

Министерство образования и науки Российской Федерации
федеральное государственное автономное образовательное учреждение
высшего образования
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Институт кибернетики
Направление подготовки 09.04.02 «Информационные системы и технологии»
Кафедра информационных систем и технологии

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

Тема работы
Алгоритм и программные средства для создания панорамных изображений

УДК 004.42:778.534.164:004.92

Студент

Группа	ФИО	Подпись	Дата
8ИМ5А	Толеген Молдир Орынбасаркызы		

Руководитель

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент каф. ИСТ	Болотова Юлия Александровна	к.т.н		

КОНСУЛЬТАНТЫ:

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Старший преподаватель	Попова Светлана Николаевна	к.э.н		

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент каф. ЭБЖ	Акулов Петр Анатольевич			

ДОПУСТИТЬ К ЗАЩИТЕ:

Зав. кафедрой	ФИО	Ученая степень, звание	Подпись	Дата
ИСТ	Мальчуков Андрей Николаевич	к.т.н		

Томск – 2017 г.

Планируемые результаты обучения

Код результатов	Результат обучения (выпускник должен быть готов)
Общепрофессиональные компетенции	
Р1	Воспринимать и самостоятельно приобретать, развивать и применять математические, естественнонаучные, социально-экономические и профессиональные знания для решения нестандартных задач, в том числе в новой или незнакомой среде и в междисциплинарном контексте.
Р2	Владеть и применять методы и средства получения, хранения, переработки и трансляции информации посредством современных компьютерных технологий, в том числе в глобальных компьютерных сетях.
Р3	Демонстрировать культуру мышления, способность выстраивать логику рассуждений и высказываний, основанных на интерпретации данных, интегрированных из разных областей науки и техники, выносить суждения на основании неполных данных, анализировать профессиональную информацию, выделять в ней главное, структурировать, оформлять и представлять в виде аналитических обзоров с обоснованными выводами и рекомендациями.
Р4	Анализировать и оценивать уровни своих компетенций в сочетании со способностью и готовностью к саморегулированию дальнейшего образования и профессиональной мобильности. Владеть, по крайней мере, одним из иностранных языков на уровне социального и профессионального общения, применять специальную лексику и профессиональную терминологию языка.
Профессиональные компетенции	
Р5	Разрабатывать стратегии и цели проектирования, критерии эффективности и ограничения применимости, новые методы, средства и технологии проектирования геоинформационных систем (ГИС) или промышленного программного обеспечения.
Р6	Планировать и проводить теоретические и экспериментальные исследования в области создания интеллектуальных ГИС и ГИС технологии или промышленного программного обеспечения с использованием методов системной инженерии.
Р7	Осуществлять авторское сопровождение процессов проектирования, внедрения и сопровождения ГИС и ГИС технологий или промышленного программного обеспечения с использованием методов

	и средств системной инженерии, осуществлять подготовку и обучение персонала.
P8	Формировать новые конкурентоспособные идеи в области теории и практики ГИС и ГИС технологий или системной инженерии программного обеспечения. Разрабатывать методы решения нестандартных задач и новые методы решения традиционных задач. Организовывать взаимодействие коллективов, принимать управленческие решения, находить компромисс между различными требованиями как при долгосрочном, так и при краткосрочном планировании.
Общекультурные компетенции	
P9	Использовать на практике умения и навыки в организации исследовательских, проектных работ и профессиональной эксплуатации современного оборудования и приборов, в управлении коллективом.
P10	Свободно пользоваться русским и иностранным языками как средством делового общения.
P11	Совершенствовать и развивать свой интеллектуальный и общекультурный уровень. Проявлять инициативу, в том числе в ситуациях риска, брать на себя всю полноту ответственности.
P12	Демонстрировать способность к самостоятельному обучению новым методам исследования, к изменению научного и научно-производственного профиля своей профессиональной деятельности, способность самостоятельно приобретать с помощью информационных технологий и использовать в практической деятельности новые знания и умения, в том числе в новых областях знаний, непосредственно не связанных со сферой деятельности, способность к педагогической деятельности.

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Институт кибернетики
Направление подготовки 09.04.02 «Информационные системы и технологии»
Кафедра информационных систем и технологий

УТВЕРЖДАЮ:

Зав. кафедрой

_____ Мальчуков А.Н.
(Подпись) (Дата) (Ф.И.О.)

ЗАДАНИЕ

на выполнение выпускной квалификационной работы

В форме:

Магистерской диссертации

Студенту:

Группа	ФИО
8ИМ5А	Толеген Молдир Орынбасаркызы

Тема работы:

Алгоритм и программные средства для создания панорамных изображений

Утверждена приказом директора (дата, номер)

897/С от 20.02.2017 г.

