-90% состоит из ГГИ, создаваемой конструктором или технологом при помощи соответствующих программных средств, входящих в состав систем автоматизированного проектирования и технологической подготовки производства. Каждая из таких систем имеет свой, только ей присущий, экраный интерфейс, свою внутреннюю структуру данных и свои алгоритмы решения элементарных геометро-графических задач. Это как раз и приводит к проблемам передачи данных между составными частями систем поддержки жизненного цикла изделия, к трудностям при составлении прикладных программ конкретного пользователя, к проблемам чтения и интерпретации данных из внутренней структуры конкретной системы. Кроме того, внутренняя структура данных каждой системы является предметом коммерческой тайны фирмы-разработчика системы и тщательно охраняется, и не публикуется (за редкими исключениями).

Для обеспечения информационной совместимости между различными системами поддержки жизненного цикла изделия существуют два основных способа передачи данных:

- через программы – конвертеры данных. Данное решение обеспечивает прямую передачу ГГИ из одной системы в другую, что приводит к необходимости создавать дополнительную программу связи (рис. 1);

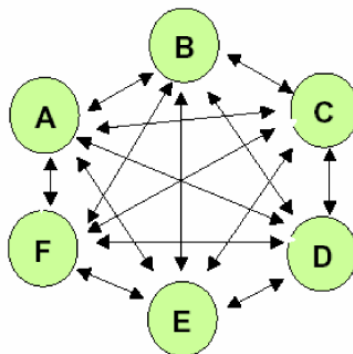


Рис. 13. Диаграмма связи форматов хранения различных систем

- через международные стандарты передачи ГГИ типа IGES или STEP (рис. 2). Недостатком данного способа является право фирмы-разработчика систем поддержки жизненного цикла изделия реализовывать лишь подмножество нейтрального формата, что неминуемо ведет к разночтениям при передаче и анализе данных в системе-приемнике.

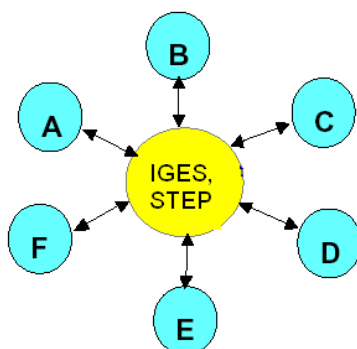


Рис. 14. Диаграмма связи форматов хранения различных систем при использовании международных стандартов передачи

В то же время в состав любой системы поддержки жизненного цикла изделия входит система управления базами данных (СУБД), которую можно использовать для целей оперативного хранения, анализа и передачи ГГИ. Это означает, что внутреннюю структуру данных системы геометрии и графики необходимо привести в соответствие с правилами создания баз данных.

Таким образом, появляется проблема создания прикладной программы, использующей внутренние форматы хранения данных систем автоматизированного проектирования. Для преодоления противоречия необходимо иметь операторы чтения и записи из внутренней структуры. Их предоставляет либо фирма-

разработчик САПР либо данные операции происходят на уровне текстовых фирменных форматов.

Далее рассмотрим перечень два основных наиболее значимых формата в контексте конвертации их в систему поддержки ЖЦИ.

## 2.1 Формат хранения .cdw

Используется в российской САПР «Компас», разработан компанией «Аскон». Предназначен для хранения и использования внутри данной САПР, содержит информацию о чертеже в бинарном виде. Содержит двухмерные данные, представляющие собой упорядоченный набор геометрических параметров: координаты точек, коэффициенты уравнений линий, в том числе высоких порядков, информацию о типе линий и другие неграфические атрибуты.

Достоинством внутреннего формата системы КОМПАС являются компактность хранения, т.е. сравнительно малый объем файла чертежа и отсутствие дублирования данных.

Недостатками формата является то, что все данные о чертеже хранятся в бинарном виде, что неудобно для расшифровки без спецификации.

## 2.2 Формат хранения .dxf

DXF (англ. Drawing eXchange Format) — открытый формат файлов для обмена графической информацией между приложениями САПР. Был создан фирмой Autodesk для системы AutoCAD. Поддерживается практически всеми CAD-системами на платформе PC.

К числу достоинств данного формата можно отнести хранение ГГИ в текстовом виде, открытую спецификацию, что облегчает чтение данных прикладной программой, распространенность формата среди различных САПР, последовательность хранения данных внутри файла в виде записей фиксированного формата и объема (80 байт).

К недостаткам данного формата можно отнести закрытую спецификацию хранения трехмерных данных, постоянно расширяющуюся спецификацию хранения двухмерных данных. Из особенностей внутренней структуры хранения данных, минусом формата является то, что в каждой записи содержится один символ или число, что дает разреженность данных и, как следствие, большой объем выходного файла.

Таблица 1. Числовые соотношения средних длин файлов систем

Система	КОМПАС	AUTOCAD	CATIA2D	CATIA3D	КОМПАС
КОМПАС	⊕	0.45	0.31	0.125	⊕
AUTOCAD	2.22	⊕	0.684	0.278	2.22
CATIA2D	3.226	1.46	⊕	0.407	3.226
CATIA3D	8	3.6	2.457	⊕	8

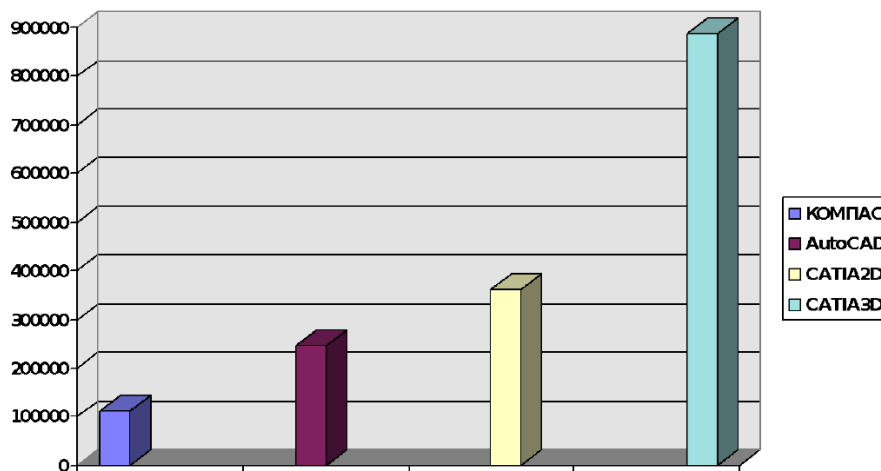


Рис. 15. Диаграмма соотношения средних длин файлов систем

В кандидатской диссертации Г.Б.Широкого «Средства архивизации геометро-графической информации в процессе автоматизированного проектирования на крупном предприятии» [8] описаны некоторые существующие форматы хранения ГГИ, а так же проведен анализ объема выходного файла у различных систем для одинакового чертежа. Часть данных приведена на рис. 3 и в таблице 1, из которых следует вывод о разреженности внутренней структуры хранения данных у систем Autocad, Catia2d и Catia3d по сравнению с системой Компас, что увеличивает время доступа к данным и их обработку.

Рассмотрим подробнее процедуры чтения и записи данных формата .cdw.

### 2.3 Чтение и запись формата .cdw

Чтение бинарного формата .cdw осуществляется из прикладного приложения путем взаимодействия с САПР «Компас» через предлагаемое компанией «Ас-кон» API и СОМ-объекты, обеспечивающие связь между двумя запущенными приложениями. Таким образом, чтение данных из файла формата .cdw возможно произвести лишь средствами САПР «Компас», когда нужный файл открывается внутри САПР и интересующие данные при помощи связующего СОМ-объекта передаются родительскому процессу прикладного приложения.

Вместе с САПР «Компас» поставляется SDK с примерами использования API, где так же описана процедура чтения данных из открытого документа при помощи обхода всех имеющихся слоев, видов и объектов внутри чертежа.

Запись формата .cdw так же осуществляется при помощи API «Компас» и средствами самой САПР.

Перейдем к исследованию возможности предобработки ГГИ и исправлению численных ошибок проектирования.



На данный момент не будем рассматривать ошибки проектирования, относящиеся к процессу производства объекта класса «деталь», такие как: несовпадение чертежей со спецификациями; различное положение одного объекта, находящегося на разных листах проектного решения (например, вентиляционной шахты) и т.д.; а остановимся лишь на ошибках, лежащих в одном поле чертежа. В основном такие ошибки возникают из-за невнимательности или же некомпетентности исполнителя (например, подобных ошибок полно в чертежах студентов). На рис. 4 приведен пример подобной работы: слева – вариант необходимыми размерными числами; справа – тот же чертеж, только размерные числа проставлены самой САПР.

Исходя из ГОСТ 2.307-2011 «Нанесение размеров и предельных отклонений», пункт 4.1: «Основанием для определения величины изображенного изделия и его элементов служат размерные числа, указанные в графическом документе.» при изготовлении детали нам следует полагаться на числа, стоящие над размерной линией.

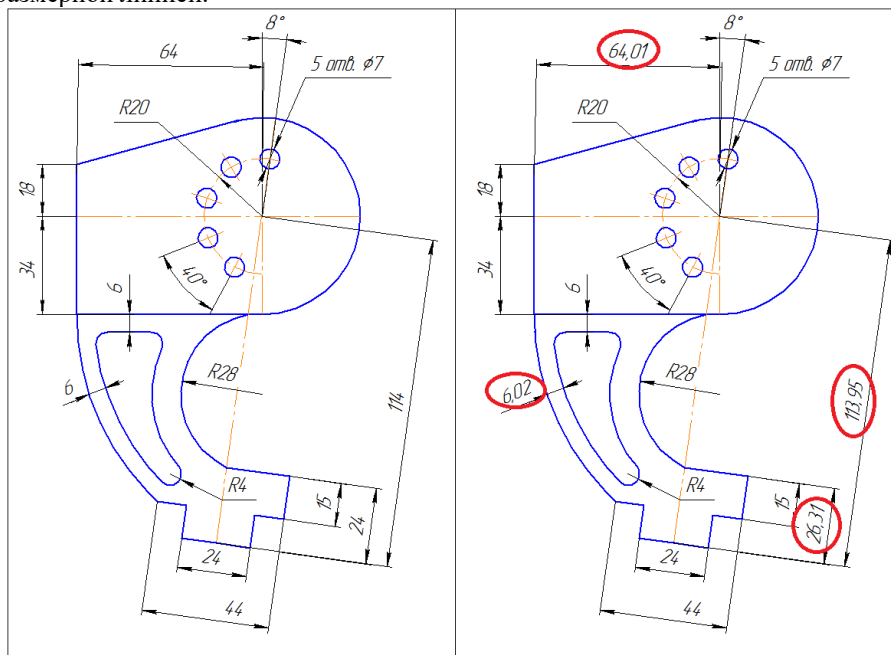


Рис. 16. Пример чертежа с различающимися длинами отрезков и размерным числом

Тем не менее, при автоматическом получении проекционных изображений с трехмерной модели, например, может возникнуть ситуация, когда величина объекта имеет размер, не реализуемый на практике (например, размерное число с точностью до тысячных долей миллиметра при строительстве панельного дома).

На рис. 5 можно заметить несколько размеров, труднореализуемых на практике в связи с очень высокой требуемой точностью исполнения.

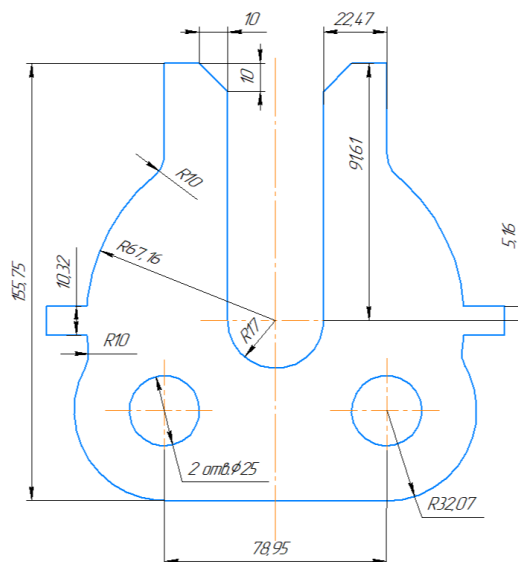


Рис. 17. Пример чертежа с размерами, требующими высокой точности при изготовлении

Исходя из ГОСТ 2.052-2006 «ЭЛЕКТРОННАЯ МОДЕЛЬ ИЗДЕЛИЯ. Общие положения», пункт 4.7: «Электронный конструкторский документ, выполненный в виде модели, должен соответствовать следующим основным требованиям:

- атрибуты (модели), обозначения и указания, приведенные в модели, должны быть необходимыми и достаточными для указанной цели выпуска (например, изготовления изделия или построения чертежа в бумажной и/или электронной форме);
- все значения размеров должны получаться из модели».

Таким образом, при автоматическом чтении прикладной программой, лишь использующей имеющуюся ГГИ, размерные числа можно не учитывать, так как при соблюдении элементарной проектной культуры и, собственно, цифровом проектировании, размерное число по умолчанию должно быть равным величине объекта, для которого этот размер нанесен. Исходя из этого, можно оперировать лишь геометрическими примитивами, обращаясь к условным обозначениям только в спорных ситуациях.

Возникает необходимость в предобработке двумерной ГГИ перед её непосредственным использованием путем выявления и обнаружения вышеописанных ошибок проектирования.

### 3 Алгоритмы и методы получения геометро-графической информации

Для реализации операций чтения и записи ГГИ на примере API САПР Компас-График [9] в прикладной программе необходимо проделать следующие действия (блок-схема на рис. 6):

1. Создать COM-объект для взаимодействия между САПР и прикладной программой.
2. Открыть необходимый графический документ для считывания информации.
3. В активном документе создать итератор для получения доступа ко всем объектам в документе.
4. При помощи итератора обойти все объекты в графическом документе.

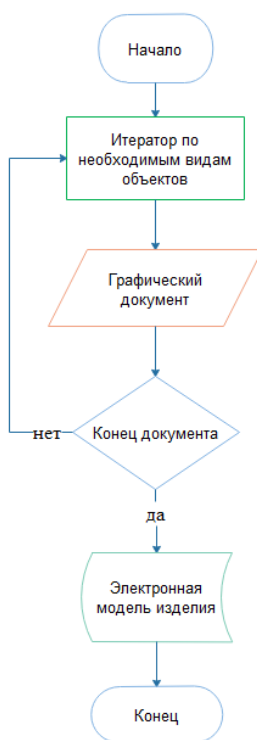


Рис. 18. Принцип работы прикладной программы при чтении ГГИ

### 3.1 Создание COM-объекта для взаимодействия с САПР Компас-График

Главным интерфейсом API системы КОМПАС является KompasObject. Получить указатель на этот интерфейс (если быть точным, на интерфейс приложения API 5) можно с помощью экспортной функции CreateKompasObject().

Методы этого интерфейса реализуют наиболее общие функции работы с документами системы, системными настройками, файлами, а также дают возможность получить указатели на другие интерфейсы (интерфейсы динамического массива, работы с математическими функциями, библиотек моделей или фрагментов и различных структур параметров определенного типа).

### 3.2 Открытие графического документа

Другой важный интерфейс API 5 – интерфейс документа модели ksDocument2D. Получить его можно с помощью функций интерфейса KompasObject:

- ActiveDocument2D – для уже существующего и активного в данный момент документа;
- Document2D – если вы планируете создавать новый двумерный документ.

Для открытия документа с жесткого диска используется метод ksDocument2D::ksOpenDocument.

### 3.3 Создание итератора

Синтаксис: reference ksIterator::CreateIterator(int searchType, reference parent).

Передвижение по модели документа (навигация) производится в соответствии с условиями, заданными в специальном блоке параметров (блок-схема на рис. 7).

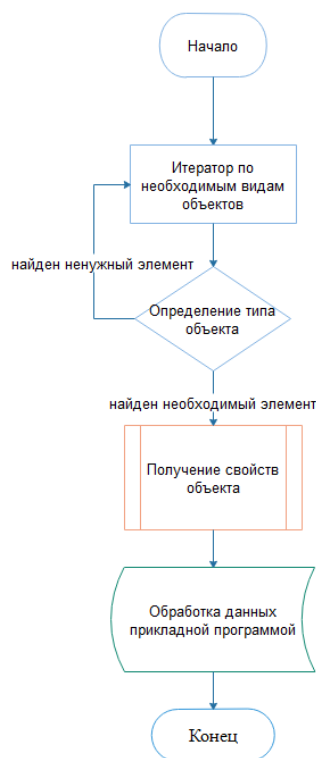


Рис. 19. Алгоритм работы итератора на модели данных

Он содержит тип движения, определяющий режим перемещения (например, по видам, слоям, всем объектам, объектам заданного типа и т.д.) и указатель комплексного объекта (макроэлемента, слоя, группы) при перемещении по составляющим его объектам.

Итератор привязан к конкретному режиму графического редактора (например, документу, виду), поэтому нет возможности использовать один и тот же итератор для навигации в разных видах, штампах и т.п. Итератор сохраняет свое действия до окончания сеанса работы с библиотекой.

### 3.4 Перемещение внутри графического документа

Синтаксис: `reference ksIterator::MoveIterator (reference iterator,unsigned char type)`.

Данная функция перемещает текущий указатель итератора (блока параметров навигации) `iterator` на очередной объект модели. Режим позиционирования (тип объектов, участвующих в поиске) определяются при создании итератора.

При позиционировании на именованную группу она автоматически становится текущей (рабочей). К ней будут относиться операции добавления и ис-

ключения объектов (при исключении всех объектов из именованной группы она автоматически удаляется).

### 3.5 Чтение свойств объекта

Из объекта, на который указывает итератор, можно получить список свойств текущего объекта. Полный список видов объектов, встречающихся в графических документах и поддерживаемых Компас-3D, можно прочитать в справке по SDK Компас-3D.

### 3.6 Запись во внутренний формат базы данных

После получения и обработки всей ГГИ прикладной программой, производится запись в базу данных.

### 3.7 Закрытие документа

По окончании работы с документом вызывается функция `int ksDocument2D::CloseDocument (reference sheet)`, где `sheet` – указатель на закрываемый документ.

По окончании работы с САПР Компас-График вызывается функция `void KompasObject::Quit()`, закрывающая COM-соединение и саму САПР.

## 4 Разработанный алгоритм формирования трехмерной каркасной модели на основе двумерных проекционных изображений

Первый этап формирования трехмерной каркасной модели [10] – получение двумерной ГГИ прикладной программой. В основном получаемая двумерная информация имеет систему координат, не связанную с будущей трехмерной моделью. Чтобы иметь возможность построения каркасной модели, необходимо преобразовать имеющуюся систему координат проекционных изображений в эпюр Монжа, содержащий информацию уже о трех  $(x, y, z)$  координатах. Укрупненный алгоритм данного преобразования координат представлен на рис. 8.

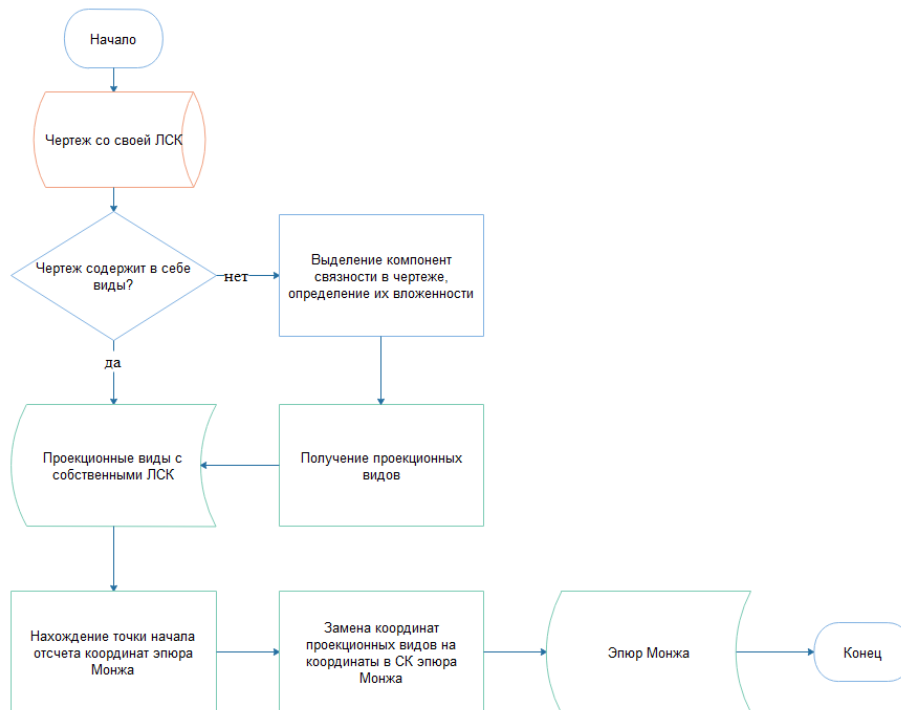


Рис. 8. Блок-схема алгоритма получения из внешней ГГИ эпюра Монжа.

Вторым этапом получения трехмерной каркасной модели является построение точечной трехмерной модели. Точечная трехмерная модель представляет собой облако точек, никак не связанных между собой. Каждая вершина данной модели получается следующим образом:

1. На одном из проекционных изображений возьмем какую-либо точку. Например, возьмем точку на виде спереди, получив таким образом сведения о координатах  $(x1, z1)$  будущей трехмерной вершины.
2. Найдём все пары подходящих по одной из координат ( $x1$  или  $z1$  в нашем примере) вершин на следующем проекционном изображении. Например, на виде сверху есть две подходящие вершины –  $(x1, y1)$  и  $(x1, y2)$ .
3. На третьем проекционном изображении попытаемся найти точки с координатами из двух предыдущих шагов. В нашем примере это вершины с координатами  $(y1, z1)$  и  $(y2, z1)$ .
4. Если данные вершины существуют на проекционном изображении, то данные трехмерные точки есть и в трехмерной точечной модели. В примере это точки с координатами  $(x1, y1, z1)$  и  $(x1, y2, z1)$ .

На третьем этапе путем добавления ребер в точечную модель получается искомая каркасная трехмерная модель. Ребра же добавляются исходя из следую-

шего утверждения: если для точек  $(x_1, y_1, z_1)$ ,  $(x_2, y_2, z_2)$  верно, что на соответствующих проекционных изображениях существуют ребра  $(x_1, z_1) - (x_2, z_2)$ ;  $(x_1, y_1) - (x_2, y_2)$ ;  $(y_1, z_1) - (y_2, z_2)$  или соответствующие точки совпадают, то, скорее всего, существует ребро  $(x_1, y_1, z_1) - (x_2, y_2, z_2)$ .

При наличии на входных проекционных изображениях ребер второго порядка алгоритм получения трехмерной каркасной модели фактически не изменяется. Отличия состоят в том, что имеющееся ребро второго порядка дополнительно разбивается несколькими характерными точками. Количество точек зависит от необходимой точности отображения, минимальное количество таких точек для окружности – 4, исходя из количества пересечений двумя секущими плоскостями, каждая из которых параллельна соответствующей координатной плоскости и проходит через центр окружности. Далее находятся ребра, являющиеся отрезками, с вершинами в этих характерных точках, и восстанавливаются трехмерные ребра с аппроксимацией кубическими сплайнами. После восстановления эти точки и ребра не учитываются при восстановлении остальной модели.

## 5 Временные характеристики разработанного алгоритма

Первый вид эксперимента – сравнение скорости построения трехмерной модели путем ручного ввода ГГИ в САПР и автоматизированного построения трехмерной каркасной модели. В среднем на создание простой трехмерной модели тратится от 2 до 5 минут. Экспериментальная программная реализация показывает результат уже через 0.5-3 секунды после запуска процесса чтения двумерной информации на простых трехмерных моделях.

Вышеизложенные результаты эксперимента позволяют говорить об ускорении процесса получения трехмерной модели по проекционным изображениям в 60-200 раз в зависимости от сложности восстанавливаемой модели и мощности центрального процессора.

Второй вид эксперимента – сравнение времени, которое тратится алгоритмом на выполнение своих различных составных частей: реализацию операций чтения чертежа, получения точечной трехмерной модели и её преобразования в каркасную.

Первая часть исследования проводилась на несложных трехмерных моделях – до 100 вершин. Эксперимент показал, что самый большой процент времени тратится на получение ГГИ из внешнего источника – до 99%. После выполнения операции чтения вся двумерная геометрическая информация помещается в виртуальное адресное пространство разработанного программного комплекса и соответственно на доступ к ней тратится существенно меньше времени. На экспериментальных чертежах время восстановления точечной и каркасной моделей не превышало 1 мс.

Вторая часть исследования включала в себя более сложные модели – более 100 вершин. Данные, полученные в ходе исследования, приведены в таблице 2.



Таблица 2. Временные характеристики разработанного алгоритма

Количество вершин	Общее время восстановления, сек.	Время чтения двумерных данных, сек.	Время восстановления точечной модели, сек.	Время получения каркасной модели, сек.
844	29.859	28.125	0.5	1.234
2440	206.547	185.140	1.282	20.125
15616	424.969	95.156	14.78	315.735

Из проведенного исследования [11] можно сделать следующие выводы:

1. на моделях до 1000 вершин около 90% времени занимает чтение геометрографической информации из внешнего источника и лишь 10% - собственно формирование точечной и каркасной моделей;
2. экспериментальные данные показывают сложность формирования точечной трехмерной модели  $O(n)$ , где  $n$  – количество вершин в обрабатываемой модели;
3. экспериментальные данные показывают сложность алгоритма формирования каркасной трехмерной модели  $O(k * n)$ , где  $n$  – количество вершин в обрабатываемой модели, -среднее количество ребер, исходящих из одной вершины.

## 6 Заключение

1. Проведен анализ публикаций по существующим методам и алгоритмам преобразования двумерных геометрических моделей в трехмерные геометрические модели. Выявлено отсутствие во всех системах геометрического моделирования и компьютерной графики оператора преобразования проекционных изображений в трехмерную модель.
2. Разработан алгоритм преобразования двумерной геометрической модели в трехмерную каркасную модель, содержащую ребра, описываемые уравнениями до второго порядка включительно, в котором сняты ограничения на количество геометрических элементов, как во входных данных, так и в синтезируемой модели.
3. Выполнена программная реализация разработанного алгоритма преобразования двумерной геометрической модели в трехмерную каркасную модель на языке программирования C++, работающая в современных версиях OS Windows и позволяющая установить временные зависимости составных частей алгоритма.
4. Экспериментальная программная реализация позволяет ускорить процесс создания трехмерной каркасной модели в 60-200 раз (в зависимости от сложности модели) по сравнению с созданием этой модели ручным способом в одной из САПР.

## Литература

1. Тани, Х.И. Ввод чертежей и синтез пространственного образа тел в ЦВМ: Автореф. дис. канд. техн. наук / Х.И. Тани. Минск, 1968. -23 с.
2. Щеколдин, В.А. Построение на ЭВМ пространственного описания многогранного тела по описаниям его проекций / В.А. Щеколдин // Вычислительная техника в машиностроении: Сб. ст. — Минск. ИТК АН БССР, 1968.
3. Полозов В.С. Моделирование и синтез операторов геометрического расчета и машинной графики в системах автоматизированного проектирования и автоматизации технологической подготовки производства. // Автореф. дисс. докт. техн. наук. М., МАИ, 1983.
4. Idesawa M. (1973). A system to generate a solid figure from three views. Bull JSME, 16, 216-225.
5. Балабан, О.М. Разработка метода синтеза 3D моделей изделий по информации, регламентированной ЕСКД. // Автореф. дисс. канд. техн. наук, Н.Новгород. 1995.
6. Тюрина, В.А. Разработка методов преобразований каркасной модели в задаче синтеза образа 3D-объекта по его проекциям // Автореф. дис. . канд. техн. наук / В.А. Тюрина. Н.Новгород, 2003
7. Зудин А.А., Ротков С.И. Графические диалоговые средства построения конечно-элементных сеток в системе «КИТЕЖ» // В сб. Прикладные проблемы прочности и пластичности. Численное моделирование физико-механических процессов. Горький, 1988.
8. Широкий Г.Б. Средства архивизации геометро-графической информации в процессе автоматизированного проектирования на крупном предприятии. // Диссертация кандидата технических наук. Нижний Новгород, 1999, 126 стр.
9. Роменский, С.А. Передача геометро-графической информации из системы автоматизированного проектирования в прикладную программу на примере САПР “Компас-График” // Системы проектирования, технологической подготовки производства и управления этапами жизненного цикла промышленного продукта (CAD/CAM/PDM-2015): сб. тр.
10. Роменский, С.А. Синтез 3D модели объекта по изображениям на поле чертежа / С.А. Роменский, С.И. Ротков, М.М. Смычѣк, В.А. Тюрина // GraphiCon 2016: труды 26-й Междунар. конф. по компьютерной графике и машинному зрению (Нижний Новгород, 19-23 сентября 2016) / Нижегородский архитектурно-строительный университет. – Нижний Новгород, 2016. с. 272-275.
11. Мошкова, Т.В. Исследование временных характеристик алгоритма восстановления каркасной модели по проекционным изображениям / Т.В. Мошкова, С.А. Роменский, С.И. Ротков, В.А. Тюрина // GraphiCon 2018: труды 28-й Междунар. конф. по компьютерной графике и машинному зрению Томск, 24–27 сент., 2018 г.) / Нац. исслед. Том. политех. ун-т. – Томск, 2018. с. 369-371

## Программно-аппаратная реализация конструктивных геометрических моделей

Д.В. Волошинов<sup>1</sup>, К.Н. Соломонов<sup>2</sup>

<sup>1</sup> СПбГУТ им. проф. М.А. Бонч-Бруевича, 193232, Санкт-Петербург, Россия  
denis.voloshinov@yandex.ru

<sup>2</sup> Филиал Ростовского государственного университета путей сообщения, 105064, Россия  
konssol@list.ru

**Аннотация.** Статья посвящена рассмотрению ряда вопросов аппаратно-программной реализации конструктивных геометрических моделей. Богатый арсенал теоретических исследований в области конструктивной геометрии длительное время не находил должного применения в виду отсутствия инструментов воплощения таких моделей средствами вычислительной техники. Разработка и совершенствование системы геометрического моделирования Симплекс, в которой любая геометрическая конструкция рассматривается как преобразователь информации, представленной сигналами геометрической природы, открыла возможность применения достижений геометрической науки в вычислительных приложениях, а также разработки аппаратных средств, реализующих геометрические методы вычислений и предоставляющих новый графический интерфейс. Развиваемая авторами концепция направлена на создание специализированных ускорителей геометрических преобразований.

**Ключевые слова:** научная визуализация; конструктивное геометрическое моделирование; геометрический эксперимент, встраиваемые системы, графический интерфейс.

### 1 Введение

Конструктивная геометрия является одним из разделов математики, который в недалеком прошлом находил широкое применение в практике инженерного проектирования, обработке и представления результатов физических экспериментов, наглядного отображения многопараметрических процессов. За свою историю синтетическая геометрия накопила богатейший арсенал средств моделирования, однако, в условиях современной информатизации и цифровизации он остается практически невостребованным из-за необходимости представления геометрической информации в аналитическом виде. Даже краткий перечень работ авторов прошлого и современности демонстрирует существование этого противоречия [1-3, 5, 14-21].

В развитие конструктивной геометрии внесли вклад многие ученые, среди которых особое место занимают работы профессора К.И. Валькова [4], предло-

жившего рассматривать любую геометрическую конструкцию в контексте действия кибернетического устройства, обрабатывающую информацию особого рода – геометрическую. Такие устройства получили название геометрических машин, производящих определенную работу по переработке информации. К сожалению, во время возникновения этих идей уровень развития вычислительной техники не только не позволял соединить возможности геометрического метода с возникающими информационными технологиями, но имел обратный эффект, в результате которого геометрия стала противопоставляться достижениям в области кибернетики. Попытка ассоциировать геометрические конструкции с механическими системами, несмотря на принципиальную возможность такого подхода, не могла по естественным причинам конкурировать с новой развивающейся технологией, в результате чего геометрическая наука была отнесена к категории устаревающих и не имеющих перспектив в сравнении с аналитической ветвью математики. В результате сложившейся тенденции интерес исследователей к геометрическому знанию стал угасать, а невозможность практического приложения результатов исследований к новым информационным реалиям привела к сорокалетнему застою в геометрических изысканиях.

Переосмысление концепции геометрических машин в контексте новых информационных парадигм позволило по-новому подойти к решению застарелой проблемы, ликвидировать искусственное противостояние синтетической геометрии и информатики. Результатом проведенных исследований стало создание системы конструктивного геометрического моделирования Симплекс [8, 10], позволившей не только дать новый толчок к развитию геометрической науки, но и определить способы практической реализации ее результатов как новых информационных технологий: в виде программ, систем и их интерфейса, а также перспективных аппаратных реализаций. Результаты, которые удалось достичь авторам в этом направлении, представлены в работах [6, 7 9, 13]. В рамках данной статьи обсуждаются некоторые новые достижения в обозначенной области исследований, касающиеся программно-аппаратной реализации конструктивных геометрических моделей на электронных устройствах с микроконтроллерным управлением и в перспективе на устройствах ПЛИС.

## **2 Механизмы расширения функциональных возможностей системы Симплекс**

Исследования, проводимые в области конструктивного геометрического моделирования и осуществляемые с помощью системы Симплекс, приводят к необходимости разработки новых все более сложных функций, компонентами которых являются ранее разработанные объекты и функции. Для обеспечения расширения функционального состава в системе в ней предусмотрены два механизма. Первый (внутренний) позволяет создавать процедуры-функции и иерархии производных объектов с использованием визуально-графических средств самой системы. Таким образом пользователь получает возможность неограни-

ченно расширять функциональный состав и, что наиболее важно, исследовать и отлаживать геометрические алгоритмы, получая полный комплекс информационных средств для влияния на значения составляющих их объектов и наблюдения за получаемыми данными как в числовой, так и в графической форме. Для обеспечения действия этого механизма система осуществляет интерпретацию команд и данных, что непременно сказывается на скорости выполнения вычислений и на производительности системы в целом. По мере отладки модели необходимость интерпретационного исполнения алгоритмов отпадает, вследствие чего более рациональным становится подход, подразумевающий трансляцию алгоритма до уровня машинного кода. Вторым предусмотренный в системе механизм реализует ретрансляцию внутрисистемного представления геометрических алгоритмов на язык Pascal системы Delphi, после чего полученный код транслируется, компонуется и становится одной из функций библиотеки системы, к которой обеспечивается типовой доступ через базовый системный интерфейс.

В работе [11] была подробно рассмотрена и обоснованы специфика организации системы конструктивного геометрического моделирования, приведены правила ее организации, соблюдение которых в наибольшей степени способствует удобному и естественному стилю деятельности разработчика конструкции геометрических машин. Особенно важными из них являются следующие требования: отказ от текстового ввода команд, соблюдения их порядка и «прозрачная» организация множественного процедурного вызова. Необходимость соблюдения этих правил определила структуру внутренних механизмов системы и способы организации функционального состава системы. Так, например, на нижнем уровне находится большинство функций, выполняющих операции над одним или несколькими семантически различными объектами геометрической модели, представленными единичными экземплярами. Пример такой функции, разработанной для решения задачи Аполлония и представленный текстом на языке Pascal, имеет префикс EExec (листинг 1). Ко второму уровню относятся функции, имеющие префикс XExec и отвечающие за организацию выходных списочных переменных, формируемых из единичных решений путем многократного обращения. В них также реализованы механизмы множественности [12], определяющие как стандартные, так и пользовательские методы согласования параметров для организации многократного процедурного вызова. Функции этого уровня имеют префикс Exec (листинг 1) и, несмотря на возникающий в этих механизмах прецедент параллелизма вычислений, выполняют это действие, как циклическую последовательность операций.

**Листинг 3.** Фрагмент программы на языке Pascal, сгенерированный системой Симплекс для моделирования геометрической конструкции задачи Аполлония

```
function EExecAppolonius(in_prm1,in_prm2,in_prm3:
TObj; var out_prm1,out_prm2: TObj; Att_1,Att_2: TAtt;
Sg1,Sg2,Sg3: integer; OW1,OW2: pointer): boolean;
var d1, d2, d3, d4, d5, d6, d7, d8, d9, o1, o2, o3,
```

```

o4, o5, o6, o7, p1, p10, p11, p12, p13, p14, p15, p16,
p17, p18, p19, p2, p20, p21, p3, p4, p5, p6, p7, p8,
p9: TObj;
    Success: boolean;
begin
    Mass-
NIL ([@d1,@d2,@d3,@d4,@d5,@d6,@d7,@d8,@d9,@o1,@o2,@o3,@o
4,@o5,@o6,@o7,@p1,@p10,@p11,@p12,@p13,@p14,@p15,@p16,@p
17,@p18,@p19,@p2,@p20,@p21,@p3,@p4,@p5,@p6,@p7,@p8,@p9]
);

    Result:=TRUE;
    d1:=in_prm1.CreateCopy(NIL);
    d1.OAtt:=in_prm1.OAtt;
    d2:=in_prm2.CreateCopy(NIL);
    d2.OAtt:=in_prm2.OAtt;
    d3:=in_prm3.CreateCopy(NIL);
    d3.OAtt:=in_prm3.OAtt;

    repeat
        Success:=FALSE;
        if not Assigned(o1) and not Assigned(p1) and
not Assigned(p2) then
            Suc-
cess:=EExecO3(d1,d2,o1,p1,p2,Att5,Att0,Att0,Sg1,Sg2,NIL
,NIL,NIL) or Success;
        if not Assigned(o2) and not Assigned(p3) and not As-
signed(p4) then
            Suc-
cess:=EExecO3(d2,d1,o2,p3,p4,Att5,Att0,Att0,Sg2,Sg1,NIL
,NIL,NIL) or Success;
        if not Assigned(o3) and not Assigned(p6) and not As-
signed(p7) then
            Success:=EExecO3(d1,d3,o3,p6,p7,Att5,Att0,Att0 Sg1
Sg3,NIL,NIL,NIL) or Success;
        if not Assigned(o4) and not Assigned(p8) and not As-
signed(p9) then
            Success:=EExecO3(d3,d1,o4,p8,p9,Att5,Att0,Att0,Sg3
Sg1,NIL,NIL,NIL) or Success;
        if not Assigned(o5) and not Assigned(p11) and not As-
signed(p12) then
            Success:=EExecO3(d2,d3,o5,p11,p12,Att5,Att0,Att0 Sg2
Sg3,NIL,NIL,NIL) or Success;
        if not Assigned(o6) and not Assigned(p13) and not As-
signed(p14) then
            Success:=EExecO3(d3,d2,o6,p13,p14,Att5,Att0,Att0 Sg3
Sg2,NIL,NIL,NIL) or Success;
        if not Assigned(d4) then

```

```

    Suc-
    cess:=EExecOK001 (d1, d2, d3, d4, Att5, Sg1, Sg2, Sg3, NIL) or
    Success;
    if not Assigned(p5) then
        Success:=EExecP2 (o1, o2, p5, Att5, 1, 1, NIL) or Success;
    if not Assigned(p10) then
        Success:=EExecP2 (o3, o4, p10, Att5, 1, 1, NIL) or Success;
    if not Assigned(p15) then
        Success:=EExecP2 (o5, o6, p15, Att5, 1, 1, NIL) or Success;
    if not Assigned(o7) then
        Success:=EExecO000 (p5, p10, p15, o7, Att5, 1, 1, 1, NIL) or
    Success;
    if not Assigned(d5) then
        Success:=EExecOK001 (d1, d4, o7, d5, Att5, Sg1, 1, 1, NIL) or
    Success;
    if not Assigned(d6) then
        Success:=EExecOK001 (d2, d4, o7, d6, Att5, Sg2, 1, 1, NIL) or
    Success;
    if not Assigned(d7) then
        Success:=EExecOK001 (d3, d4, o7, d7, Att5, Sg3, 1, 1, NIL) or
    Success;
    if not Assigned(p16) and not Assigned(p17) then
        Suc-
    cess:=EExecP3 (d1, d5, p16, p17, Att0, Att0, Sg1, 1, NIL, NIL) or
    Success;
    if not Assigned(p18) and not Assigned(p19) then
        Suc-
    cess:=EExecP3 (d2, d6, p18, p19, Att0, Att0, Sg2, 1, NIL, NIL) or
    Success;
    if not Assigned(p20) and not Assigned(p21) then
        Suc-
    cess:=EExecP3 (d3, d7, p20, p21, Att0, Att0, Sg3, 1, NIL, NIL) or
    Success;
    if not Assigned(d8) then
        Success:=EExecD4 (p17, p18, p20, d8, Att5, 1, 1, 1, NIL) or
    Success;
    if not Assigned(d9) then
        Success:=EExecD4 (p16, p19, p13, d9, Att5, 1, 1, 1, NIL) or
    Success;
        until not Success;
        out_prm1:=d9.CreateCopy (OW1);
        out_prm1.OAtt:=Att_1;
        out_prm2:=d8.CreateCopy (OW2);
        out_prm2.OAtt:=Att_2;
    Mass-
    Free ([d1, d2, d3, d4, d5, d6, d7, d8, d9, o1, o2, o3, o4, o5, o6, o7, p
    1, p10, p11, p12, p13, p14, p15, p16, p17, p18, p19, p2, p20, p21, p3
    , p4, p5, p6, p7, p8, p9]);

```

```

end; // EExecAppolonius

function XExecAppolonius (FM: string;
_W1,_W2,LC1,LC2,LC3: TNewList; Att1,Att2: TAtt): boolean;
var
  I,J,s_N, I1,I2,I3, Sg1,Sg2,Sg3: integer;
  CI,CO,Seq: TNewList;
  s_Prml,s_Prml2,s_Prml3 : TObj;
label lb;

procedure CALC;
var
  s_Out1, s_Out2: TObj;
begin
  Re-
  sult:=EExecABCD(s_Prml,s_Prml2,s_Prml3,s_Out1,s_Out2,Att1
,Att2,Sg1,Sg2,Sg3,_W1,_W2);
  if Assigned(s_Out1) then
  begin
    s_Out1.Parents.Add(s_Prml);
  s_Out1.Parents.Add(s_Prml2);
  s_Out1.Parents.Add(s_Prml3);
  _W1.Add(s_Out1);
  case Att1.View of
    view_AND: Tobj(_W1[_W1.Count-
1]).View:=s_Prml.View and s_Prml2.View and s_Prml3.View;
    view_OR: Tobj(_W1[_W1.Count-
1]).View:=s_Prml.View or s_Prml2.View or s_Prml3.View;
  end; // case
  end;
  if Assigned(s_Out2) then
  begin
    s_Out2.Parents.Add(s_Prml);
  s_Out2.Parents.Add(s_Prml2);
  s_Out2.Parents.Add(s_Prml3);
  _W2.Add(s_Out2);
  case Att2.View of
    view_AND: Tobj(_W2[_W2.Count-
1]).View:=s_Prml.View and s_Prml2.View and s_Prml3.View;
    view_OR: Tobj(_W2[_W2.Count-
1]).View:=s_Prml.View or s_Prml2.View or s_Prml3.View;
  end; // case
  end;
end; // CALC

var
Lim1,Lim2,Lim3,Lim4: integer;

```



```

begin // XExecAppolonius
  if FM = '0' then
  begin
    s_N:=Amin1(Amin1(LC1.Count-1,LC2.Count-
1),LC3.Count-1);
    for I:=0 to s_N do
    begin
      s_Prml:=GetLC(LC1,I,SG1,Lim1);
      s_Prml2:=GetLC(LC2,I,SG2,Lim2);
      s_Prml3:=GetLC(LC3,I,SG3,Lim3);
      CALC;
    end;
    goto lb;
  end; // FM=0

  if FM = '1' then
  begin
    for I1:=0 to LC1.Count-1 do
    for I2:=0 to LC2.Count-1 do
    for I3:=0 to LC3.Count-1 do
    begin
      s_Prml:=LC1[I1]; s_Prml2:=LC2[I2]; s_Prml3:=LC3[I3];
      s_Prml:=GetLC(LC1,I1,SG1,Lim1);
      s_Prml2:=GetLC(LC2,I2,SG2,Lim2);
      s_Prml3:=GetLC(LC3,I3,SG3,Lim3);
      CALC;
    end;
    goto lb;
  end; // FM=1

  CI:=TNewList.Create(NIL);
  CI.Add(LC1); CI.Add(LC2); CI.Add(LC3);
  SEQ:=TNewList.Create(NIL);
  Sequensor(FM,CI,CO,SEQ);
  for I:=0 to CO.Count-1 do
  begin
    s_Prml:=TNewList(CO[I])[integer(SEQ[0]^)];
    s_Prml2:=TNewList(CO[I])[integer(SEQ[1]^)];
    s_Prml3:=TNewList(CO[I])[integer(SEQ[2]^)];
    if zn_Minus in
TNewList(CO[I]).Chars[integer(SEQ[0]^)] then Sg1:=-1
else Sg1:=1;
    if zn_Minus in
TNewList(CO[I]).Chars[integer(SEQ[1]^)] then Sg2:=-1
else Sg2:=1;
    if zn_Minus in
TNewList(CO[I]).Chars[integer(SEQ[2]^)] then Sg3:=-1
else Sg3:=1;
  end;
end;

```

```

        CALC;
        TNewList(CO[I]).DelDestroy;
    end;
    CO.DelDestroy;
    CI.DelDestroy;
    for I:=0 to Seq.Count-1 do
FreeMem(SEQ[I],SizeOf(integer));
        SEQ.DelDestroy;

    lb:
end; // XAppolonius

function ExecAppolonius (var PTS: TStroka): boolean;
var
    All,R: boolean;    I: integer;  WL1,WL2:
TNewList;_W1,_W2: TNewList; Att1,Att2: TAtt;
    LC1,LC2,LC3: TNewList; LCR1,LCR2,LCR3: TNewList;
    PN1,PN2: TNames;

begin // ExecAppolonius
    All:=TRUE;
    Result:=FALSE;
    WL1:=PTS.ULeft[0];  WL2:=PTS.ULeft[1];
    _W1:=TPointer(WL1[0]).PN.OList;
    _W2:=TPointer(WL2[0]).PN.OList;
    Att1:=TPointer(WL1[0]).Att;
    Att2:=TPointer(WL2[0]).Att;
    Pn1:=TPointer(WL1[0]).Pn;
    All:=MakeAtt(PN1.AList,Att1,PN1.FAtt) and All;
    Pn2:=TPointer(WL2[0]).Pn;
    All:=MakeAtt(PN2.AList,Att2,PN2.FAtt) and All;

    All:=TAlg(PTS.Alg).Present(PTS,0,['?'],LC1,LCR1) and
All;
    All:=TAlg(PTS.Alg).Present(PTS,1,['?'],LC2,LCR2) and
All;
    All:=TAlg(PTS.Alg).Present(PTS,2,['?'],LC3,LCR3) and
All;

    if (LC1.Count=0) or (LC2.Count=0) or (LC3.Count=0)
or not All then
    begin
        LC1.DelDestroy;  LC2.DelDestroy;  LC3.DelDestroy;
Exit;
    end;

    R:=XExecABCD(PTS.FM,_W1,_W2,LC1,LC2,LC3,Att1,Att2);
    PTS.Executed:=R;
end;

```

```
Result:=R;  
LC1.DelDestroy; LC2.DelDestroy; LC3.DelDestroy;  
  
Markus(_W1,Att1.View); Markus(_W2,Att2.View);  
end; // ExecAppolonius
```

Подсистема трансляции на язык Pascal позволяет автоматически получать сгенерированные пользователем функции обслуживающих функций всех трех уровней, обеспечивая тем самым возможность расширения самой системы Симплекс. Однако разработанная концепция позволяет расширить ее применение и в других целях. В частности, генерация только лишь функций с форматом префикса «EExec» позволяет создавать отторгаемые от системы Симплекс приложения для решения конкретных пользовательских задач без применения самой системы Симплекс. Этот же механизм в совокупности с разработкой соответствующих библиотек на языках JavaScript и MaxScript позволили получить скрипты, обеспечивающие интерактивную работоспособность геометрических схем в среде интернет-браузеров, а также разработать методику автоматизированного синтеза программ, функционирующих в среде системы 3DSMax фирмы Autodesk и обеспечивающих высококачественную визуализацию геометрических объектов в моделях не только трехмерного пространства, но и на трехмерных проекционных картинах пространств более высоких размерностей [7]. Ввиду того что языки JavaScript и MaxScript являются интерпретируемыми, время получения результатов в интернет-браузерах и в системе 3DSMax значительно больше, чем собственно в системе Симплекс, которая пополняет свой состав за счет трансляции кода. Однако, несмотря на неизбежные временные издержки, предложенный подход позволил качественно изменить ситуацию с системным распространением геометрических знаний и обеспечить их представление в доступной и понятной визуальной форме. Предлагаемая информационная технология позволяет рассматривать синтетическую геометрию как специализированный раздел информатики.

### 3 Аппаратная реализация конструктивных геометрических моделей

Развитие концепции геометрических машин как информационных преобразователей универсального назначения естественным образом согласуется с современными тенденциями развития автономных электронных устройств. Так, например, фирма Mikroelektronika (Сербия) специализируется на разработке и внедрении систем быстрого прототипирования современных электронных устройств, основными областями применения которых являются получение и обработка данных, автоматизация, локальная и мобильная связь, средства организации человеко-машинного интерфейса в распределенных и встраиваемых системах. Парадигма, которой придерживается фирма MikroElektronika, заключается в организации аппаратно-модульного принципа проектирования, в котором центральное место отводится специальным интегрированным отладочным

платам (таких как EasyMx PRO v7 for STM32, Fusion for ARM v8 и многие др.), функциональность которых может неограниченно расширяться дополнительными модулями и устройствами, получившими название Click, подключающихся к фирменной шине MikroBUS™. Единая концепция, в рамках которой удалось охватить широкую номенклатуру микроконтроллеров с различной архитектурой и аппаратной реализацией, позволила создать удобную развивающуюся аппаратно-программную платформу для проведения экспериментальных исследований как в традиционных для разработчиков электронной техники областях, так и в различных смежных направлениях

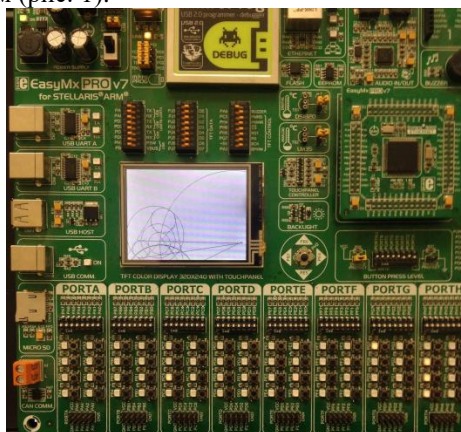
Программирование устройств системы осуществляется в среде быстрой подготовки программ, внешне напоминающей среду программирования Delphi, с опциональным использованием трех алгоритмических языков: MikroPascal, MikroC и MikroBasic, являющихся расширенными диалектами широко известных языков программирования общетехнического назначения Pascal, C и Basic. Общая функциональность этих языков расширяется за счет использования специализированных библиотек, функций и процедуры которых в большинстве своем направлены на организацию работы аппаратной части системы прототипирования и поддержки протоколов интерфейсов связи.

Следует отметить, что представленные фирмой MikroElektronika языки программирования не являются объектно-ориентированными, что, в известной степени, обусловлено отсутствием необходимости поддержки такой парадигмы программирования при разработке аппаратной части радиоэлектронных устройств. В свою очередь, создание графического интерфейса и иных прикладных систем без поддержки объектно-ориентированного способа проектирования сопряжено с большими трудностями.

Для развития концепции представления и обработки информации средствами геометрических машин, система прототипирования, разработанная фирмой MikroElektronika, представляет большой интерес, поскольку входные параметры геометрических моделей могут быть легко ассоциированы с сигналами различных физических датчиков и устройств. В свою очередь, это открывает перед геометрическим моделированием новые возможности приложения своих методов и превращает его в средство программирования, расширения функциональных возможностей электронных устройств, приборов и гаджетов, создания специализированного графического интерфейса таких устройств посредством плоских LCD экранов и манипуляторов.

Для проведения экспериментальных исследований в обозначенной области был разработан комплекс функций, запрограммированных на языке MikroPascal. В виду отсутствия в этом языке средств программирования классов и типизированных указателей, в библиотеке реализован метод интерпретации объектно-ориентированного принципа проектирования методами декларативного программирования на основе расширения структур полями, имитирующими записи о классах. Библиотека состоит из функций общего назначения, выполняющих такую интерпретацию и ряд вычислительных действий общего назначения, а также комплекса функций геометрического содержания, выполняющих операции с объектами геометрической природы на проективной плоскости с

учетом возможности работы с комплекснознаными данными объектов. Данный подход позволил в полной мере реализовать методы проекционного моделирования пространств произвольных размерностей и разработать их графическую интерпретацию под управлением устройств быстрого прототипирования микроконтроллерных систем (рис. 1).



**Рис. 20.** Результат функционирования геометрической модели на плате разработки EasyMx Pro v7 for Stellaris® Arm® фирмы MikroElektronika

Специально разработанная для этих целей подсистема трансляции проекта системы Симплекс на язык MikroPascal синтезирует программу, выполняющую согласованное обращение к функциям геометрической библиотеки в точном соответствии с алгоритмом функционирования геометрической модели. Полученный текст программы представляет собой текст основного модуля программы, размещаемый в трансляторе системы быстрого прототипирования, который транслируется, компонуется и загружается в аппаратную часть платы разработки микроконтроллерной системы (листинг 2). Дальнейшее функционирование полученной программы могут выполняться автономно и управляться интерфейсными средствами этой платы.

**Листинг 4.** Фрагмент программы на языке MikroPascal фирмы MikroElektronika, сгенерированный системой Симплекс для моделирования геометрической конструкции задачи Аполлония

```
var
Chisl1, Chisl2, Chisl3, Chisl4, Chisl5, Chisl6, Chisl7, Chisl8,
Chisl9: DWord;
var
d1, d2, d3, o1, p1, p2, o2, p3, p4, p5, o3, p6, p7, o4, p8, p9, p10, o5, p
11, p12, o6, p13, p14, p15, o7, d4, d5, p16, p17, d6, p18, p19, d7, p20
, p21, d8, d9: DWord;
```

```

const
    Att_d1: TAtt = (0, 'basic', 1, 1, 0, 0, 0);
    Att_d2: TAtt = (0, 'basic', 1, 1, 0, 0, 0);
    . . . . .
    Att_d9: TAtt = (5, 'basic', 1, 1, 0, 0, 0);
begin
    InitSimplex;
    IY:=0;
    MM_Init();
    TFT_Init_ILI9340_8bit(320, 240);
    TFT_Fill_Screen(cl_white);

    Chisl1:=TOChisl_Create(-53.19, 0);
    Chisl2:=TOChisl_Create(-154.6, 0);
    Chisl3:=TOChisl_Create(100, 0);
    Chisl4:=TOChisl_Create(99, 0);
    Chisl5:=TOChisl_Create(152, 0);
    Chisl6:=TOChisl_Create(148, 0);
    Chisl7:=TOChisl_Create(307.7, 0);
    Chisl8:=TOChisl_Create(-82.44, 0);
    Chisl9:=TOChisl_Create(47, 0);

    B:=EExecD00(Chisl1, Chisl2, Chisl3, d1, 1, 1, 1, Att_d1);
    B:=EExecD00(Chisl4, Chisl5, Chisl6, d2, 1, 1, 1, Att_d2);
    B:=EExecD00(Chisl7, Chisl8, Chisl9, d3, 1, 1, 1, Att_d3);

    B:=EExecO3(d1, d2, o1, p1, p2, 1, 1, Att_o1, Att_p1, Att_p2);

    B:=EExecO3(d2, d1, o2, p3, p4, 1, 1, Att_o2, Att_p3, Att_p4);
    B:=EExecP2(o1, o2, p5, 1, 1, Att_p5);

    B:=EExecO3(d1, d3, o3, p6, p7, 1, 1, Att_o3, Att_p6, Att_p7);

    B:=EExecO3(d3, d1, o4, p8, p9, 1, 1, Att_o4, Att_p8, Att_p9);
    B:=EExecP2(o3, o4, p10, 1, 1, Att_p10);

    B:=EExecO3(d2, d3, o5, p11, p12, 1, 1, Att_o5, Att_p11, Att_p12);

    B:=EExecO3(d3, d2, o6, p13, p14, 1, 1, Att_o6, Att_p13, Att_p14);
    B:=EExecP2(o5, o6, p15, 1, 1, Att_p15);
    B:=EExecO000(p5, p10, p15, o7, 1, 1, 1, Att_o7);
    B:=EExecOK001(d1, d2, d3, d4, 1, 1, 1, Att_d4);
    B:=EExecOK001(d1, d4, o7, d5, 1, 1, 1, Att_d5);
    B:=EExecP3(d1, d5, p16, p17, 1, 1, Att_p16, Att_p17);
    B:=EExecOK001(d2, d4, o7, d6, 1, 1, 1, Att_d6);
    B:=EExecP3(d2, d6, p18, p19, 1, 1, Att_p18, Att_p19);
    B:=EExecOK001(d3, d4, o7, d7, 1, 1, 1, Att_d7);
    B:=EExecP3(d3, d7, p20, p21, 1, 1, Att_p20, Att_p21);

```

```
B:=EExecD4(p17,p18,p20,d8,1,1,1,Att_d8);
B:=EExecD4(p16,p19,p13,d9,1,1,1,Att_d9);

TList_Create(List);
TList_Add(List,d1);
TList_Add(List,d2);
. . . . .
TList_Add(List,d9);

DrawList(List);

TList_Destroy(List);

Destroy(d1); ... Destroy(d9);
end.
```

Следует отметить, что, в отличие от микропроцессорных средств общетехнического назначения, работающих под управлением операционных систем, микроконтроллерные устройства обладают существенно меньшей оперативной памятью и недостаточными инструментами ее обслуживания. В частности, менеджер динамической памяти, представленный фирмой Mikroelektronika в ее инструментах прототипирования, не позволяет с достаточной эффективностью производить компактификацию памяти, требующей динамического распределения и освобождаемой после выполнения промежуточных вычислений. Это обстоятельство послужило причиной ряда ограничений, накладываемых на сложность реализованных к настоящему времени геометрических моделей. Ограниченными возможностями обладают также встроенные в платы дисплейные модули, скорость вывода изображений на которые не соответствует задачам конструктивного геометрического моделирования. Однако эти ограничения не являются принципиальными и могут быть преодолены в результате реорганизации вычислительных функций и посредством внешнего подключения к средствам прототипирования внешних высокоскоростных средств синтеза и отображения графической информации.

#### 4 Перспективы реализации конструктивных геометрических алгоритмов в HDL-системах

Анализ структуры конструктивных геометрических моделей показывает их высокую предрасположенность к распараллеливанию реализуемых с их помощью вычислений, а также возможность конвейерной обработки данных. В настоящее время система Симплекс выполняет однотипные, но параллельные по своей вычислительной природе операции, в едином цикле, последовательно в соответствии с представленным в [11, 12] алгоритмом выполнения вычислительной работы геометрической машины. На рис. 2. представлена структура упрощенного алгоритма решения задачи Аполлония о сопряжении трех окруж-

ностей [7]. Несмотря на свою простоту, она демонстрирует особенности геометрических алгоритмов, которые могут получить высокоэффективную реализацию в аппаратуре, построенной на базе программируемых логических интегральных схем. Каждый представленный на схеме блок представляет собой законченную геометрическую операцию, порождающую набор выходных геометрических объектов из совокупности входных объектов.

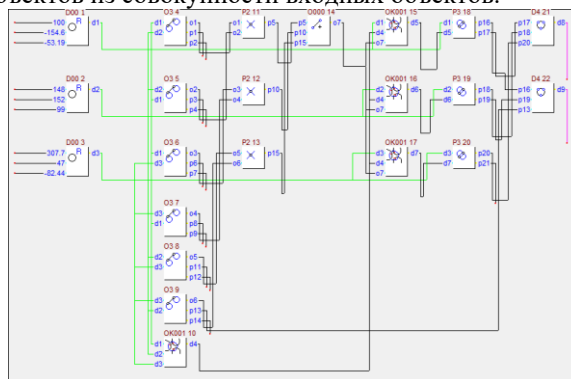


Рис. 21. Структура алгоритма решения задачи Аполлония, допускающая распараллеливание вычислений и их конвейерную организацию

Алгоритм подразумевает обращение к двадцати двум блокам-модулям семи типов (построение окружности по координатам центра и радиусу; построение касательных к двум окружностям; нахождение точки пересечения прямых, построение прямой по трем коллинейным точкам; построение ортогональной окружности; нахождение точек пересечения окружностей; построение окружности по трем точкам). Автоматический анализ структуры алгоритмов позволяет выделить в нем те группы блоков, которые могут совершать вычислительную работу согласованно и одновременно по мере готовности данных.

Такие группы на схеме (рис. 2.) организовались в вертикальные ряды, число которых равно семи. Условно считая время выполнения операций разного типа одинаковым, получаем трехкратное сокращение общего времени вычислений. Разрабатывая архитектуру вычислительного устройства, в котором данные могут синхронно передаваться между представленными слоями, можно обеспечить конвейерную обработку потока данных, при которой скорость вычислений возрастает еще в семь раз. Предложенный подход оказывается особенно эффективным при множественном согласовании параметров геометрических функций [11], когда модули, принадлежащие одному ряду, многократно реплицируются, не нарушая общую структуру алгоритма, что позволяет кратно увеличивать производительность вычислений.

Как было показано ранее, система Симплекс позволяет создавать новые геометрические операции, представляемые в виде отдельных иерархически организованных алгоритмов и использовать их в качестве процедур, вызываемых из других алгоритмов проекта. Возможность пополнения функционального состава системы, безусловно, расширяет области ее применения. Однако использу-



мый для этого процедурный подход непременно приводит к снижению производительности вычислений из-за неизбежного появления дублированных операций, устранение которых при процедурном стиле программирования затруднительно, а в ряде случаев и невозможно. При аппаратной реализации алгоритмов избыточность вычислений недопустима. В связи с этим представляется актуальной задача выполнения логического анализа развернутых структур алгоритмов с целью устранения дублирований и приведения «геометрических подподобных» по аналогии с алгебраическими преобразованиями с целью поиска оптимальных аппаратных конструкций по критериям минимизации времени выполнения операций и общего числа вычислительных элементов.

В настоящее время авторами ведется разработка библиотеки общих и геометрических функций, представленных на языке VHDL, для их апробации и корректировки на отладочных платах с ПЛИС фирм Xilinx и Altera .

## 5 Заключение

Результатом проведенных исследований стала разработка концепции представления конструктивных геометрических моделей в виде программ на устройствах специализированного назначения.

Разработана методика трансляции и подготовки программ конструктивного геометрического синтеза на отладочных платах фирмы MikroElektronika.

Созданы и отлажены библиотеки геометрических процедур на языках Pascal, MikroPascal, JavaScript, MaxScript.

Осуществляется разработка библиотеки геометрических функций на языке VHDL с целью создания экспериментального образца геометрического процессора.

## Литература

1. Адамар Ж. Элементарная геометрия. Часть I. Планиметрия. М.: Учпедгиз, 1948. — 608 с.
2. Аргунов Б.И., Балк М.Б. Геометрические построения на плоскости. М.: Учпедгиз, 1957. — 268 с.
3. Бакельман И.Я. Инверсия. М.: Наука, 1966. — 79 с.
4. Вальков К.И. Введение в теорию моделирования [Текст] / К.И. Вальков. — Л.: ЛИСИ. — 1974. — 152 с.
5. Волков В.Я., Юрков В.Ю., Панчук К.Л., Кайгородцева Н.В. Курс начертательной геометрии на основе геометрического моделирования: учебник / В.Я. Волков, В.Ю. Юрков, К.Л. Панчук, Н.В. Кайгородцева. Омск: Изд.-во СибАДИ, 2010. — 253 с.
6. Волошинов Д.В. Геометрическая лаборатория. Закладываем основы [Текст] // Качество графической подготовки: проблемы, традиции и инновации: Материалы VII международной Интернет-конференции. Февраль - март 2017 г. Пермь, 2017.
7. Волошинов Д.В. Конструктивная геометрическая модель четырехмерного пространства как основа для решения задач зонирования и позиционирования при проекти-

- ровании сетей мобильной связи [Текст] // Труды учебных заведений связи. 2018. Т. 4. No 2. С. 44–60.
8. Волошинов Д.В. Конструктивное геометрическое моделирование. Теория, практика, автоматизация: монография [Текст] / Д.В. Волошинов. — Saarbrücken: Lambert Academic Publishing, 2010. — 355 с.
  9. Волошинов Д.В. О перспективах развития геометрии и ее инструментария [Текст] // Проблемы качества графической подготовки: материалы IV Междунар. интернет-конф.; февраль-март 2014 г. — Пермь, 2014.
  10. Волошинов Д.В. Симплекс. Свидетельство о регистрации программы для ЭВМ RU 2019619710, 23.07.2019.
  11. Волошинов Д.В. Технологии применения геометрического инструмента. Избавление от рутины. [Текст] // Проблемы качества графической подготовки: материалы VIII Междунар. интернет-конф.; февраль-март 2019 г. — Пермь, 2019.
  12. Волошинов Д.В. Технологии применения геометрического инструмента. Логический синтез. [Текст] // Проблемы качества графической подготовки: материалы VIII Междунар. интернет-конф.; февраль-март 2019 г. — Пермь, 2019.
  13. Волошинов Д.В., Соломонов К.Н. Конструктивное геометрическое моделирование как перспектива преподавания графических дисциплин [Текст] / Д.В. Волошинов // Геометрия и графика. — 2013. — Т. 1. — № 2. — С. 10–13.
  14. Вольберг А.О. Основные идеи проективной геометрии. М.-Л.: Учпедгиз, 1949. — 188 с.13.
  15. Гирш А.Г. Наглядная мнимая геометрия / А.Г. Гирш. М.: Маска, 2008. — 216 с.14.
  16. Глаголев Н.А. Проективная геометрия. М.: Высшая школа, 1963. — 342 с.15.
  17. Жижилкин И.Д. Инверсия. М.: Изд-во МЦНМО, 2009. —72 с.
  18. Короткий В.А. Центральное проецирование двух компланарных коник в две окружности// Проблемы качества графической подготовки Материалы IV международной Интернет-конференции. Февраль–март 2014 г. Пермь, 2014.
  19. Пеклич В.А. Мнимая начертательная геометрия: учеб. пособие / В.А. Пеклич. — М.: АСВ, 2007. —104 с.
  20. Сальков Н.А. Свойства циклид Дюпена и их применение. Ч.1 / Н.А.Сальков // Геометрия и графика, 2015. — Т.3.—№1.—С. 16–25. DOI: 10.12737/10454.
  21. Филиппов П.В. Начертательная геометрия многомерного пространства и ее приложения. Изд. 2-е. М.: ЛЕНАНД, 2016. —282 с.



**КОМПЬЮТЕРНОЕ ЗРЕНИЕ**

---

**COMPUTER VISION**

**GraphiCon 2020**

---

22-25 сентября 2020  
Санкт-Петербург, Россия

## Алгоритм подсчета автомобилей в крупномасштабных системах видеонаблюдения\*

А.П. Широков<sup>1</sup>[0000-0001-7597-8511], Д.А. Купляков<sup>1,3</sup>[0000-0002-2957-3297],

А.С. Конушин<sup>1,2</sup>[0000-0002-6152-0021]

<sup>1</sup> Факультет вычислительной математики и кибернетики,  
Московский государственный университет им. М.В. Ломоносова, Москва, Россия  
arsenii.shirokov@graphics.cs.msu.ru

<sup>2</sup> НИУ Высшая Школа Экономики, Москва, Россия  
denis.kuplyakov@graphics.cs.msu.ru

<sup>3</sup> ООО «Технологии видеоанализа», Москва, Россия  
anton.konushin@graphics.cs.msu.ru

**Аннотация.** В работе рассматривается задача подсчета автомобилей в крупномасштабных системах видеонаблюдения. Применяется подход, основанный на сопровождении автомобилей и генерации событий пересечения траектории их движения с заданным сигнальным отрезком. За счет распределенности обработки и применения детекции не ко всем кадрам, а к их разреженному множеству, удается снизить требуемое число вычислительных ресурсов, увеличить производительность и получить алгоритм, способный качественно работать в реальном времени. Адаптируется и модифицируется подход, ранее предложенный для отслеживания людей [1]. Улучшение модуля оценки скорости и уточнение модели движения позволяют уменьшить частоту детекции в 3 раза. Предложенный алгоритм может работать при частоте детекции 3 Гц, сохраняя приемлемое качество.

**Ключевые слова:** компьютерное зрение, сопровождение в видео, подсчет автомобилей.

### 1 Введение

Автоматический подсчет автомобилей дает информацию, необходимую в ряде прикладных задач. Так, зная количество машин, проехавших по каждой дороге в конкретный промежуток времени, мы можем оптимизировать нагрузку на транспортное полотно, эффективно планировать ремонтные работы и проектировать новые дороги. Также полученную информацию можно использовать в различных маркетинговых исследованиях.

---

\* Публикация выполнена при поддержке гранта РФФИ №19-07-00844

Одним из возможных способов решения данной задачи является сопровождение автомобилей, построение их траекторий и подсчет количества пересечений этих траекторий с заданным сигнальным отрезком.

### 1.1 Обзор существующих методов

Современные методы сопровождения автомобилей используют подход к сопровождению через обнаружение. Общей чертой таких алгоритмов является то, что их можно условно разбить на 4 этапа: получение обнаружений, получение признаков обнаружений, построение матрицы стоимости, объединение обнаружений в траектории.

Отличаться же они могут, например, по типу детекций. Так, алгоритмы [4], [5], [7], [8] используют ограничивающие прямоугольники в плоскости кадра, а [6] параллелепипеды. Это различие является существенным, в первом случае мы можем судить только о пространственном расположении автомобиля, а во втором нам известно направление движения и часть траектории. Для более точного сопоставления обнаружений можно проводить вычисления не в плоскости кадра, а в координатах земли [13].

Также алгоритмы могут отличаться связями между этапами трекинга. В некоторых случаях все этапы реализованы отдельными модулями, которые оптимизируются отдельно ([7, 8, 9, 10]). А есть подходы, в которых 3 последних этапа объединяются в единую end-to-end сеть и оптимизируются совместно ([4], [5], [6]).

Еще одно отличие - информация, на основе которой принимается решение о сопоставлении новых обнаружений с траекториями. Могут быть рассмотрены связи между соседними обнаружениями [9], могут учитываться несколько последних обнаружений [5] или же на вход подается вся видеопоследовательность целиком, и есть возможность учитывать обнаружения на всех кадрах: предыдущих и последующих. А в некоторых работах предлагают строить траектории из обнаружений с помощью MCMC (Markov chain monte carlo) [12], [14].

### 1.2 Крупномасштабные системы

Системы дорожного видеонаблюдения содержат большое количество камер, что накладывает ограничения на стоимость установки и эксплуатации отдельно взятой камеры. Интерес представляет разработка алгоритма, способного работать именно в таких системах.

Полностью обрабатывать видеопоследовательность с одной камеры на одном локальном узле становится слишком дорого, поэтому возникает необходимость использовать распределенную систему с центром обработки данных. Все вычисления делать в ЦОД также невыгодно, так как тогда большая нагрузка приходится на каналы передачи информации.

Удачным выходом является распределенная система, описанная в [1] (рис. 1). В ней в ЦОД вычисляются только самые "дорогостоящие" операции, т.е. детекции. Все остальные этапы работы алгоритма вычисляются в локальных узлах,

расположенных возле камеры. Скорость работы системы определяется частотой обращений в ЦОД, т.е. частотой детекции. Поэтому выгодно считать детекции не на всех кадрах, а на разреженном их множестве.

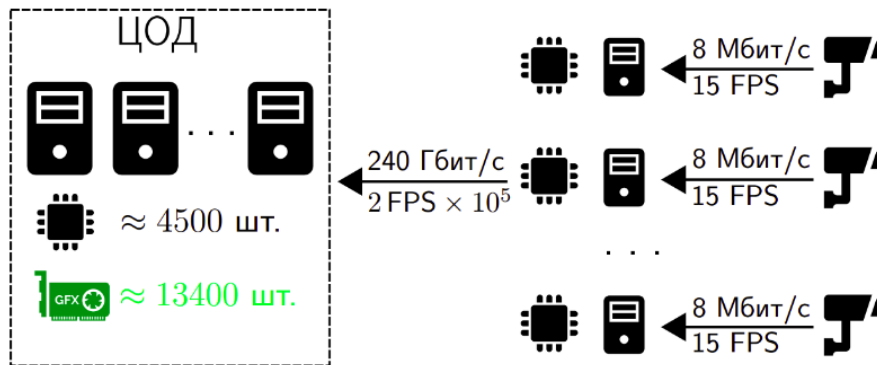


Рис. 1. Обработка видеопотоков в распределенной системе [1] при частоте детекций 2 Гц

## 2 Постановка задачи

В работе рассматривается задача автоматического подсчета автомобилей на интересующей части дороги.

**Формальная постановка задачи.** На вход подается видеопоток  $\{F_i\}$ ,  $F_i \in [0;255]^{W \times H \times 3}$ , кадров, полученных с неподвижной камеры. Сигнальный отрезок, заданный упорядоченной парой точек  $(L_a, L_b)$ .

На выходе — поток событий пересечения сигнального отрезка  $\{E_i\}$ , где  $E_i = (k_i, r_i, d_i)$  описывается номером кадра  $k_i$ , направлением пересечения  $r_i \in \{0, 1\}$  и ограничивающим изображением автомобиля прямоугольником  $d_i = (x_i, y_i, w_i, h_i)$ .

## 3 Предложенный метод

Предложенный алгоритм представляет собой модификацию алгоритма [1]. Он состоит из 4 этапов:

1. Получение детекции;
2. Построение матрицы стоимостей;
  - а. Оценка скорости с помощью визуального сопровождения;
  - б. Предсказание положения детекции на промежуточных кадрах;
3. Сопоставление траекторий и обнаружений;
4. Генерация событий.

Алгоритм был адаптирован к задаче подсчета автомобилей путем изменения детектора, также была добавлена фильтрация обнаружений, о которой подроб-

нее будет написано ниже. Был изменен и 2 этап: улучшена оценка скорости и изменена модель движения. Этапы 3 - 4 остались без изменения.

### 3.1 Получение детекции

По аналогии с базовым алгоритмом детекция производится не на всех кадрах, а только на ключевых:  $F_1, F_{1+\Delta}, F_{1+2\Delta}, F_{1+3\Delta}$  и т.д.,  $\Delta$  — параметр алгоритма, определяющий частоту детекции. Для получения набора ограничивающих изображения автомобилей в кадре прямоугольников используется детектор Detectron Faster R-CNN R-101-FPN [11], обученный на COCO 2017. Далее детекции проходят через ряд фильтров:

1. Фильтрация классов, не относящихся к автомобилям.  
Так как детектор обучен на COCO 2017, он выдает обнаружение неинтересующих нас классов, которые мы отсеиваем;
2. Фильтрация обнаружений, находящихся за пределами ROI (Region of interest).  
На этом этапе отсеиваем детекции, выходящие за пределы рассматриваемых участков дороги;
3. Фильтрация неуверенных обнаружений.  
Убираются детекции с уверенностью ниже заданного порога (0,6);
4. Последняя фильтрация связана с особенностью работы детектора. Чтобы исключить случаи, когда на 1 объект приходится несколько обнаружений (рис. 2(a)), соответствующих разным классам, используется алгоритм подавления не максимумов с высоким порогом (IOU  $\geq 0,95$ ).

**Таблица 1.** Параметры детектора [11]. *Box AP* - средняя точность по всем классам валидационной части COCO 2017; *производительность* – среднее время работы с одним изображением на GPU

Модель	Производительность (сек.)	Box AP
Faster R-CNN R-101-FPN	0,119	39,8



**Рис. 2.** (a) Визуализация работы детектора. На белом фургоне в центре кадра 2 обнаружения различных классов (b) Визуализация траекторий и их пересечений с сигнальным отрезком

### 3.2 Модификация оценки скорости

Важным этапом базового алгоритма является оценка скорости. Так как при разряжении ключевых кадров увеличивается расстояние между ними, то мы должны с достаточной точностью предсказывать положение автомобиля на промежуточном кадре, для чего нужна достаточно точная оценка скорости. С этой целью была предложена модификация модуля оценки скорости.

Для оценки скорости автомобиля на кадре  $F_N$  нужно найти смещение положения автомобиля за следующие  $\varepsilon$  кадров. Для этого выберем параметр алгоритма  $steps\_cnt$ , построим последовательность  $\{F_N, F_{N+\varepsilon/steps\_cnt}, F_{N+2\varepsilon/steps\_cnt}, \dots, F_{N+\varepsilon}\}$  и последовательно применим визуальный трекер к соседним парам кадров.

Отметим, что значение параметра  $steps\_cnt = 1$  соответствует работе оценки скорости в базовом алгоритме. Улучшение точности достигается за счет того, что визуальный трекер лучше работает на небольших смещениях, а использование промежуточных кадров уменьшает это смещение. Однако увеличение параметра  $steps\_cnt$  усложняет вычисления на локальном узле, визуальное сопровождение запускается для большего числа пар кадров.

### 3.3 Модель равноускоренного движения

Исходя из того, что автомобили часто меняют свою скорость, было решено изменить модель фильтра Калмана с равномерного движения на равноускоренное. Для этого достаточно добавить новые состояния, отвечающие за ускорение.

Таким образом, получим модель фильтра Калмана с состояниями  $x = (x, y, v_x, v_y, a_x, a_y, s, v_s, k)$ , где единственное скрытое состояние  $v_s$  - скорость изменения площади, а другие состояния:  $(x, y)$  - положение автомобиля,  $(v_x, v_y)$  - скорость движения,  $(a_x, a_y)$  - ускорение движения,  $s$  - площадь ограничивающего прямоугольника,  $k$  - отношение высоты и ширины ограничивающего прямоугольника. Матрица перехода представлена в формуле (1) ( $t$  - половина времени между соседними ключевыми кадрами):

$$\bar{x}^t = \begin{pmatrix} x + v_x t + \frac{a_x t^2}{2} \\ y + v_y t + \frac{a_y t^2}{2} \\ v_x + a_x t \\ v_y + a_y t \\ a_x \\ a_y \\ s + v_s t \\ v_s \\ k \end{pmatrix}. \quad (1)$$

Для оценки ускорения на кадре мы использовали следующий алгоритм:

1. Предположим,  $F_N$  - кадр, на котором нужно оценить ускорение.



2. С помощью визуального сопровождения найдем положение автомобиля с ключевого кадра  $F_N$  на  $F_{N+\varepsilon}$ .
3. По найденному смещению оценим начальную скорость  $v_0=(v_{0x}, v_{0y})$ .
4. Аналогично найдем смещение на кадре  $F_{N+\varepsilon}$  и оценим ускорение по формуле (2):

$$a_x = 2 \frac{x' - x - v_{0x}t'}{t'^2}, a_y = 2 \frac{y' - y - v_{0y}t'}{t'^2}, \quad (2)$$

где  $(x', y')$  положение автомобиля на кадре  $F_{N+\varepsilon}$ ,  $(x, y)$  положение автомобиля на кадре  $F_N$ ,  $t'$  – время между кадрами  $F_N$  и  $F_{N+\varepsilon}$ .

## 4 Экспериментальная оценка

### 4.1 Описание метрики

Для оценки качества полученного решения используется метрика, описанная в [1]. Разобьем видеопоследовательность на видеофрагменты, на которых произошло хотя бы 10 эталонных событий. Построим множество  $Q$  пар номеров кадров, ограничивающих эти видеофрагменты.

Для каждой пары  $(a, b)$  из  $Q$  посчитаем относительную ошибку подсчета  $E_{(a,b)}$  по формуле (3):

$$E_{(a,b)} = \frac{|FP_{(a,b)} - FN_{(a,b)}|}{GT_{a,b}}. \quad (3)$$

Итоговая метрика вводится, как средняя относительная ошибка на всех рассматриваемых отрезках и представлена в формуле (4):

$$E = \frac{\sum_{(a,b) \in Q} E_{(a,b)}}{|Q|}. \quad (4)$$

### 4.2 Описание набора данных для тестирования

Для тестирования базового алгоритма и его модификации были использованы наборы с ручной разметкой траекторий автомобилей для задачи сопровождения. Всего использовали 2 набора: UA-Detrac challenge [2] и GRAM Road-Traffic Monitoring [3], содержащие в общей сложности 103 видеопоследовательности, снятые с 27 различных камер. Встречались сцены различного уровня сложности: простые (камера расположена сверху и направлена вдоль движения, окклюзий не происходит) и достаточно сложные (постоянные окклюзии из-за столбов, разделителей полос, пересечения потоков автомобилей). Для каждой видеопоследовательности вручную был проведен сигнальный отрезок и получена эталонная разметка. В общей сложности наборы содержат 10,5 часов видео и примерно 7200 эталонных событий.



Рис. 3. Примеры сцен для тестирования

### 4.3 Качество подсчета автомобилей

Был проведен ряд экспериментов с каждой модификацией в отдельности и их объединением. Результаты приведены в таблице 2. Используемые обозначения:

- SORT - simple online and real time tracking[15];
- baseline - базовый алгоритм [1] с заменой детектора (см. раздел 3.1);
- ул. оц. скорости (steps\_cnt=N) - улучшение оценки скорости с указанием значения параметра (см. разд. 3.2);
- равноуск. движ. - модификация модели фильтра Калмана для учета равноускоренного движения (см. раздел 3.3).

Сравнив строку 2 со строками 3 и 4, можно заметить, что модификация оценки скорости дает улучшение точности. Аналогично, сравнив строки 2 и 5 можно убедиться в эффективности замены модели фильтра Калмана. Можно сделать вывод, что каждая из предложенных модификаций в отдельности улучшает качество алгоритма.

Наиболее значимый результат достигается при частоте детекции 3 Гц и применении обеих модификаций (последняя строка таблицы 2). Относительная ошибка составляет 5,4%, что незначительно хуже наилучшей ошибки базового алгоритма (5,2% при 10 Гц). Таким образом, удастся сократить число обращений к ЦОД в  $\frac{10}{3}$  раз при сохранении приемлемого качества.

Стоит отметить, что предложенные модификации начинают работать в полной мере при частоте детекции 10 Гц и ниже. Связано это с тем, что при 20 Гц почти все кадры являются ключевыми и не хватает промежуточных кадров для задуманной работы модулей оценки скорости и ускорения. В таблице результаты соответствующих экспериментов не указаны.

Современные методы трекинга автомобилей обычно не преследуют цель работать с разреженными кадрами. Качество таких алгоритмов сильно падает при снижении частоты детекции. Чаще всего это связано с тем, что алгоритмы полагаются на малое смещение объекта в кадре. Высокая частота детекции приводит к выполнению этого условия. В строке 1 Таблицы 2 в качестве примера было рассмотрено одно из призовых решений [9] NVIDIA AI City Challenge, использовавшее для отслеживания автомобилей метод SORT [15].

**Таблица 2.** Результаты экспериментов. Указаны значения ошибки подсчета  $E$  (в процентах). Выделены лучшие результаты в столбце.

Метод / Частота детекции	20 Гц	10 Гц	5 Гц	3 Гц
SORT[15]	5,5	8,6	16,5	50,9
baseline	5,2	5,2	7,6	8,3
baseline + ул. оц. скорости (steps_cnt=2)	-	4,7	4,7	5,9
baseline + ул. оц. скорости (steps_cnt=3)	-	4,7	4,7	5,9
baseline + равноускор. движ	-	4,9	6,8	6,9
baseline + ул. оц. скорости (steps_cnt=3) + равноускор. движ	-	4,8	4,9	5,4

## 5 Заключение

Предложен алгоритм подсчета автомобилей, разработанный на основе [1]. Частота детекций была снижена более чем в 3 раза по сравнению с базовым методом, что позволяет сократить объем вычислительных ресурсов, требуемых для детекции, и увеличить скорость работы без потери качества.

В будущем планируется перейти от плоскости кадра к мировой системе координат за счёт автоматической калибровки камеры [13]. Это сделает модель движения фильтра Калмана более точной, что улучшит качество трекинга.

## Литература

1. Kuplyakov, D.A., Shalnov, E.V., Konushin, V.S. et al. A Distributed Tracking Algorithm for Counting People in Video. Program Comput Soft 45, 163–170 (2019).
2. Wen L. et al. UA-DETRAC: A new benchmark and protocol for multi-object detection and tracking //arXiv preprint arXiv:1511.04136. – 2015.

3. Guerrero-Gómez-Olmedo R. et al. Vehicle tracking by simultaneous detection and viewpoint estimation //International Work-Conference on the Interplay Between Natural and Artificial Computation. – Springer, Berlin, Heidelberg, 2013. – C. 306-316.
4. Chu P., Ling H. Famnet: Joint learning of feature, affinity and multi-dimensional assignment for online multiple object tracking //Proceedings of the IEEE International Conference on Computer Vision. – 2019. – C. 6172-6181.
5. Sun S. J. et al. Deep affinity network for multiple object tracking //IEEE transactions on pattern analysis and machine intelligence. – 2019.
6. Li C. et al. TrackNet: Simultaneous Object Detection and Tracking and Its Application in Traffic Video Analysis //arXiv preprint arXiv:1902.01466. – 2019.
7. Pirsiavash H., Ramanan D., Fowlkes C. C. Globally-optimal greedy algorithms for tracking a variable number of objects //CVPR 2011. – IEEE, 2011. – C. 1201-1208.
8. Ooi H. L. et al. Multiple object tracking in urban traffic scenes with a multiclass object detector //International Symposium on Visual Computing. – Springer, Cham, 2018. – C. 727-736.
9. Kumar A. et al. A semi-automatic 2D solution for vehicle speed estimation from monocular videos //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. – 2018. – C. 137-144.
10. López-Sastre R. J. et al. Boosting multi-vehicle tracking with a joint object detection and viewpoint estimation sensor //Sensors. – 2019. – T. 19. – №. 19. – C. 4062.
11. Girshick, R., Radosavovic, I., Gkioxari, G., Doll'ar, P., He, K.: Detectron. <https://github.com/facebookresearch/detectron> (2018)
12. Kuplyakov D., Shalnov E., Konushin A. Markov chain monte carlo based video tracking algorithm // Programming and Computer Software. — 2017. — Vol. 43, no. 4. — P. 224–229.
13. Shal'nov E. V., Gringauz A. D., Konushin A. S. Estimation of the people position in the world coordinate system for video surveillance // Programming and Computer Software. — 2016. — Vol. 42, no. 6. — P. 361–366.
14. Benfold B., Reid I. Stable multi-target tracking in real-time surveillance video //CVPR 2011. – IEEE, 2011. – C. 3457-3464.
15. Bewley A. et al. Simple online and realtime tracking //2016 IEEE International Conference on Image Processing (ICIP). – IEEE, 2016. – C. 3464-3468.

## Роль HD-карт в беспилотном транспорте\*

А.И. Гайош<sup>1</sup>

<sup>1</sup>Университет ИТМО, 197101, Санкт-Петербург, Россия  
alexander.gaiosh@gmail.com

**Аннотация.** В этой статье рассматриваются возможности применения картографических решений в автономном вождении. Приводится описание и сравнение форматов OpenDRIVE и Lanelet, приводится описание использования карт в системах восприятия, локализации и планирования движения автономного транспортного средства, а также приводится концепция цифровой модели дороги.

**Ключевые слова:** автономный транспорт, HD-карты, навигация, моделирование дорожно-транспортных сетей, восстановление дороги

### 1 Введение

Высокоавтоматизированные транспортные средства (ВТС) разрабатываются с целью снижения влияния человеческого фактора на дорожный трафик и повышение безопасности на дорогах. Первые попытки создания ВТС проводились ещё в 80-ых годах 20-го века. В период с 1987 по 1995 год в Европе проводился проект Eureka PROMETHEUS [1], являющийся первым крупным исследованием в области автоматизированного вождения. Опираясь на результаты исследований, компания Daimler-Benz разработала автомобиль VITA II [2], в котором была представлена первая система адаптивного круиз-контроля.

Во время DARPA Urban Challenge [3], состоявшегося в 2007 году, многие исследовательские группы со всего мира получили возможность опробовать свои решения в тестовой среде, смоделированной в виде городского трафика. Это соревнование позволило привлечь общественный интерес к сфере беспилотного транспорта. Разработанные решения требовали внедрения и дальнейших исследований. Однако, для выхода систем автономного вождения на рынок требовалось расширение набора используемых технологий как в автомобилях, так и в дорожной инфраструктуре.

Одним из ключевых механизмов автономного вождения, позволяющим связать транспортное средство с окружающей средой, стало использование карт высокого разрешения (HD-карты). Применение этой технологии охватывает три основные проблемы автономного вождения. Первая проблема - способность транспортного средства определять свое местоположение и местоположение

---

\* Публикация выполнена при поддержке гранта РФФИ № 19-07-00844

других объектов с высокой точностью по отношению к окружающей среде. Вторая - реакция на события, возникающие как на пути движения беспилотного автомобиля, так и вне зоны видимости бортовых датчиков. Третья – построение маршрута и маневрирование с соблюдением правил дорожного движения [4].

В данной статье приводится обзор того, какую роль играют HD-карты в архитектуре автономного автомобиля, и почему для беспилотного вождения вообще требуются специальные карты. Статья организована следующим образом: часть II раскрывает понятие цифровой модели дороги, описывает, какую информацию должна включать HD-карта, содержит описание и сравнение форматов хранения HD-карт; часть III рассказывает как HD-карты влияют на системы восприятия, планирования движения и локализации транспортного средства в пространстве; в заключении приводятся выводы, а также выделяются актуальные направления исследовательских работ в сфере HD-картографии.

## 2 Форматы представления HD-карт

Применение географических информационных систем (ГИС) решает задачи поиска оптимального маршрута и отслеживания движения по нему. Однако, точность этих карт находится в метровом диапазоне, тогда как беспилотники требуют точности в 5-10 сантиметров, поэтому обеспечивающие такую точность карты и называются картами высокого разрешения (англ. High Definition Maps, HD-Maps). Помимо этого, ГИС не предоставляют полной информации о дорожной сети, которая требуется для движения автономного автомобиля. Вся эта информация объединяется под общим понятием – цифровая модель дороги. Эта сущность содержит внутри себя описание геометрии дороги, количество полос и их границы, логическую связь полос между собой, дорожную разметку и скоростные ограничения, дорожные знаки и их привязку к полосам движения, светофоры, стоп-линии и пешеходные переходы. Существует несколько принципиально разных форматов представления и хранения HD-карт. Текущий раздел посвящен обзору и сравнению этих форматов.

### 2.1 Формат OpenDRIVE

Формат хранит данные в виде XML документа [5]. Дорожное пространство рассчитывается на основе эталонной кривой, которая описывает геометрию всего пространства на заданном участке дороги. Полосы движения генерируются путем построения кривых Безье на определенном расстоянии от эталонной кривой. Каждая кривая имеет собственный номер от 1 до N для полос в направлении движения и от -1 до -N на встречном движении, где N – количество полос на текущем участке дороги. Данный формат нашел широкое применение в различных симуляторах и средах моделирования дорожного трафика [6], где не требуется сантиметровая точность и абсолютное соответствие геометрии дорожного пространства.

## 2.2 Формат lanelet

Lanelet – формат, в котором дорожная сеть представляется в виде логически связанных друг с другом участков конечной длины – полос [7]. Полосы ограничены двумя кривыми, для каждой из которых явно указывается её назначение – правый или левый край. Границы представлены конечным множеством точек, каждая из которых описывается координатами трехмерного пространства. Каждая полоса хранит в себе метаинформацию о логической связи с другими полосами, а также об объектах, привязанных к данной полосе, например, дорожных знаках или пешеходных переходах. Наличие связей между полосами и ориентация границ позволяет построить ориентированный граф дорожно-транспортной сети, на основе которого можно генерировать множество траекторий для маневрирования внутри полосы. Данный формат нашел широкое применение в различных научных проектах, таких, как Autoware [8].

## 2.3 Анализ и сравнение форматов хранения HD-карт

Отличительной чертой формата OpenDRIVE является то, что благодаря кривым Безье кривизна границ дороги равномерна на всем участке дороги, что обеспечивает генерацию плавной траектории. Однако, далеко не всегда полосы имеют одинаковую ширину на протяжении всего участка дороги, что требует дробления эталонной кривой на большое количество сегментов. Кроме того, этот формат спроектирован исключительно для хранения HD-карт, он не предусматривает возможности внесения изменений в карты и не предоставляет никакого функционала для этого. Таким образом, каждый проект, разрабатывающий HD-карты в формате OpenDRIVE самостоятельно реализует инструментарий для создания карт и их редактирования. Также, зачастую многие проекты берут этот формат за основу и модифицируют под собственные задачи. Следствием этого является большое количество неунифицированных закрытых форматов, которые несовместимы друг с другом.

Один из самых крупных и известных проектов в области автономного вождения, создавших свой формат хранения карт на базе OpenDRIVE является Apollo [8]. Проект также обеспечил совместимость собственного формата со стандартным OpenDRIVE. Также существует инструмент, позволяющий конвертировать карты OpenStreetMap в OpenDRIVE [9]. Применение таких карт ограничивается средами симуляции, когда решающим фактором является возможность быстро сгенерировать карту на основе открытых источников для запуска симуляции, а требованиями к точности и соответствия геометрии можно пренебречь.

Альтернатива OpenDRIVE - формат Lanelet, набирающий все большую популярность. Самая распространённая реализация этого формата – библиотека Lanelet2 [10]. Она содержит как описание самого формата, так и предоставляет API на языках C++ и Python для работы с картами. Помимо этого, представление данных основывается на формате OpenStreetMap, что обеспечивает интеграцию с инструментом для создания и редактирования карт JOSM [11].

Формат Lanelet удобен как на этапе релиза, когда карты эксплуатируются, так и на этапе разработки, когда постоянно требуется добавлять в карты новые элементы и редактировать существующие. Границы полос, выполненные в виде множества точек, позволяют гибко и просто очерчивать и изменять геометрию пространства и местоположение элементов дорожной инфраструктуры. Это гарантирует точное соответствие геометрии дорожного пространства карты с реальной дорожной разметкой, благодаря чему отпадает необходимость в дополнительной верификации генерируемых маршрутов. Однако, к его недостаткам, можно отнести отсутствие решений для автоматизации создания карт. Сейчас все карты создаются вручную, что занимает гораздо больше времени, чем генерация карт на базе открытых ГИС.

### 3 HD-карты в автономном транспорте

#### 3.1 HD-карты и система локализации

Ключевым компонентом в автономном автомобиле является система локализации. Автомобилю недостаточно знать свое абсолютное местоположение и ориентацию, ему также необходима информация об окружающих его объектах, таких как границы дороги или пешеходные переходы. Эту информацию можно получить из карты, определив свое местоположение в ней.

Самый распространенный подход для получения абсолютного местоположения – использование GNSS приёмников, которые позволяют получать местоположение автомобиля со спутников, затем следует поиск ближайшей точки в карте к полученным со спутника координатам, тем самым автомобиль определяет свое местоположение в карте. Этот подход широко используется в современных навигационных системах и дорожных информационных сервисах. Однако, получаемая точность недостаточна для движения автономных автомобилей в городе, поскольку в лучшем случае GNSS достигает метровой точности, в то время как автономное вождение требует точности определения местоположения до 5-10 сантиметров. В таком случае используют технологию RTK-поправок [12], которые позволяют достичь требуемой точности. Однако, в чистом виде GNSS RTK плохо подходит для движения в городских условиях из-за большого количества зон, где сигнал не может быть принят, например, в тоннеле или при движении в условиях плотной застройки.

Альтернативным способом локализации является применение алгоритмов одновременной локализации и картирования (SLAM) [13]. Проецирование собранных в реальном времени данных с бортовых датчиков на HD-карту позволяет точно определить местоположение автомобиля как в карте, так и по отношению к другим участникам дорожного движения. Недостатком являются крайне высокие требования к вычислительной мощности и каналу передачи данных.

Независимо от того, какой метод применяется, использование HD-карт позволяет определить местоположение автомобиля в пространстве и получить из



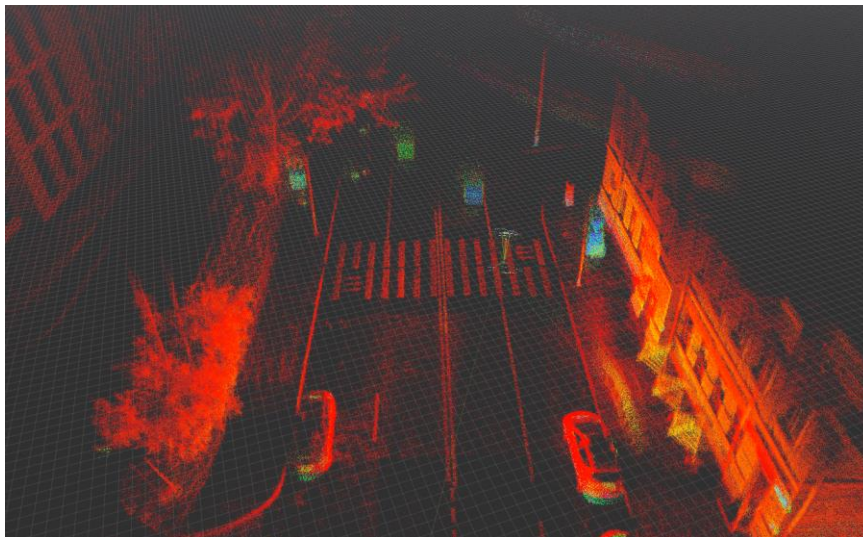
карты весь доступный контекст об окружающей обстановке. Этот контекст в свою очередь используют система планирования маршрута и система восприятия.

### 3.2 HD-карты и система восприятия

Автономный автомобиль должен уметь обнаруживать и объезжать препятствия, причем делать это в соответствии с правилами дорожного движения. В дополнение к неподвижным препятствиям на дорогах находятся и движущиеся объекты, такие как автомобили, пешеходы и велосипедисты. За решение подобных задач в беспилотном автомобиле отвечает система восприятия. Полученные объекты она проецирует на HD-карту, чтобы объекты и препятствия учитывались при планировании движения. Как правило, система восприятия состоит из набора разнообразных датчиков и механизмов обработки, сегментации и классификации информации, полученной от них.

Сенсорика автономного автомобиля достаточно разнообразна, она включает в себя целый ряд датчиков, таких как радары, лидары, камеры, средства инерциальной навигации. Каждый датчик имеет свои ограничения по эксплуатационным условиям и области видимости. Поэтому набор датчиков подбирают из задач и условий эксплуатации ВТС так, чтобы недостатки одних датчиков компенсировались преимуществами других. Многие крупные проекты, такие как Waymo [14] и HERE [15], конфигурируют специальные картографические автомобили, задача которых - сбор данные с разных локаций. Такие автомобили, как правило, укомплектовываются несколькими лидарами с круговым обзором, камерами и GNSS-приемником для получения абсолютных координат. Подобный набор датчиков позволяет получить плотное облако точек сантиметровой точности, в котором содержатся все необходимые данные для создания карты, пример такого облака точек можно увидеть на рис.1. В облаке отчетливо видны границы дороги, линии разметки, припаркованные автомобили, пешеходный переход и дорожные знаки.

В облаке точек сперва выделяются края дороги и границы полос. Это позволяет обрисовать основную геометрию дорожного пространства с точной привязкой к реальному расположению объектов. Эта геометрия ограничивает пространство, в котором автомобиль может двигаться. Дальше границы полос в облаке точек аннотируются как линии разметки. Линии разметки определяют возможность перестроения и маневрирования внутри очерченной геометрии. Кроме того, в облаке точек выделяется расположение элементов дорожной инфраструктуры - дорожных знаков, стоп-линий, пешеходных переходов, светофоров, парковочных мест.



**Рис. 4.** Сырое облако точек, содержащее в себе информацию о дорожной инфраструктуре

Разметка данных зачастую происходит в ручном режиме. Также существует конвейер, позволяющий при помощи методов глубокого обучения восстанавливать из облака точек дорожно-транспортную сеть [16], однако его результаты требуют дополнительной верификации, потому что часто возникают ошибки при привязке светофоров к полосе движения, из-за чего автомобиль может получить ошибочную информацию и выполнить некорректный маневр.

### 3.3 HD-карты и система планирования движения

Для автономного движения автомобиль должен самостоятельно уметь планировать маршрут. Использование HD-карт позволяет формировать граф пространства с точной привязкой к абсолютным координатам для поиска глобального маршрута. Также это позволяет рассчитать локальный маршрут фиксированной длины для фильтрации объектов и генерации траекторий маневрирования внутри полосы, что планировщики на базе ГИС делать не позволяют.

Планировщик глобального маршрута располагает всей информацией о геометрии дороги и ее разметке. Для сформированного глобального маршрута генерируется набор возможных траекторий движения с учетом разметки и границ полос. Каждая траектория представляет собой набор последовательных точек с фиксированным шагом, содержащих информацию о позиции и ориентации транспортного средства, а также содержит метаинформацию об элементах дорожной инфраструктуры. Таким образом, используя HD-карты, система планирования может находить маршрут и траектории маневров с привязкой к реальной геометрии дороги и расположению автомобиля в полосе.

### 3.4 Предсказание траекторий объектов на основе HD-карт

Значительным преимуществом использования HD-карт в автономном вождении является возможность предсказать поведение других участников дорожного движения. Пример работы системы предсказания представлен на рис. 2. Исходя из траекторий, которые извлекаются из цифровой модели дороги, можно сгенерировать все возможные траектории движения любого объекта, который способна детектировать система восприятия автомобиля, и отображать эти траектории на HD-карту для поиска коллизий с траекторией беспилотного автомобиля [17]. Это значительно упрощает процесс перестроений между полосами, а также позволяет избегать столкновений с другими транспортными средствами.

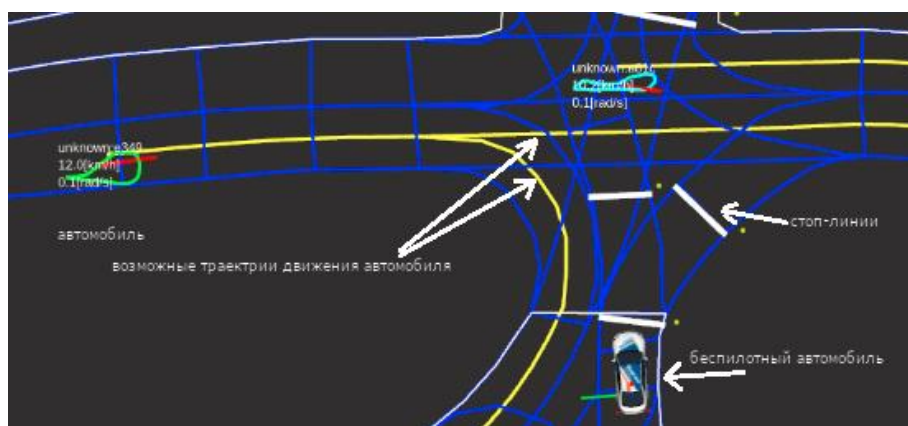


Рис 5. Визуализация возможных траекторий движения обнаруженных автомобилей

## 4 Заключение

Максимальный уровень автономности автомобиля — полностью автоматизированное вождение в любых условиях окружающей среды, без вмешательства человека, на данный момент времени не достигнут. Это связано с большим количеством проблем, особенно в городских условиях: высокая плотность движения, строгие требованиями к реальному времени, разнообразие участников дорожного движения, при отсутствии гарантии соблюдения правил дорожного движения этими участниками.

Использование HD-карт позволяет упростить программное обеспечение, а также позволяет выгружать из карт различную информацию о дорожно-транспортной сети для ее использования в узлах системы управления. Это позволяет автомобилю определять свое местоположение по отношению к другим объектам, генерировать оптимальные маршруты с учетом актуальной дорожной обстановки, выполнять маневры с учетом траекторий движения других транспортных средств.

Актуальными инженерными и научными проблемами в HD-картографии на сегодняшний день являются: унификация формата HD-карт; автоматизация сбора данных; автоматизации построения карт; разработка новых алгоритмов и методов построения маршрута, которые будут учитывать геометрию дороги и габариты транспортного средства.

Существующие форматы хранения HD-карт предоставляют разные возможности и их применение отличается. OpenDRIVE широко используется в средах моделирования, где важна скорость создания карт, а их точностью можно пренебречь, однако практически не предоставляет инструментов для работы с картами, кроме того на базе данного формата создается множество закрытых форматов, использующихся в различных проектах, но они несовместимы друг с другом. Lanelet набирает популярность в исследовательских проектах для создания карт, используемых при автономном движении, где важна точность, предоставляет различные инструменты для создания и обработки карт, однако не имеет средств автоматической генерации карт, из-за чего создание карт требует времени и проверок на ошибки, вызванные человеческим фактором. Выбирать формат следует, исходя из целей и области применения создаваемых карт.

## Литература

1. Williams, M.: PROMETHEUS-The European research program for optimizing the road transport system in Europe. IEE Colloquium on Driver Information, London, UK, 1988, pp. 1/1-1/9.
2. Ulmer, B.: Vita ii-active collision avoidance in real traffic. Intelligent Vehicles' 94 Symposium, Proceedings of the. IEEE, 1994, pp. 1–6.
3. The darpa urban challenge: autonomous vehicles in city traffic, vol. 56, 2009. <https://www.springer.com/gp/book/9783642039904>
4. Redzic, O., Rabel, D.: A location cloud for highly automated driving. In: Meyer G, Beiker S, editors Road vehicle automation 2. Switzerland: Springer International Publishing; 2015. p. 49–60
5. Detailed Specification – OpenDRIVE, <http://www.opendrive.org/docs/OpenDRIVEFormatSpecRev1.4H.pdf>. Last accessed 14 Jul 2020
6. Eclipse SUMO - Simulation of Urban MObility, <https://www.eclipse.org/sumo/>. Last accessed 14 Jul 2020
7. Bender, P., Ziegler J., Stiller, C.: Lanelets: Efficient map representation for autonomous driving. in Proc. of the IEEE Intelligent Vehicles Symposium, 2014, pp. 420–425.
8. Raju, V., Gupta, V., Lomate, S.: Performance of Open Autonomous Vehicle Platforms: Autoware and Apollo. 2019 IEEE 5th International Conference for Convergence in Technology (I2CT), Bombay, India, 2019, pp. 1-5. <https://doi.org/10.1109/I2CT45611.2019.9033734>.
9. Althoff, M., Urban, S., Koschi, M.: Automatic Conversion of Road Networks from OpenDRIVE to Lanelets. 2018 IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI), Singapore, 2018, pp. 157-162, <https://doi.org/10.1109/SOLI.2018.8476801>

10. F. Poggenshans, F.: Lanelet2: A high-definition map framework for the future of automated driving. 2018 21st International Conference on Intelligent Transportation Systems (ITSC), Maui, HI, 2018, pp. 1672-1679. <https://doi.org:10.1109/ITSC.2018.8569929>
11. JOSM Homepage, <https://josm.openstreetmap.de>. Last accessed 14 Jul 2020
12. Dabove, P., Di Pietra, V.: Single-baseline RTK positioning using dual-frequency GNSS receivers inside smartphones. *Sensors*, 19(19), 4302 (2019).
13. Huang, B., Zhao, J., Liu, J.: A Survey of Simultaneous Localization and Mapping. arXiv preprint arXiv:1909.05214 (2019).
14. Bansal, M., Krizhevsky, A., & Ogale, A.: Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst. arXiv preprint arXiv:1812.03079 (2018).
15. HD Maps for Autonomous Driving and Driver Assistance | HERE, <https://www.here.com/platform/automotive-services/hd-maps>. Last accessed 14 Jul 2020
16. Elhousni, M., Lyu, Y., Zhang, Z., & Huang, X.: Automatic Building and Labeling of HD Maps with Deep Learning. arXiv preprint arXiv:2006.00644 (2020).
17. Marks, P.: Quantum positioning system steps in when GPS fails. *New Scientist*, 222, 19 (2014).

## Метод распознавания шрифтов на основе медиального представления\*

А.Л. Липкина<sup>1</sup>[0000-0002-7177-6450], Л.М. Местецкий<sup>2</sup>

<sup>1</sup> Московский государственный университет имени М.В. Ломоносова, Москва, Россия  
lipkina96@mail.ru, mestlm@mail.ru

**Аннотация.** В статье описывается метод распознавания шрифтов на основе медиального представления, интегрированный в систему распознавания шрифтов по цифровому изображению текста. Эта система ищет похожие шрифты, упорядоченные по схожести, на шрифт, изображенный на введенном пользователем изображении текста. Работа системы основана на решении двух задач машинного обучения: распознавания текста на изображении и распознавания шрифта по изображению текста. Для решения первой задачи используется понятие математической модели графемы, основанной на непрерывном медиальном представлении символа. Решение задачи распознавания шрифта основано на понятии морфологической ширины фигуры, также тесно связанной с медиальным представлением. Мы предлагаем метод использования функции морфологической ширины для поиска наиболее похожих шрифтов из известной базы. Проведенные эксперименты показывают высокую точность поиска наиболее похожих шрифтов. Для базы, состоящей из 2543 шрифтов, точность составляет 0.991 по метрике  $\text{top}@5$  для правильно распознанного текста в размере шрифта 100 пикселей на изображении.

**Ключевые слова:** графема, морфологическая ширина, цифровое изображение текста, цифровой шрифт, медиальное представление.

### 1 Введение

В мире существует огромное количество различных шрифтов. У графических дизайнеров, художников, предпринимателей и других людей часто возникает необходимость определить шрифт, в котором слово изображено на картинке, сканированном изображении или фотографии или найти похожие на него шрифты. Причины могут быть разные, например, проверить, что на шрифт есть лицензия или найти похожий свободный шрифт. Сейчас существуют форумы, на которых люди могут вывесить свою картинку со шрифтом и попросить кого-нибудь определить шрифт, но не всегда можно получить таким образом ответ. Как другой вариант, можно попросить эксперта определить шрифт на изображении, однако ввиду большого визуальный метод определения шрифтов может

---

\* Работа выполнена при поддержке гранта РФФИ № 20-01-00664.

приводить к ошибкам. Поэтому для данной задачи важны автоматизированные системы распознавания шрифтов.

В настоящее время существует ряд web-сайтов, например: WhatTheFont\*, WhatFontIs†, FontSquirrel‡, Identifont§, которые способны распознавать шрифты по загруженному изображению. Однако не все из них являются удобными с точки зрения пользователя (сложный диалог взаимодействия с пользователем) и не все приспособлены к определению кириллических шрифтов. В дополнение к этому сами алгоритмы в сервисах закрытые, что создает неудобства для исследователя в области распознавания шрифтов.

Статей по рассматриваемой тематике довольно мало, и тем более еще меньше статей, в которых рассматриваются методы, умеющие распознавать шрифт по большой базе различных шрифтов. В большинстве из самых цитируемых статей [9, 8] тестирование методов проводится на маленьких выборках (20-50 шрифтов) и результаты экспериментов показывают высокую точность распознавания. В некоторых статьях [1, 2, 7] рассматриваются большие базы тестируемых шрифтов (1000 - 2000 шрифтов), однако в них по результатам экспериментов наблюдается меньшая точность по сравнению с предыдущими перечисленными статьями.

В данной статье мы предлагаем метод распознавания шрифтов (см. рис. 1), заложенный в системе распознавания шрифтов, которая умеет распознавать текст на изображении и находить похожие шрифты из базы, состоящей из 2543 шрифтов, поддерживающих кириллический алфавит [13], причем от и до она завязана на морфологических признаках, характеризующих структуру символа. Система умеет не только определять изображенный на картинке шрифт, но и выдавать похожие на него шрифты из выбранной базы в виде списка изображений шрифтов. Изображения в списке отсортированы по убыванию степени похожести шрифтов на действительный шрифт.

## 2 Постановка задачи

На вход программе подается изображение с текстом. Необходимо определить наиболее похожие шрифты из выбранной базы эталонных шрифтов и упорядочить их по степени убывания похожести. Формально задача состоит из следующих шагов:

- Обработка изображения.
- Сегментация изображения с целью выделения глифов.
- Распознавание символа в каждом сегментированном фрагменте.
- Поиск похожих шрифтов для каждой классифицированной буквы.

---

\* <https://www.myfonts.com/WhatTheFont>

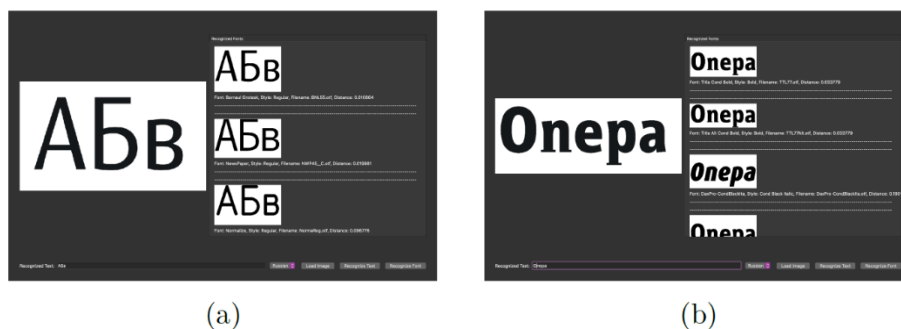
† <https://www.whatfontis.com/all-fonts.html>

‡ <https://www.fontsquirrel.com/matcherator>

§ <http://www.identifont.com/>

- Упорядочивание шрифтов по степени схожести на изображенный шрифт. Эта сортировка основана на информации о найденных шрифтах для каждой отдельной буквы.
- Создание графического интерфейса, позволяющего пользователю загрузить изображение и получить ранжированный список наиболее похожих шрифтов.

Методы обработки и сегментации осуществляются в соответствии с выбранным форматом входных данных, поэтому в данной статье они не описываются.



**Рис. 6.** Демонстрация разработанной системы распознавания шрифтов. В обоих случаях правильно распознаны символы и найдены похожие шрифты (в том числе и шрифт на изображении на 1-м месте выдачи)

### 3 Метод решения

Решение задачи распознавания шрифта по цифровому изображению основано на двух элементах: распознавание символов на изображении и распознавание шрифта на изображении известной буквы. Подход к распознаванию символов описан в [4, 12], а построение признаков, используемых в решении задачи распознавания шрифта, описано в [11].

Алгоритм распознавания символов основан на построении непрерывного скелетного графа для изображения буквы [10], выделении признаков по расположению вершин [12] в полученном графе и последующем обучении классификатора. Построение признаков, используемых в задаче распознавания шрифта, базируется на анализе гистограммы морфологической ширины [5, 11], которая показывает зависимость площади дискового покрытия фигуры от радиуса покрывающих дисков.

#### 3.1 Метод распознавания символов

В основе решения задачи распознавания символов на цифровом изображении лежит понятие математической модели графемы [4, 12] и метод выделения при-



знаков из нее для последующей возможности классификации символа при помощи алгоритмов машинного обучения.

**Графема** — это графический примитив, геометрический граф, универсальный образ буквы, понятный любому грамотному человеку и используемый в обучении грамоте. Он является основой любого шрифта и может быть представлен в виде контура букв в тонком шрифте (см. рис. 2). Эти свойства графемы используются для построения универсального алгоритма распознавания символов.

На содержательном уровне графемы обладают следующими свойствами:

5. Любые две графемы хорошо отличимы друг от друга.
6. Если два изображения представляются одной графемой, то различия между изображениями несутельственны (по структуре графемы).

Основной идеей для построения математической модели графемы является построение *непрерывного представления* скелетного графа бинарного изображения буквы и дальнейшее выделение существенного подграфа из него, отражающего свойства букв. Процесс построения можно увидеть на рис. 3.

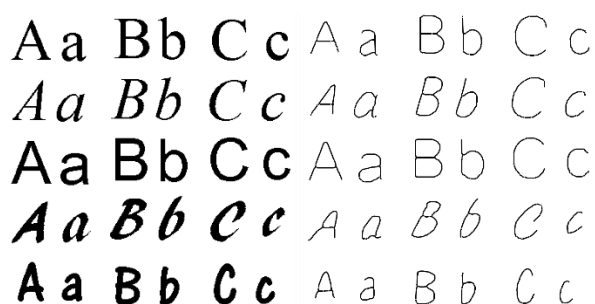


Рис. 7. Примеры букв в различных шрифтах (слева) и их графем (справа)



Рис. 8. Процесс построения графемы, слева направо: бинарное изображение символа; аппроксимирующий многоугольник; вписанные пустые круги фигуры; скелетный граф; граф с удаленными по некоторому принципу рёбрами; метаграф - математическая модель графемы

В конечном счете математическая модель графемы представляет собой метаграф  $S$ , вершины которого могут иметь только степени 1 и 3, а метаребра которого - «цепочки» последовательных ребер, в которых концевые вершины цепочек имеют степень 1 или 3, а остальные - степень 2. Далее из метаграфа выделяются признаки, которые впоследствии используются в классификаторе символов.

### 3.2 Метод распознавания шрифтов

В основе метода лежит понятие функции морфологической ширины фигуры. Быстрый и эффективный способ вычисления данной функции был предложен в [11]. Приведем необходимые определения и кратко опишем используемое понятие.

**Дисковое  $r$ -покрытие фигуры** - объединение всех пустых кругов радиуса  $r$ .  
 **$r$ -площадь фигуры** - площадь её дискового  $r$ -покрытия.

**Морфологическая ширина фигуры** - это  $r$ -площадь фигуры как функция от  $r$  (см. рис. 4).

Заметим, что морфологическая ширина является невозрастающей функцией от  $r$ . Примеры различных функций морфологической ширины и соответствующие им изображения букв можно увидеть на рис. 5. Также отметим, что в нашей постановке эта функция является дискретной.

#### Описание предлагаемого подхода

Сначала приведем несколько обозначений. Пусть  $DB$  - зафиксированная база шрифтов,  $H_{DB}$  - высота шрифта в базе,  $R_{DB}$  - шаг дискретизации (радиуса), с которым строятся все функции морфологической ширины в базе шрифтов,  $F(L)$  - высота шрифта, в котором изображена буква  $L$ ,  $H(L)$ ,  $W(L)$  - непосредственно высота и ширина ограничивающей рамки изображенной буквы,  $SQ(L)$  - количество черных пикселей на изображении буквы (площадь буквы). Также, пусть  $SP_r(L)$  - функция морфологической ширины с шагом радиуса  $r$  для буквы  $L$ ,  $SP_r(L)|_x$  - значение морфологической ширины в точке  $x$ ,  $R_{max}(L)$  - максимальный радиус пустого вписанного круга в фигуру буквы  $L$ ,  $Dom(R, r)$  - равномерная сетка на отрезке  $[0, R]$  с шагом радиуса  $r$ ,  $\#Dom(R, r)$  - число точек в сетке, аналогично  $Dom(SP_r(L))$  - носитель морфологической ширины,  $\#SP_r(L)$  - число точек носителя функции морфологической ширины.

В общем виде схема выглядит следующим образом:

- Выбираем базу шрифтов; для каждого шрифта и каждой буквы из фиксированного алфавита строим морфологическую ширину, полагая размер шрифта  $H_{DB}$  и шаг радиуса  $R_{DB}$  для всех случаев одинаковый (размер шрифта = размер заглавной буквы «Н»).
- Для тестового изображения символа вычисляем морфологическую ширину с некоторым шагом.
- Из всей базы эталонных шрифтов выделяем подмножество кандидатов  $E_F$  на звание похожих шрифтов.
- Сравниваем морфологическую ширину символа со всеми функциями морфологической ширины из  $E_F$  для этой буквы.
- Сравнение этих функций производится посредством метрики  $L_I$ .
- Все шрифты в базе упорядочиваются на основе значений линейной комбинации некоторых функций от метрик всех букв.



Рис. 9. Демонстрация дисковых  $r$ -покрытий фигур для разных значений  $r$

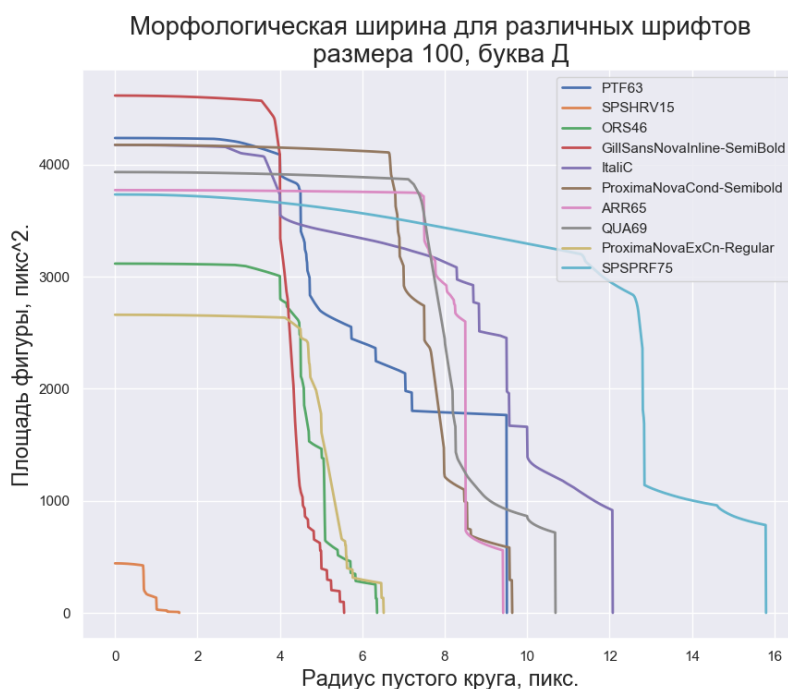


Рис. 10. Примеры морфологических ширин на равномерной сетке в различных шрифтах

### Выделение кандидатов близких шрифтов

Для больших баз шрифтов сравнение функций морфологической ширины тестового слова со всеми шрифтами из базы может быть дорогостоящей операцией: подсчет метрик имеет сложность  $O(NW)$ , где  $N$  - число шрифтов в базе,  $W$  - число распознанных букв в слове. Также для упорядочивания используется сортировка, работающая за  $O(N \log N)$ . Поэтому мы хотим среди всей базы оставить некоторое подмножество шрифтов-кандидатов, которые являются похожими на шрифт на картинке по некоторым статистическим и быстро вычисляемым признакам.

Для шрифтов нет универсальных стандартов пропорций изображенных в них букв. Количество черных пикселей (площадь буквы) при одной и той же высоте шрифта для разных типов шрифтов также различается. В качестве легко вычисляе-

мых статистических признаков шрифта мы выделяем 4 признака для изображения буквы  $L$  (1).

$$\left[ \frac{H(L)}{W(L)}, \frac{SQ(L)}{H(L) \cdot W(L)}, \frac{SQ(L)}{H(L)^2}, \frac{SQ(L)}{W(L)^2} \right]. \quad (5)$$

На этапе обучения для каждого символа алфавита из базы строятся вышеприведенные признаки, нормируются при помощи z-нормализации [6] и на полученных данных обучаются модели  $K$  ближайших соседей (k-NN) [3].

На этапе тестирования для изображения вычисляются и нормируются признаки и при помощи соответствующего букве классификатора k-NN ищутся  $K_F$  ближайших соседей, где  $K_F$  – заранее выбранное число. Далее, для каждого шрифта из полной базы определяется, оказался ли этот шрифт среди  $K_F$  хотя бы для  $p\%$  от числа букв в слове. Если это так, то шрифт добавляется в множество шрифтов-кандидатов. В конце концов алгоритм сравнения функций морфологической ширины будет работать только с множеством шрифтов-кандидатов.

### Сравнение функций морфологической ширины

При масштабировании одного и того же изображения в  $k$  раз, морфологические ширины изменяются в  $k$  раз по значениям их носителя и в  $k^2$  раз по значениям площадей покрытия. Для корректного сравнения данных функций необходимо, чтобы число точек, на которых задана эта функция, было одинаковым, а также чтобы совпадали размеры шрифтов на тестовом символе и в базе эталонных шрифтов. Пусть  $L$  - тестовая буква в некоторой высоте шрифта  $F(L)$ . Для удачного масштабирования морфологической ширины тестового изображения  $L$  необходимо знать размер шрифта  $F(L)$  тестового символа. Если мы знаем размер шрифта, то сможем рассчитать коэффициент масштабирования на основе размеров этих шрифтов  $s_{DB,L} = \frac{H_{DB}}{F(L)}$ , а также рассчитать сам шаг радиуса  $r'$  для построения  $SP_{r'}(L)$ :  $r' = \frac{R_{DB}}{s_{DB,L}}$ . Далее возможно корректное сравнение функций морфологической ширины посредством умножения на коэффициент масштабирования, описанный в начале секции.

Но на практике не всегда возможно узнать размер шрифта в тестовом изображении: например, если всё изображение состоит из одного символа, скажем, «с», то нельзя с уверенностью сказать, является ли это заглавной или строчной буквой. Поэтому на практике можно столкнуться с тем, что у морфологической ширины в базе эталонов и морфологической ширины тестового изображения будет разное число точек в носителе.

Из-за этого возникает отдельная задача построения морфологической ширины для тестового изображения буквы и ее корректного сравнения с аналогичными функциями из базы эталонных шрифтов. Для решения этой проблемы мы предлагаем следующий подход. В базе эталонов ищется максимальная морфологическая ширина по числу точек определения (2).

$$N_{max} := \max_{L \in DB} \#SP_{R_{DB}}(L). \quad (6)$$

Затем для тестового изображения символа  $L$  строим морфологическую ширину  $SP_r(L)$ , определенную в  $N_{max}$  равномерных точек на отрезке  $[0, R_{max}(L)]$  ( $r' = \frac{R_{max}(L)}{N_{max}-1}$ ). Далее сравниваем  $SP_r(L)$  с функцией морфологической ширины  $SP_{r_{DB}}(M)$  символа некоторого  $M$  из  $DB$ , определенной в  $K := \#SP_{r_{DB}}(M)$  равномерных точках. Для корректного сравнения мы будем линейно интерполировать  $SP_r(L)$  на  $K$  равномерных точек. Так как линейная интерполяция сохраняет монотонность, то полученное приближение также будет некоторой функцией морфологической ширины. После приведения к одному носителю, можно рассчитать коэффициент масштабирования площадей для  $SP_r(L)$  как  $s_{M,L} := \frac{H(M)}{H(L)}$ . Так как теперь количество точек в носителях совпадает, то перенумеруем их на  $Dom(K - 1, 1)$ . Расстояние между функциями считается по метрике  $L_1$  (3).

$$d(M, L) = \sum_{x \in Dom(K-1, 1)} (|SP_1(L)|_x - |SP_1(M)|_x). \quad (7)$$

Итоговое расстояние между шрифтом слова  $W$  и шрифтом из базы  $F$  считается по всем буквам  $L_i$  слова  $W$  по следующей формуле (4).

$$dist(W, F) = \frac{1}{|W|} \sum_{i=1}^{|W|} (1 - e^{-d(M_i, L_i)}), \quad (8)$$

где  $|W|$  – число букв в слове  $W$ , а  $M_i$  изображает ту же букву, что и  $L_i$ , но в шрифте  $F$ .

### Метрика качества

На этапе тестирования мы используем метрику качества  $Stop@k$  для различных значений  $k$ . Пусть есть упорядоченный по некоторому принципу список классов  $C = \{c_1, c_2, c_3, \dots, c_{|C|}\}$ , его длина  $|C|$ . Тогда метрика  $top@k$  для правильного ответа  $c'$  считается вычисляется по формуле (5).

$$top@k(c', C) = I[c' \in \{c_1, \dots, c_k\}]. \quad (9)$$

На тестовой выборке, состоящей из нескольких слов, метрика качества считается как усреднение всех  $top@k$  для входящих в нее слов.

## 4 Вычислительные эксперименты

### 4.1 Используемое программное обеспечение

Построение скелетного графа и функции морфологической ширины бинаризованного изображения реализовано на языке C++. Построение математической модели графемы, предлагаемый в работе алгоритм классификации и метод поиска наиболее схожих шрифтов были реализованы с помощью средств языка

Python3. Графический интерфейс системы реализован с помощью библиотеки QT\* на Python3.

#### 4.2 Эксперименты на качество системы распознавания шрифтов

В экспериментах мы тестируем качество распознавания шрифтов, используя предлагаемый метод. Эксперименты проводились на базе из 2543 шрифтов компании Paratype [13], среди которых система будет определять наиболее похожие шрифты, и упорядочивать их по степени схожести.

Модель распознавания символов обучается на 34 буквах кириллического алфавита («ы» считается за две графемы: «i» и «ь») в 2 начертаниях (строчное и прописное) из базы шрифтов. Размеры символов в тренировочном множестве берутся равными 120 и 70, изображения генерируются как в зашумленной (при помощи фильтра Гаусса с дисперсией 1), так и в не зашумленной форме, а количество классов в задаче классификации полагается равным 34.

Значения параметров полагаем следующими:

- $H_{DB} = 1000$  пикс;
- $R_{DB} = 0.1$  пикс;
- $K_F = 70$ ;
- $p = 50$ .

#### Сравнение качества для разных стратегий выбора шрифтов-кандидатов

В данном эксперименте мы сравниваем качество распознавания шрифтов при разных множествах шрифтов-кандидатов. Для создания тестовой выборки проводилась генерация примерно 500 случайных слов произведения Л.Н.Толстого «Война и мир», изображенных в 7 случайных шрифтах из доступной базы  $DB$  в размерах шрифта 100 пикселей.

Для случая, обозначаемого через «Native», наиболее похожие функции морфологической ширины искались по всей базе эталонных шрифтов  $DB$ . В случае, обозначаемом через «kNN» мы использовали поиск множества шрифтов-кандидатов при помощи алгоритма поиска ближайших соседей; в этом случае наиболее похожие шрифты искались только среди отобранных шрифтов. Пометка «All» означает, что качество считалось по всем изображениям в тестовой выборке. Пометка «Good» означает, что качество считалось по тем словам, истинная длина которых и предсказанная длина совпадали. Пометка «Right» означает, что качество считалось только по тем словам, на которых алгоритм распознавания не ошибся. Результаты можно видеть на рис. 6.

---

\* <https://www.qt.io/>

### Исследование попадания шрифтов в множество шрифтов-кандидатов на стадии использования kNN

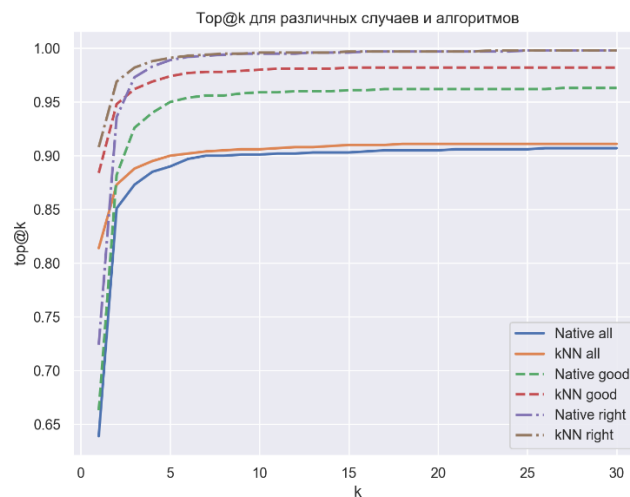
В данном эксперименте исследуется процент попадания правильного шрифта во множество шрифтов-кандидатов, которых не более  $K_f=70$  шрифтов. Исследование проводилось на той же выборке, что и в предыдущем эксперименте. Результаты можно увидеть в таблице 1.

**Таблица 3.** Точность попадания правильного шрифта в отобранное множество шрифтов-кандидатов

	kNN All	kNN Good	kNN Right
Точность	0.907	0.987	0.999

### Исследование зависимости качества от размера шрифта и качества тестовой выборки

В данном эксперименте проводится исследование зависимости качества от размера шрифта в тестовой выборке и качества тестовой выборки. Генерация тестовой выборки производится так же, как и ранее. Всего создается 3 тестовых выборки. Одна из них в размере шрифта 100 и не зашумлённая («100 pure»), другая - в размере шрифта 150 и не зашумленная («150 pure») и третья в размере шрифта 150 и зашумленная («150 spoil»). Результаты находятся на рис. 7.



**Рис. 11.** Визуализация метрик  $top@k$  для  $k=1..30$  для различных стратегий выбора наиболее похожих шрифтов

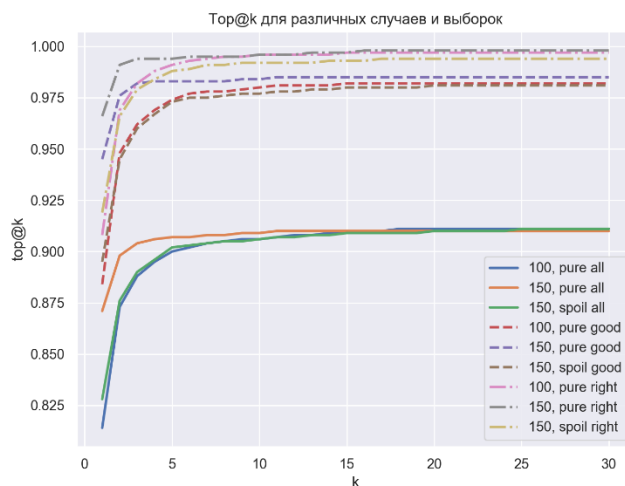


Рис. 12. Визуализация метрик  $top@k$  для  $k=1..30$  для различных высот шрифтов в выборках и различных качеств изображений

## 5 Выводы

Эксперименты подтвердили работоспособность предлагаемого в данной работе метода: для  $top@5$  уже достигаются качества 0.9, 0.974, 0.991 для всех шрифтов, для шрифтов, в которых распознанное слово по длине равно слову-ответу, для шрифтов, в которых алгоритм распознавания текста выдал точный ответ соответственно. Также из экспериментов видно, что метод выделения шрифтов-кандидатов значительно улучшает качество. Хотя и текущее значение  $K_F=70$  не является оптимальным, подбор наилучшего с точки зрения метрик значения  $K_F$  открывает возможности для будущих исследований в области улучшений работы метода. Алгоритм является достаточно устойчивым к размеру шрифта на изображении и качеству изображения: по метрике  $top@10$  метод показывает примерно одинаковое качество распознавания в перечисленных случаях, однако по  $top@1$  качество распознавания в большем размере шрифта (150) существенно выше, чем в меньшем (100).

## 6 Заключение

В данной статье был предложен метод распознавания шрифтов по цифровым изображениям, показывающей высокие результаты распознавания для отдельных классов шрифтов. Метод основан на вычислении функции морфологической ширины в контексте задачи поиска наиболее похожих шрифтов и способе сравнения данных функций. Также была разработана система распознавания



шрифтов с графическим интерфейсом. Эксперименты показали высокие результаты качества распознавания шрифтов на изображении.

Работа выполнена при финансовой поддержке гранта РФФИ, проект 20-01-00664 «Морфологический анализ изображений и видеопоследовательностей на основе непрерывного медиального представления и машинного обучения».

## Литература

1. Bui, T., Collomosse, J.: Font finder: Visual recognition of typeface in printed documents. In: 2015 IEEE International Conference on Image Processing (ICIP). pp. 3926–3930. IEEE (2015) Author, F., Author, S.: Title of a proceedings paper. In: Editor, F., Editor, S. (eds.) CONFERENCE 2016, LNCS, vol. 9999, pp. 1–13. Springer, Heidelberg (2016).
2. Chen, G., Yang, J., Jin, H., Brandt, J., Shechtman, E., Agarwala, A., Han, T.X.: Large-scale visual font recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3598–3605 (2014).
3. Cover, T., Hart, P.: Nearest neighbor pattern classification. IEEE transactions on information theory 13(1), 21–27 (1967).
4. Lipkina, A., Mestetskiy, L.: Grapheme approach to recognizing letters based on medial representation. In: Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2019). vol. 4, pp. 351–358. Setubal, Portugal (2019).
5. Maragos, P.: Pattern spectrum and multiscale shape representation. IEEE Trans. Pattern Anal. Mach. Intell. 11, 701–716 (1989).
6. Maxfield, M., Schweitzer, J., Gouvier, W.: Measures of central tendency, variability, and relative standing in nonnormal distributions: alternatives to the mean and standard score. Archives of physical medicine and rehabilitation 69(6), 406–409 (1988).
7. Wang, Z., Yang, J., Jin, H., Shechtman, E., Agarwala, A., Brandt, J., Huang, T.S.: Deepfont: Identify your font from an image. In: Proceedings of the 23<sup>rd</sup> ACM international conference on Multimedia. pp. 451–459 (2015).
8. Zhu, Y., Tan, T., Wang, Y.: Font recognition based on global texture analysis. IEEE Transactions on pattern analysis and machine intelligence 23(10), 1192–1200 (2001).
9. Zramdini, A., Ingold, R.: Optical font recognition using typographical features. IEEE Transactions on pattern analysis and machine intelligence 20(8), 877–882 (1998).
10. Местецкий Л.М.. Непрерывная морфология бинарных изображений: фигуры, скелеты, циркуляры. М.: ФИЗМАТЛИТ (2009).
11. Липкина А., Ломов Н., Местецкий Л.. Распознавание шрифтов по изображениям на основе анализа структуры букв // ГрафиКон-2017 Труды 27-й Международной научной конференции. Пермский государственный национальный исследовательский университет. С. 359-366. (2017).
12. Липкина А., Местецкий Л.. Структурный подход к классификации букв в изображениях // ГрафиКон-2019 Труды 29-й Международной конференции по компьютерной графике и машинному зрению. Брянский государственный технический университет. С. 136-141. (2019).
13. Якупов Э., Петрова И., Фридман Г., Королькова А., Левин Б.. 2008-2014. ПАРАТАЙП. Цифровые шрифты. Каталог. Москва: ParaType (2014).



**ВЫЧИСЛИТЕЛЬНАЯ ОПТИКА И  
ОПТИЧЕСКИЙ ДИЗАЙН**

---

**COMPUTATIONAL OPTICS AND  
OPTICAL DESIGN**

**GraphiCon 2020**

---

22-25 сентября 2020  
Санкт-Петербург, Россия

## Возможности цифрового контроля и классификации зерновых культур

Д.Е. Трошкин<sup>1</sup>, А.Н. Чертов<sup>1</sup>, Е.В. Горбунова<sup>1</sup>, А.А. Алёхин<sup>1</sup>

<sup>1</sup> Университет ИТМО, 197101, Санкт-Петербург, Россия

**Аннотация.** Представлены результаты исследований по определению параметров качества зерна пшеницы и пивоваренного ячменя методом компьютерного зрения, подтверждающие возможность перехода от субъективного визуального анализа к цифровому контролю.

**Ключевые слова:** компьютерное зрение, зерновые культуры, контроль качества, пшеница, ячмень, стекловидность.

### 1 Введение

Обеспечение продовольственной безопасности - одна из наиболее актуальных глобальных задач, стоящих на сегодняшний день перед человечеством. И одним из ее существенных элементов является контроль безопасности и качества пищевого сырья и готовых продуктов питания.

Основой рациона питания современного человека являются зерновые культуры и продукты их переработки, поэтому технологические процедуры анализа и сортировки зерна по показателям качества крайне важны.

Все методы, используемые для определения показателей качества зерна, можно условно разделить на две группы: физико-химические и органолептический контроль.

Физико-химические методы используются для определения широкого спектра показателей качества, таких как содержание белка и крахмала, качество клейковины, зольность и др. Данные методы отличаются достаточно высокой точностью, но часто требуют значительного времени на проведение экспертизы и наличия химических реагентов. Кроме того, объект исследования в большинстве случаев разрушается, что делает невозможными арбитражные процедуры. Альтернативой является использование автоматизированных аналитических комплексов, однако они дают усредненную информацию по пробе или партии, не позволяя оценить диапазон изменений показателей качества зерна, а значит и принять объективное и эффективное решение о его дальнейшем использовании.

Органолептические методы применяются для определения внешнего вида, вкуса, запаха. Как понятно из названия, эти методы опираются на сенсорные ощущения человека, что автоматически делает их субъективными, а значит ненадежными, поскольку восприятие человека зависит как от физического и морально-психологического состояния эксперта на момент проведения анализа,

так и от внешних условий контроля. Данное утверждение в полной мере касается и работы зрительного аппарата человека. Между тем, по визуальным критериям анализируются такие показатели качества зерновых культур как цвет, стекловидность, размеры и форма, засоренность и даже зараженность [1].

Цвет зерна связан с его генетическими особенностями и указывает на сорт и тип. Изменение цвета может свидетельствовать о наличии механических повреждений, несоблюдении условий хранения и переработки, а также на зараженность зерна различными вредителями и патогенами [2]. При этом зерно является сложным объектом для визуального анализа ввиду малых размеров, неоднородности формы, внешнего и внутреннего строения, а также неоднородной и часто не ярко выраженной окраски поверхности.

Стекловидность – один из наиболее интересных и важных показателей качества. Зерно с плотной упаковкой белковых зерен лучше пропускает свет, чем зерно с рыхлой структурой, содержащее большее количество крахмала. Таким образом, данный показатель качества связывает способность зерна пропускать оптическое излучение с особенностями его внутренней структуры и пищевой ценностью. Традиционно стекловидность определяется на специальных приборах - диафаноскопах. Анализ стекловидности сводится к подсчету оператором стекловидных и частично стекловидных зерен, лучше пропускающих свет, чем мучнистые. Другим методом определения данного показателя качества является метод визуального осмотра поперечного среза зерна [3]. Разумеется, в обоих случаях полученные результаты критически зависят от человеческого фактора.

Несмотря на очевидные недостатки, визуальный метод контроля, наряду с другими органолептическими методами, прочно закреплен в национальных и межгосударственных стандартах на зерновые культуры. Безусловно, данный факт препятствует развитию зерновой отрасли российского АПК и принятой Национальной программе «Цифровая экономика Российской Федерации».

Современный уровень развития информационных и оптических технологий позволяет решить задачи объективного контроля качества зерновых культур и в перспективе заместить до сих пор широко используемые методы визуального контроля технологиями компьютерного зрения.

Еще одним существенным преимуществом объективного контроля является возможность выявления корреляционных зависимостей между физико-техническими показателями качества зерновых культур и характеризующими их информационными параметрами, а также и между различными показателями качества. Прогнозные модели, построенные на подобных зависимостях, в свою очередь, позволяют оптимизировать технологические процессы сбора, подработки, хранения, переработки и использования зерна.

В материалах данной работы обобщены результаты исследований коллектива авторов, проведенных в 2019-2020 гг. на зерне пшеницы и пивоваренного ячменя. В первую очередь, исследованию подлежал показатель качества «стекловидность» и принципы обеспечения его объективного контроля для единичного зерна. Подобный фокус исследования был обусловлен, в том числе, наличием подтвержденной корреляционной связи стекловидности с содержанием в зерне

белка и качеством клейковины [4, 5], а значит, и возможностями более точного их определения.

## 2 Материалы и методы решения задач

Для проведения экспериментальных исследований использовались образцы пшеницы, предоставленные Всероссийским Институтом Растениеводства им. Н.И. Вавилова (г. Санкт-Петербург), а также образцы пивоваренного ячменя, предоставленные ООО «Ностерс» (г. Санкт-Петербург). Пробы зерна отбирались в соответствии с ГОСТ 13586.3-2015 [6].

Для проведения экспериментальных исследований использовалось устройство собственной разработки, вид в разрезе которого представлен на рис. 1. Оно содержит два источника, верхний и нижний, излучающие в видимом диапазоне длин волн, рассеивающую пластину, специальную кассету на 100 ячеек для размещения образцов зерна и телевизионную камеру с объективом, передающую цифровые изображения зерен на персональный компьютер для дальнейшей обработки.



**Рис. 1.** Аппаратно-программный комплекс: 1 – верхний источник излучения, 2 – камера с объективом, 3 – кассета для размещения образцов зерна, 4 – светорассеивающая пластина, 5 – нижний источник излучения

Для обеспечения объективности результатов анализа перед проведением измерений проводились колориметрическая, фотометрическая и метрическая калибровки.

Измерения проводились следующим образом. Анализируемые зерна пшеницы и ячменя помещались в специальную кассету на 100 ячеек, как показано на рис. 2. Кассета с образцами помещалась в зону анализа, производилась съемка зерен в режиме «на пропускание» с использованием нижнего источника излучения.

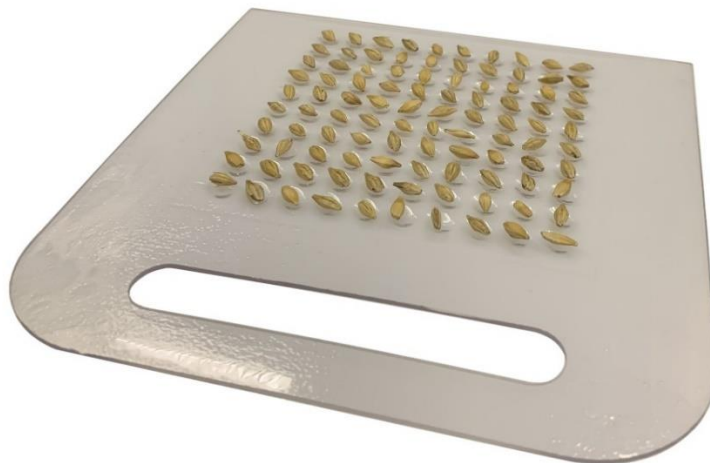


Рис. 2. Размещение исследуемых образцов в кассете

Полученные изображения обрабатывались алгоритмом, реализованным в среде MATLAB. Алгоритм обработки цифровых изображений зерен осуществлял предобработку (устранение шумов, неравномерности облученности зоны анализа, увеличение контрастности), выполнял сегментацию. В результате данной процедуры находили области изображения, соответствующие зерновкам.

Далее выполнялся расчет общей стекловидности пробы. Анализ стекловидности предполагает разделение зерна на три группы: стекловидное, частично-стекловидное и мучнистое. Для этого рассчитывался показатель стекловидности каждой зерновки, полученные значения сравнивались с пороговыми значениями для каждой из трех групп, определенными эмпирически, методом осмотра поперечного разреза [3]. По результатам сравнения каждая зерновка относилась к одной из указанных групп.

Расчет показателя стекловидности выполнялся с использованием областей, полученных при сегментации цифровых изображений зерен. В каждой из полученных областей оригинального изображения, переведенного в градации серого, показатель стекловидности каждой зерновки рассчитывался как:

$$V_i = \frac{\sum_{n=1}^N Int_n}{N}, \quad (1)$$

где  $V_i$  – показатель стекловидности  $i$ -го объекта,  $Int_n$  – значение пикселя изображения, принадлежащее объекту,  $N$  – количество пикселей на цифровом изображении, принадлежащих данной зерновке.

После разделения всех зерновок по трем группам рассчитывалась общая стекловидность пробы зерна по формуле (2) из ГОСТ 10987-76 [3]:

$$O = F + \frac{P}{2}, \quad (2)$$

где  $O$  – общая стекловидность пробы,  $F$  – количество стекловидных зерен,  $P$  – количество частично-стекловидных зерен.

### 3 Результаты экспериментальных исследований

#### 3.1 Особенности цифрового анализа стекловидности зерна пшеницы

##### Влияние плотности размещения зерен на результаты анализа

Первоначально предполагалось, что, помимо оговоренной субъективности, единственным недостатком стандартного метода анализа стекловидности с использованием диафаноскопа является обязательное размещение исследуемых образцов в кассете на сто отдельных ячеек. Для проведения исследований стекловидности без разделения зерен был реализован алгоритм определения стекловидности пробы пшеницы по цифровым изображениям зерен с функцией сегментации. Для проверки работоспособности алгоритма была проведена серия из 10 измерений пробы краснозерной пшеницы сорта Лютеценс 275.

Измерения проводились при постепенном увеличении плотности размещения зерен (рис. 3).



**Рис. 3.** Изменение плотности размещения зерен: измерение №1 – слева и измерение № 10 – справа

Результаты определения общей стекловидности пробы при различной кучности расположения зерен представлены в таблице 1.

**Таблица 1.** Результаты расчета общей стекловидности пробы пшеницы при различной кучности расположения зерен в зоне анализа

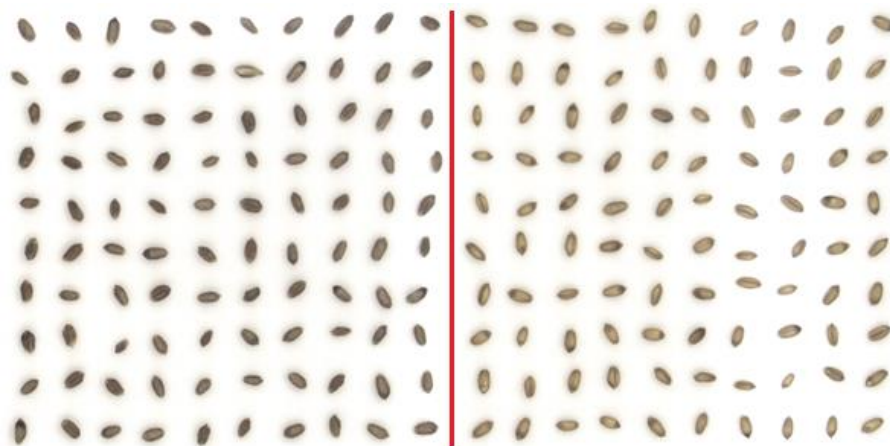
№	Кол-во стекловидных зерен, шт	Кол-во частично-стекловидных зерен, шт	Кол-во мучнистых зерен, шт	Общая стекл-ть, %
1	52	39	9	72
2	51	37	12	70
3	46	39	15	66
4	37	39	24	57
5	33	44	23	55
6	41	36	23	59
7	30	46	24	53
8	41	44	15	63
9	37	40	23	57
10	19	55	26	47

Как видно из таблицы 1, при увеличении плотности размещения зерен в зоне анализа происходит снижение интенсивности оптического излучения, проходящего через них. Данное снижение интенсивности происходит вследствие уменьшения светового потока, достигающего поверхности зерна в результате его частичного поглощения соседними зернами. Поскольку достоверность полученных результатов зависит от плотности размещения, при анализе стекловидности без кассеты в видимом диапазоне длин волн необходимо обеспечить достаточное расстояние между зернами.

#### **Влияние цвета зерен на результаты анализа стекловидности**

При анализе стекловидности в пробе пшеницы могут оказаться зерна разных сортов, красные и белые. При этом в явном виде в ГОСТ не указывается, стоит ли определять стекловидность для красного и белого зерна отдельно. По всей видимости, причиной является отсутствие объективной методики неразрушающего контроля цвета зерна. Тем не менее, было решено провести соответствующее исследование с использованием проб пшеницы двух разных сортов: Лютеценс 275 (стекловидная, красnozерная) и HD 1639 (мучнистая, белозерная). Цветные изображения проб представлены на рис. 4.





**Рис. 4.** Краснозерная и белозерная пшеницы: Лютесценс 275 (слева) и HD 1639 (справа)

После проведения измерений стекловидность проб определялась методом осмотра поперечного среза. Значения общей стекловидности, полученные при помощи алгоритма и по стандартной методике представлены в таблице 2.

**Таблица 2.** Результаты расчета общей стекловидности пробы пшеницы методом разрезания и при помощи алгоритма

Сорт	Цвет	Общая стекловидность по ГОСТ, %	Общая стекловидность, рассчитанная алгоритмом, %
Лютесценс-275	красный	86	80
HD 1639	белый	14	88

Из таблицы 2 видно, что при расчете общей стекловидности проб пшеницы разработанным авторами алгоритмом получили завышенное значение для белозерной пшеницы сорта HD 1639. Данный эффект возник вследствие наличия красного пигмента в плодовых оболочках краснозерной пшеницы, обладающего большим поглощением в видимой области, и отсутствия данного пигмента у образцов белозерного сорта.

Поскольку в партиях зерна практически всегда присутствует смесь обоих типов (красной и белой) пшеницы, результаты оценки стекловидности, получаемые в видимом диапазоне длин волн, не надежны, и могут давать большую ошибку.

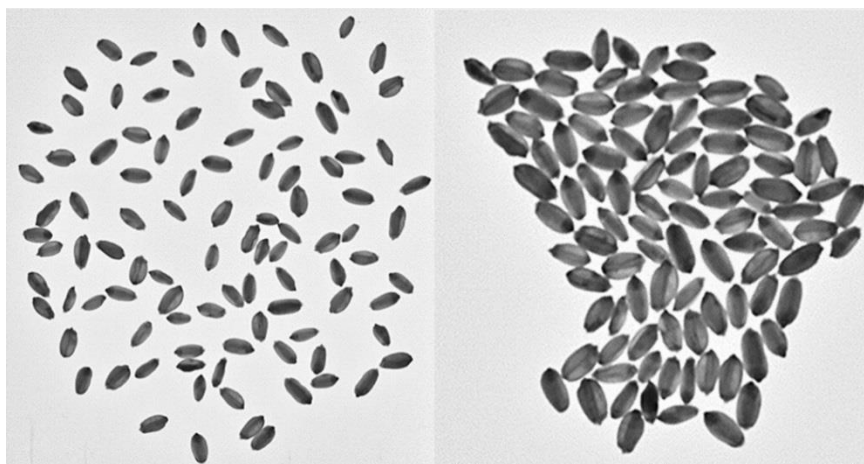
#### **Использование источника излучения ближнего ИК-диапазона для анализа стекловидности.**

В качестве решения проблем влияния плотности размещения и цвета зерна было предложено использовать источник излучения ближнего ИК-диапазона в модуле нижней подсветки. При этом были продублированы эксперименты, описанные выше.

Данные по исследованию влияния плотности размещения представлены в таблице 3. Визуализация опытов с максимальной и минимальной плотностью размещения зерен краснозерной пшеницы в зоне анализа представлена на рисунке 5.

**Таблица 3.** Результаты расчета общей стекловидности пробы краснозерной пшеницы в ИК-диапазоне

№	Стекл-х зерен, шт	Частично-стекл-х, шт	Мучнистых, шт	Общ. стекл., %
1	82	12	6	88
2	86	10	4	91
3	83	9	8	87,5
4	84	11	5	89,5
5	80	14	6	87
6	80	15	5	87,5
7	84	10	6	89
8	84	9	7	88,5
9	85	8	7	89
10	83	14	3	90



**Рис. 5.** Изменение плотности размещения зерен при съемке в ИК-диапазоне

Для оценки повторяемости результатов работы алгоритма определения общей стекловидности было рассчитано среднее квадратичное отклонение (СКО), полученное значение составило 1.1. Максимальный разброс значений, полученных в ходе эксперимента, составил 4%, что соответствует требованиям ГОСТ [3].

При определении влияния цвета зерна на показатель стекловидности исследование проводилось отдельно для красной и белой пшениц в серии из 5 повторных опытов. Результаты представлены в таблицах 4 – 5.

**Таблица 4.** Результаты расчета общей стекловидности пробы краснозерной пшеницы в ИК-диапазоне

№	Кол-во стекловидных зерен, шт	Кол-во частично-стекловидных зерен, шт	Кол-во мучнистых зерен, шт	Общая стекловидность, %
1	82	12	6	88
2	86	10	4	91
3	83	9	8	87,5
4	84	11	5	89,5
5	80	14	6	87
СКО	0,37			

**Таблица 5.** Результаты расчета общей стекловидности пробы белозерной пшеницы в ИК-диапазоне

№	Кол-во стекловидных зерен, шт	Кол-во частично-стекловидных зерен, шт	Кол-во мучнистых зерен, шт	Общая стекловидность, %
1	12	10	78	17
2	14	8	78	18
3	13	7	80	16,5
4	13	6	81	16
5	12	11	77	18,5
СКО	0,16			

Как видно из таблиц 4-5, для стекловидного (красного) и мучнистого (белого) сортов пшеницы получили адекватные значения общей стекловидности, соответствующие действительности. Кроме того, разброс полученных значений общей стекловидности исследуемых проб при проведении повторных измерений составил менее 5%, что соответствует требованиям ГОСТ 10987-76 [3]. Таким образом, можно сделать вывод, что использование источника ИК-излучения в модуле нижней подсветки позволяет практически полностью исключить влияние плотности размещения образцов и их цвета на результаты анализа стекловидности.

### 3.2 Использование технологий компьютерного зрения для анализа качества пивоваренного ячменя

Определение стекловидности пивоваренного ячменя в настоящее время не регламентируется государственными стандартами. Для решения этой задачи используется ГОСТ 10987-76 [3] для определения стекловидности пшеницы и риса. Между тем, данный показатель широко применяется в работе пивоварен и

солодовен, так как позволяет быстро оценить качество исходного сырья для солодоращения [7]. Данный показатель также используется при оценке friability и выравненности пробы, которые, в свою очередь, связаны со способностью к прорастанию: более плотное стекловидное зерно прорастает медленнее мучнистого, поэтому для изготовления качественного сула нужно, чтобы зерна были примерно одинаковы по стекловидности – так они прорастут примерно в одинаковое время и сопутствующие физико-химические процессы, такие, как например, превращение крахмала в простые сахара, пройдут максимально полно и эффективно. Почему же стекловидность ячменя не включена в указанный ГОСТ? Дело в том, что ячмень отличается наличием плотных плодовых оболочек, сильно поглощающих любое падающее на них излучение.

Таким образом, при анализе на диафаноскопе разницы между мучнистыми и стекловидными зернами практически не будет видно невооруженным глазом. Но при использовании систем компьютерного зрения возможно увеличить экспозицию, тем самым расширив динамический диапазон. При проведении соответствующего анализа по методике, описанной ранее, можно получить численные значения показателя стекловидности для каждого из зерен (рис. 6).

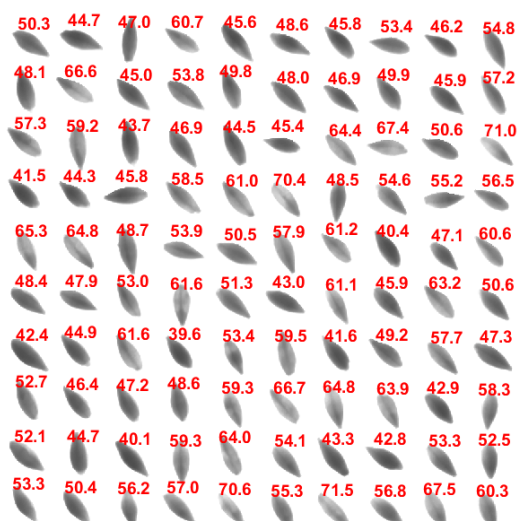


Рис. 6. Показатели стекловидности для пробы ячменя

Таким образом, становится возможным определение единых пороговых значений по показателю стекловидности для разделения зерен ячменя на стекловидные, частично-стекловидные и мучнистые, по аналогии с тем, как это сделано для зерна пшеницы [8].

Кроме того, совокупность полученных данных по одной или нескольким пробам может быть использована для получения распределений зерен по показателю стекловидности. Пример подобной гистограммы по пробе, состоящей из 100 зерен, приведен на рис. 7.

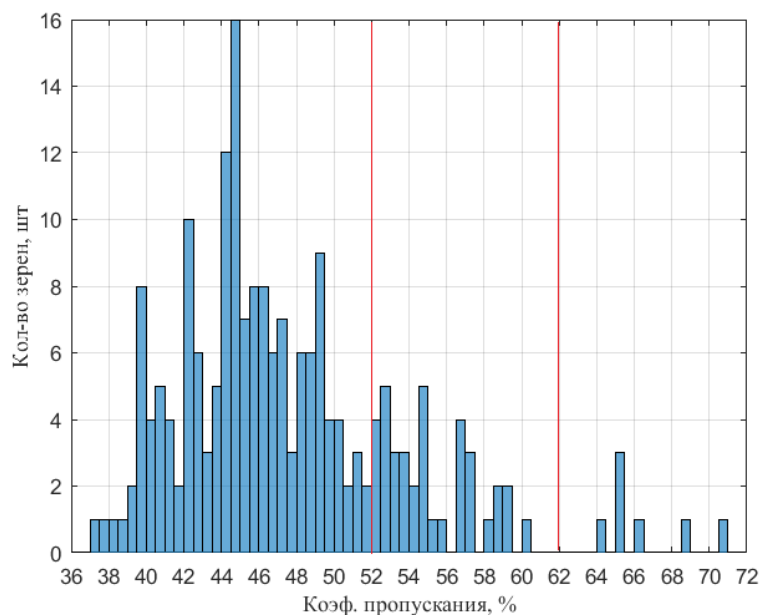


Рис. 7. Пример распределения зерен пробы ячменя по показателю стекловидности

Такая гистограмма очень полезна для оценки выравненности пробы и позволяет определять качество уже не только в усредненном, выраженном одним числом, виде, но и более детально. В частности, данный инструмент может быть полезен для выявления недобросовестного подмешивания в партию зерна низкого качества.

### 3.3 Иные показатели, определяемые методом компьютерного зрения, перспективы использования

Компьютерное зрение, безусловно, может быть полезно и для определения параметров размера и формы. Наличие метрической калибровки позволяет определять размеры зерен с высокой точностью. В устройстве, описанном в разделе 2, точность определения размера составляет 0,001 мм. А зная длину и ширину зерна, возможно определить и его индекс формы, а также и выравненность пробы по данному показателю (рис. 8).

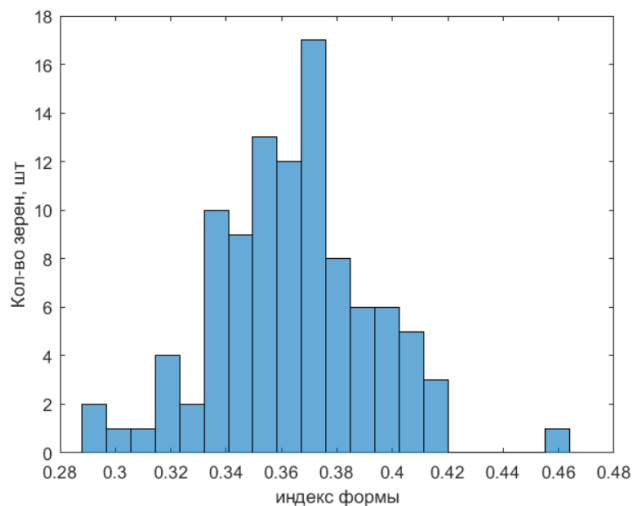


Рис. 8. Пример распределения зерен пробы ячменя по индексу формы

Метрическая информация может быть использована при определении процента мелких зерен в партии, поврежденных зерен, наличия примесей, а также при решении задач селекции и генетики.

#### 4 Заключение

Описанные результаты исследований, полученные коллективом авторов, являются основой для цифровизации контроля показателей качества зерновых культур на принципах обеспечения единства измерений в соответствии с метрологическими стандартами.

Научная новизна проведенных исследований заключается:

- в предложенном принципе неразрушающей объективной оценки стекловидности каждого зерна пробы на примере пшеницы и пивоваренного ячменя;
- в выявленных зависимостях результата определения стекловидности пробы зерна методом компьютерного зрения от плотности расположения зерен в зоне анализа и цвета зерна;
- во впервые предложенном подходе к оценке выравненности пробы зерна по стекловидности и параметрам размера и формы.

Следует отметить практическую важность полученных результатов не только для задач лабораторного контроля, но и для промышленной сортировки, т.к. при наличии объективных критериев классификации зерна по качеству появляется возможность и для решения на их основе задач целевой подработки уже очищенного от сорных примесей зерна.

В настоящее время коллектив авторов проводит исследования с целью установления устойчивых корреляционных зависимостей между стекловидностью зерна и другими показателями качества, в том числе, содержанием белка в отдельном зерне, фриабильностью и стекловидностью пивоваренного солода и др.

## Литература

1. ГОСТ 9353-2016 Пшеница. Технические условия
2. Комышев Е., Генаев М., Афонников Д.: Анализ цветовых и текстурных характеристик зерен злаков на цифровых изображениях // Вавиловский журнал генетики и селекции 2020, vol 24(4), pp. 340-347, doi:10.18699/VJ20.626
3. ГОСТ 10987-76. Зерно. Методы определения стекловидности.
4. Bin Xiao Fu, Kun Wang, Brigitte Dupuis, Dale Taylor, Shin Nam. Kernel vitreousness and protein content: Relationship, interaction and synergistic effects on durum wheat quality // Journal of Cereal Science. 2018. Vol. 79. PP. 210-217. doi: 10.1016/j.jcs.2017.09.003.
5. Alisa-N. Sieber, Tobias Würschum, C. Friedrich H. Longin. Vitreosity, its stability and relationship to protein content in durum wheat // Journal of Cereal Science. 2015. Vol. 61. PP. 71-77. doi: 10.1016/j.jcs.2014.10.008.
6. ГОСТ 13586.3-2015. Зерно. Правила приемки и методы отбора проб
7. Н. И. Васько, М. Р. Козаченко, П. Н. Солонечный, О. В. Солонечная, О. Е. Важенина, А. Г. Наумов, А. В. Зимогляд, Т. А. Шелякина Стекловидность эндосперма и содержание белка в зерне сортов пленчатого и голозерного ячменя // Зернобобовые и крупяные культуры. 2018. №4 (28).
8. Трошкин Д.Е., Горбунова Е.В., Алёхин А.А., Горбачёв А.А., Чертов А.Н., Лоскутов И.Г., Зуев Е.В. Количественная оценка стекловидности пшеницы методом технического зрения // Хлебопродукты -2019. - № 6.- С. 52-56.

## Полигармоническое разложение цифрового изображения

А.Н. Марковский<sup>1</sup>

<sup>1</sup> Кубанский государственный университет, 350040, Краснодар, Россия  
mrkvsk@yandex.ru

**Аннотация.** Получено разложение пространства  $L_2(Q)$  в прямую сумму полигармонических подпространств. Доказана полнота систем сдвигов фундаментального решения полигармонического уравнения. Разработан сходящийся алгоритм решения задачи выделения полигармонической составляющей функции из  $L_2(Q)$ . Полученное разложение применяется к некоторым задачам обработки цифровых изображений; приводятся результаты вычислительных экспериментов.

**Ключевые слова:** полигармонические функции, ортогональная сумма, цифровые изображения.

### 1 Введение

В обработке изображений важную роль играет декомпозиция изображения в сумму элементарных изображений [1]. Особо важно, чтобы такое разложение было единственно, и элементарные изображения были в определенном смысле независимы. Такой подход позволяет свести анализ исходного изображения к анализу элементарных изображений.

Если изображение моделировать кусочно-постоянной функцией из пространства Лебега  $L_2(Q)$  ( $Q=[0,1]^2$  - поле зрения,  $x=(x_1, x_2) \in R^2$ ), то одним из примеров такого разложения может служить ряд Фурье (1) - разложение по некоторой полной ортогональной системе  $\varphi_{ij}$ ,  $i, j = 1, 2, \dots$

$$f(x) = \sum_{i=1}^{\infty} \sum_{j=1}^{\infty} c_{ij} \varphi_{ij}(x), \quad (3)$$

где  $c_{ij} = (f, \varphi_{ij})$  - коэффициенты Фурье, при этом базисные функции  $\varphi_{ij}$  называют гармониками, а набор коэффициентов  $c_{ij}$  - спектром.

Полных ортогональных систем существует бесконечно много, поэтому для разных классов изображений стараются использовать такие базисные системы, которые наилучшим образом отражают свойства класса.

Описание классов изображений и соответствующих систем, а также разработка новых методов численного анализа и обработки изображений является весьма актуальной задачей.

В настоящей работе приводится новое общее ортогональное полигармоническое разложение (2) для любой функции  $f$  из пространства  $L_2(Q)$ .



$$f(x) = \sum_{k=1}^{\infty} a_k(x), \quad (4)$$

где  $a_k(x)$  - полигармонические компоненты.

На основе этого разложения определяется класс гладких изображений и на этом классе вводится новая метрика. Для анализа изображений используются свойства и методы полигармонических функций.

## 2 Ортогональная сумма полигармонических подпространств

Рассмотрим однородное полигармоническое уравнение (3).

$$\Delta^m u(x) = 0, \Delta^m = \Delta(\Delta^{m-1}), m \in N, \quad (5)$$

в ограниченной области  $Q \subset R^n$  с кусочно-гладкой границей  $S = \partial Q$ . Введем обозначение (4).

$$e_m(Q) = \{E_{m,n}(x - y) | x \in Q, y \in R^n \setminus \bar{Q}\}, m \geq 1, \quad (6)$$

множество сдвигов фундаментального решения  $m$ -гармонического уравнения в  $n$ -мерном пространстве  $R^n$ , и  $G_m(Q)$  - замыкание  $e_m(Q)$  в норме  $L_2(Q)$ . Подпространство  $G_m(Q)$  будем называть *полигармоническим пространством*.

Обозначим  $N_m(Q)$  - ортогональное дополнение к  $G_m(Q)$  в  $L_2(Q)$ , ( $m \geq 1$ ), так что  $L_2(Q) = G_l(Q) \oplus N_l(Q)$ . Далее, рассматриваем расширение по непрерывности классического оператора Лапласа и корректно строим обратный оператор (5).

$$\Delta^{-1}: L_2(Q) \rightarrow N_1(Q). \quad (7)$$

**Теорема 1.** Если  $f \in G_m(Q)$ , ( $m \geq 1$ ), то существуют гармонические из  $G_l(Q)$  функции  $g_0, g_1, \dots, g_{m-1}$ , такие что

$$f = g_0 + \Delta^{-1}g_1 + \dots + \Delta^{-(m-1)}g_{m-1}, \quad (8)$$

и такое представление единственно.

Обозначим  $A_l(Q) = G_l(Q)$  - подпространство гармонических функций, и

$$A_k(Q) = \{f \in N_{k-1}(Q) | f = \Delta^{-(k-1)}g, g \in G_1(Q)\}, k \geq 2.$$

**Теорема 2.** Пространство функций, суммируемых по области с квадратом, разлагается в ортогональную сумму (7).

$$L_2(Q) = \bigoplus_{k=1}^{\infty} A_k(Q). \quad (9)$$

### 3 Метод фундаментальных решений

Системы сдвигов фундаментальных решений различных уравнений математической физики рассматривались в связи с построением проекционного метода решения соответствующих краевых задач [2]. Такой подход в дальнейшем назывался методом разложения по неортогональным функциям [3], методом базисных потенциалов [4] и, в зарубежной литературе, методом фундаментальных решений [5].

Всякая такая система порождается сдвигами аргументов одной функции -- фундаментального решения дифференциального оператора или некоторой его модификации.

Сходимость алгоритмов метода обеспечивается полнотой рассматриваемых систем; получение достаточных условий полноты - условий на множество сдвигов, является одной из основных задач теоретического обоснования использования тех или иных фундаментальных решений.

Достаточное условие полноты гармонической системы – условие базисности – сформулировал проф. В.Г. Лежнев в работе [4] и заметил, что сдвиги можно выбирать локализовано.

Последовательность точек

$$x^{(k)} \in Q^+ = R^n \setminus \bar{Q}, k = 1, 2, \dots,$$

будем называть *m-базисной*, если она отделена от границы  $S$  и удовлетворяет условию единственности  $m$ -гармонических (полигармонических) в  $Q^+$  функций.

Обозначим

$$\gamma_{mk}(x) = E_{m,n}(x^{(k)} - x), x \in Q, k = 1, 2, \dots,$$

семейство ( $m=1, 2, \dots$ ) систем функций, где  $E_{m,n}(x)$  - фундаментальное решение  $m$ -гармонического уравнения в  $R^n$  ( $n \geq 2$ ) [7, стр. 520]:

а) в случае нечетных  $n > 1$  и четных  $n$ , для которых  $n > 2m$ ,

$$E_{m,n} = \kappa_{m,n} |x|^{2m-n}, \kappa_{m,n} = \frac{(-1)^m \Gamma(\frac{n}{2} - m)}{\Gamma(m) 2^{2m} \pi^{n/2}}$$

б) в случае четных  $n$ ,  $n \leq 2m$ ,

$$E_{m,n}(x) = \kappa_{m,n} |x|^{2m-n} \ln|x|, \kappa_{m,n} = \frac{(-1)^{\frac{n}{2}-1}}{\Gamma(m) \Gamma(m - \frac{n}{2} + 1) 2^{2m-1} \pi^{n/2}}.$$

Для гармонических систем сдвигов  $m=1$  справедлива теорема В.Г. Лежнева [4, стр. 24]: система функций  $\gamma_{1,k}(x)$ ,  $k=1, 2, \dots$ , линейно независима и замкнута в подпространстве  $G_1(Q)$ , если последовательность  $x^{(k)}$  является 1-базисной в  $Q^+$ .

Одним из основных результатов является обобщение теоремы В.Г. Лежнева на случай полигармонических систем сдвигов. Бигармонический случай ( $m=2$ ) подробно рассмотрен в [8].

**Теорема 3.** Для любого  $m \geq 2$  система функций  $\gamma_{m,k}(x)$ ,  $k=1, 2, \dots$ , замкнута и линейно независима в подпространстве  $G_m(Q)$ , если  $x^{(k)}$  -  $m$ -базисная последовательность в  $Q^+$ .

#### 4 Задача выделения полигармонической составляющей функции

Пусть задана функция  $f(x)$  из  $L_2(Q)$ . Зафиксируем  $m$  - порядок полигармонической составляющей извлекаемой из заданной функции  $f$ .

Согласно теореме 2, функция  $f(x)$  раскладывается в сумму ортогональных слагаемых, так что

$$f(x) = a_1(x) + a_2(x) + \dots + a_m(x) + h_m(x),$$

где  $a_k \in A_k(Q)$ ,  $k=1,2,\dots,m$ , а  $h_m \in N_m(Q)$ .

Рассмотрим следующую задачу: для заданной функции  $f(x) \in L_2(Q)$  определить ее полигармоническую составляющую  $a_m(x)$ .

Случай  $m=1$  рассмотрен в [6]. Будем считать  $m \geq 2$ . Нетрудно видеть, что

$$a_m(x) = g_m(x) - g_{m-1}(x),$$

где  $g_m$  и  $g_{m-1}$  - проекции функции  $f$  на подпространства  $G_m(Q)$  и  $G_{m-1}(Q)$ .

Рассмотрим  $m$ -базисную в  $Q^+$  последовательность  $x^{(k)}$  и соответствующую ей систему функций  $\gamma_{m,k}(x)$ ,  $k=1,2,\dots$ . Обозначим  $g_m^N(x)$  и  $g_{m-1}^N(x)$  - проекции на конечномерные подпространства, натянутые на полные системы:  $\gamma_{m,k}(x)$  и  $\gamma_{m-1,k}(x)$ ,  $k=1,2,\dots,N$ :

$$g_m^N(x) = \sum_{k=1}^N c_{m,k} \gamma_{m,k}(x), \quad g_{m-1}^N(x) = \sum_{k=1}^N c_{m-1,k} \gamma_{m-1,k}(x),$$

тогда

$$a_m^N(x) = g_m^N(x) - g_{m-1}^N(x).$$

По теореме 3 последовательность  $a_m^N$  сходится в норме  $L_2(Q)$  к  $a_m$ , при  $N \rightarrow \infty$ . Определение коэффициентов проекции  $g_m^N$  стандартным образом сводится к решению системы линейных уравнений с матрицей Грама для системы функций  $\gamma_{m,k}(x)$ ,  $k=1,2,\dots,N$ :

$$\sum_{k=1}^N c_{m,k} (\gamma_{m,k}, \gamma_{m,j}) = (f, \gamma_{m,j}), \quad j = 1, \dots, N,$$

где  $(\cdot, \cdot)$  - скалярное произведение в  $L_2(Q)$ . Решая систему линейных уравнений при заданном  $m$  находим матрицу коэффициентов  $c_{m,k}$  ( $k=1,2,\dots,N$ ) приближенной проекции  $g_m^N$  функции  $f$  на полигармоническое подпространство  $G_m(Q)$ . Аналогично, определяется проекция  $g_{m-1}^N$ . Количество функций используемых для аппроксимации теоретически характеризует точность приближенного решения, но при больших  $N$  ( $N > 100$ ) матрица Грама становится плохо обусловленной и для обеспечения высокой точности требуются специальные методы решения линейной системы.

#### 5 Полигармоническое разложение изображения

Под изображением, заданным на поле зрения  $Q=(0,1) \times (0,\beta)$  будем понимать кусочно-постоянную функцию  $f(x)$ , равную  $a_{ij} \in [0,1]$ , при непрерывном

$x=(x_1, x_2) \in \Omega_{ij}$ ,  $i=1, 2, \dots, p$ , и  $j=1, 2, \dots, q$ , при равномерном разбиении (дискретизации)  $Q$  на растры  $\Omega_{ij}$  - квадраты, где  $(a_{ij})$  - заданная матрица яркостной интенсивности пикселей;  $p, q$  - количество пикселей по горизонтали и вертикали соответственно, и  $\beta=p/q$ . Таким образом, изображение  $f(x)$  можно считать элементом из  $L_2(Q)$  - пространства функций, суммируемых с квадратом по области  $Q$ .

Согласно теореме 2, всякое изображение  $f(x) \in L_2(Q)$  может быть разложено единственным образом в сумму изображений (8).

$$f = g_m(f) + h_m(f) = a_1(f) + a_2(f) + \dots + a_m(f) + h_m(f), \quad (10)$$

где:

- $g_m(f)$  - проекция изображения  $f$  на полигармоническое подпространство  $G_m(Q)$ ; полигармоническая составляющая, характеризует «гладкую» часть изображения;
- $a_k(f)$  - ортогональные  $k$ -гармонические компоненты изображения; чистые полигармоники, характеризует гладкую часть по полигармоническим «слоям»;
- $h_m(f)$  - проекция исходного изображения на подпространство  $N_m(Q)$ ; сингулярная компонента, характеризует «точечные» особенности изображения.

В общем случае  $h_m(f) \neq 0$ , но можно показать, что  $\|h_m(f)\| \rightarrow 0$ , при  $m \rightarrow \infty$ , так что для любого  $\varepsilon > 0$  найдется  $m > 0$ , такое что  $\|f - g_m(f)\| < \varepsilon$ , и тогда всякое изображение можно аппроксимировать полигармонической функцией  $g_m(f)$  достаточно большого порядка  $m$ , и, соответственно, приближенное изображение может быть разложено на ортогональные чистые полигармоники:

$$g_m(f) = a_1(f) + a_2(f) + \dots + a_m(f).$$

По теореме 1,  $a_k(f) = \Delta^{1-k} p_k(f)$ , где  $p_k \in G_1(Q)$ ,  $k=1, 2, \dots, m$ , и тогда всякому изображению можно сопоставить единственный набор гармонических функций  $p_k(f)$ ,  $k=1, 2, \dots, m$ , который вместе с компонентой  $h_m(f)$  идентифицирует изображение  $f$ :

$$f = p_1(f) + \Delta^{-1} p_2(f) + \dots + \Delta^{1-m} p_m(f) + h_m(f).$$

Если для изображения при некотором  $m$  сингулярная компонента  $h_m(f)$  равна нулю, то такое изображение будем называть гладким. Для класса гладких изображений справедливо следующее утверждение.

**Теорема 4.** Пространство гладких изображений погружается в степень декартова произведения пространства гармонических функций.

Пусть гармонические функции  $\tilde{p}_k$ ,  $k=1, 2, \dots, m$ , близки функциям  $p_k(f)$ ,  $k=1, 2, \dots, m$ , в норме пространства  $L_2(Q)$ , тогда изображение

$$\tilde{f} = \tilde{p}_1 + \Delta^{-1} \tilde{p}_2 + \dots + \Delta^{1-m} \tilde{p}_m,$$

близко исходному изображению  $f$ , и выполняется неравенство (9).

$$\|f - \tilde{f}\| \leq C \sum_{k=1}^m \|p_k - \tilde{p}_k\|. \quad (11)$$

Можно говорить, что два изображения близки, если близки их гармонические компоненты. Таким образом на пространстве гладких изображений возникает естественная метрика  $\rho$  для которой если  $\|p_k - \tilde{p}_k\| < \varepsilon$ , для любого  $k=1,2,\dots,m$ , то  $\rho(f, \tilde{f}) < \varepsilon$ .

В случае гладкого изображения сингулярная компонента равна нулю, и такое разложение может быть использовано, например, в задачах обработки размытых изображений [9, 10], в вычислительной газодинамике при анализе локализации структур [11, 12] и в анализе рентгенограмм [13].

Ниже на рис. 1 представлены результаты численного эксперимента: количество базисных точек  $N$  равно 60, точки  $x^{(k)}$  располагаются равномерно во внешности  $Q=[0,1]^2$  на прямых, параллельных границе  $Q$ , отстоящих от нее на расстоянии  $\delta=0.2$ . Для заданного изображения  $f$  приводятся вычисленные проекции  $g_m$  на полигармонические подпространства  $G_m(Q)$ ,  $m=1,2,\dots,8$ , и проекция  $h_8$  на подпространство  $N_8(Q)$ .

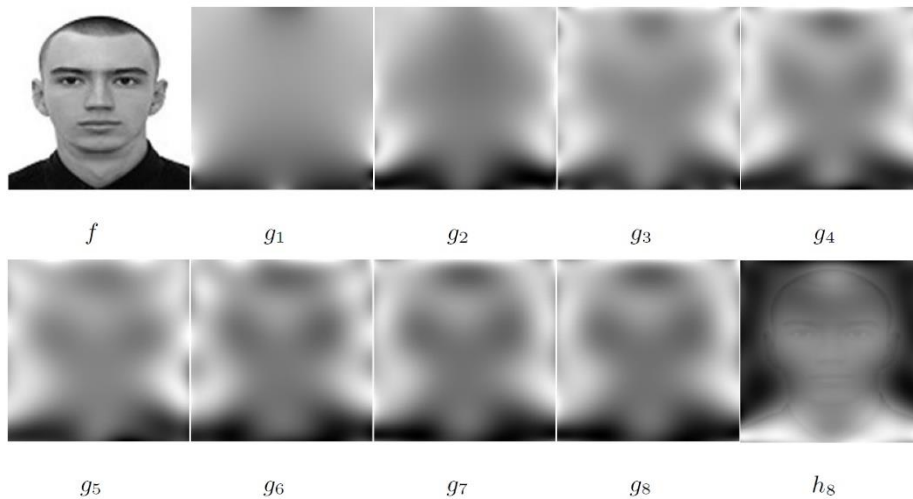


Рис. 9. Результаты численного эксперимента

## 6 Заключение

В работе приводится разложение пространства функций, суммируемых с квадратом, в ортогональную сумму полигармонических подпространств. Рассматриваются системы сдвигов фундаментальных решений полигармонических уравнений, и приводится достаточное условие полноты таких систем. Разрабатывается сходящийся алгоритм определения проекций заданной функции на полигармонические подпространства. Определяется класс гладких изображений и на

этом классе вводится естественная метрика. Показано, как свойства и методы полигармонических функций применяются к анализу цифровых изображений.

Разрабатываемый подход, а также прямые и обратные полигармонические операторы могут быть эффективно применены к некоторым задачам обработки цифровых изображений, таких как: сглаживание и концентрирование, мера гладкости и структурированности, выделение контуров, восстановление изображений, алгоритмы сжатия, устранение импульсных шумов, стирание объектов и обработка края изображения. Некоторые вопросы в гармоническом случае  $m=1$  были рассмотрены в [14].

## Литература

1. Соيفер В.А. и др. Методы компьютерной обработки изображений/ Под ред В.А. Соифера. - 2-е изд., испр. М.:Физматлит, 2003
2. Купрадзе В.Д. О приближенном решении задач математической физики // Успехи математических наук, 22:2 (1967), 59--107.
3. Алексидзе М.А. Решение граничных задач методом разложения по неортогональным функциям. М.: Наука, 1978.
4. Лежнев А.В., Лежнев В.Г. Метод базисных потенциалов в задачах математической физики и гидродинамики. Краснодар: КубГУ, 2009.
5. Vgomonlly A. Fundamental solutions method for elliptic boundary value problems // J. Num. Anal. 22:4 (1985), 644-669.
6. Лежнев В.,Г. Выделение гармонической составляющей // Численный анализ: теория, приложения, программы: Сборник научных трудов, МГУ, 1999, 90-95.
7. Соболев С.Л. Введение в теорию кубатурных формул. М.:Наука, 1974.
8. Марковский А.Н. Замкнутость бигармонической системы базисных потенциалов // Экологич. вестн. науч. центр. Черноморского экономического сотрудничества. Т.~17. N.~1. Ч.~2. (2020), 20--26.
9. Кольцов П.,П. Оценка размытия изображения // Компьютерная оптика. 31:1 (2011), 95-102.
10. Асатрян Д.Г. Оценивание степени размытости изображения путём анализа градиентного поля // Компьютерная оптика. 41:6 (2017), 957-962.
11. Базаров С.Б. Применение методов обработки изображений в вычислительной газодинамике // Труды GraphiCon 98, Москва, 1998, 258-264.
12. Афендииков А.Л., Луцкий А.Е., Пленкин А.В. Применение вейвлет анализа для локализации структур в расчетах по идеальной и вязкой моделям и для адаптации сеток // Препринты ИПМ им. М. В. Келдыша, 2012, 045, 24 стр.
13. Гайдель А.В., Крашенинников В.Р. Отбор признаков для задачи диагностики остеопороза по рентгеновским изображениям шейки бедра // Компьютерная оптика. 40:6 (2016), 939-946.
14. Василенко В.В., Сафронов А.П., Смыслов А.А., Цепляев Д.П., Марковский А.Н. Гармоническое сглаживание цифровых изображений // Экологич. вестн. науч. центр. Черноморского экономического сотрудничества. Т.~17. N.~1. Ч.~2. (2020), 8--15.

*Научное издание*

# **GraphiCon 2020**

30-я Международная конференция  
по компьютерной графике  
и машинному зрению

## **Труды конференции. Том 1**

Издается в авторской редакции  
Компьютерная верстка: *Болдырева Е.А.*

Институт прикладной математики имени М.В. Келдыша РАН  
125047, Москва, Миусская пл., д.4