Срок сдачи студентом выполненной работы:

ТЕХНИЧЕСКОЕ ЗАДАНИЕ:

Исходные данные к работе	Объектом исследования является задача создания панорамных изображений. Алгоритм создание панорамных изображений позволяет создавать трехмерные модели окружающего пространства, что может быть применено в сферах туризма, картографии, космосе и т.д..
Перечень подлежащих исследованию, проектированию и разработке вопросов	Обзор алгоритмов обработки и анализа изображений, проектирование структуры и содержания основных классов инфраструктуры разрабатываемого программного обеспечения, разработка основных компонентов программного обеспечения, расчет ресурсоэффективности и ресурсосбережения, анализ вредных производственных факторов.
Перечень графического материала	Блок-схема алгоритма, структура основных классов библиотеки, результатов панорамных изображений, изображений работы алгоритмов

	нахождения особых точек, сопоставление, сшивание и трехмерных модели.
--	---

Консультанты по разделам выпускной квалификационной работы

Раздел	Консультант
Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	Доцент каф. менеджмента, к.э.н., Попова Светлана Николаевна
Социальная ответственность	Ассистент каф. экологии и безопасности жизнедеятельности, Акулов Петр Анатольевич
Раздел на иностранном языке	Старший преподаватель иностранных языков, Горбатова Татьяна Николаевна

Названия разделов, которые должны быть написаны на русском и иностранном языках:

Разделы введение, аналитический обзор, алгоритм и программные средства для создания панорамных изображений, результаты тестирования, финансовый менеджмент, ресурсоэффективность и ресурсосбережение, социальная ответственность и заключение должны быть написаны на русском языке.

Раздел аналитический обзор должен быть написан на английском языке.

Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику	
---	--

Задание выдал руководитель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Болотова Юлия Александровна	к.т.н		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8ИМ5А	Толеген Молдир Орынбасаркызы		

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Институт кибернетики
Направление подготовки 09.04.02 «Информационные системы и технологии»
Уровень образования магистр
Кафедра информационных систем и технологий
Период выполнения осенний семестр 2016 г. – весенний семестр 2017 учебного года

Форма представления работы:

Магистерская диссертация

КАЛЕНДАРНЫЙ РЕЙТИНГ-ПЛАН
выполнения выпускной квалификационной работы

Срок сдачи студентом выполненной работы:

Дата контроля	Название раздела (модуля) / вид работы (исследования)	Максимальный балл раздела (модуля)
10.11.2016	Основная часть	62
02.02.2017	Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	17
17.03.2017	Социальная ответственность	17
06.04.2017	Обязательное приложение на иностранном языке	17

Составил преподаватель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Болотова Юлия Александровна	К.Т.Н.		

СОГЛАСОВАНО:

Зав. кафедрой	ФИО	Ученая степень, звание	Подпись	Дата
ИСТ	Мальчуков Андрей Николаевич	К.Т.Н.		

РЕФЕРАТ

Итоговая квалификационная работа содержит 116 с., 26 рис., 5 табл., 35 источников, 2 прил.

Ключевые слова: создание панорамных изображений, особые точки, дескрипторы, геометрические преобразования, гомография.

Объектом исследования является задача создания панорамных изображений.

Цель работы – разработка алгоритма создания панорамных изображений.

В ходе работы были исследованы и определены существующие алгоритмы вычисления особых точек, проанализированы метрики их сопоставления, поиска перспективного преобразования между двумя изображениями и синтеза панорамных изображений.

В результате исследования был предложен и реализован алгоритм создания панорамных изображений.

Область применения: создание панорамных изображений позволяет создавать трехмерные модели окружающего пространства, что может быть применено в сферах туризма, картографии, космосе и т.д.

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

IT – информационные технологии.

SIFT – (Scale Invariant Feature Transform) – масштабное инвариантная функция преобразования.

PCA – (Principal Component Analysis) – метод главных компонент.

SURF – (Speeded up Robust Features) – ускоренные устойчивые признаки.

BRIEF – (Binary Robust Independent Elementary Features) – бинарные устойчивые независимые элементарные признаки.

FAST – (features from accelerated segment test) признаки, полученные из ускоренной проверки сегментов.

ORB – (Oriented FAST and Rotated BRIEF) – Ориентированный FAST, полученный из ускоренной проверки сегментов и BRIEF.

RANSAC – (random sample consensus) метод поиска матрицы перспективного преобразования по особым точкам.

Патч – отдельный участок изображения.

FLANN – (Fast Library for Approximate Nearest Neighbors) – метод сопоставления особых точек.

СОДЕРЖАНИЕ

ВЕДЕНИЕ	12
1.1. ОБЗОР АЛГОРИТМОВ ПОИСКА КЛЮЧЕВЫХ ТОЧЕК	15
1.2. МАТЕМАТИЧЕСКОЕ ОПИСАНИЕ ДЕСКРИПТОРОВ ОСОБЫХ ТОЧЕК	19
1.2.1 Алгоритм SIFT	20
1.2.2 Алгоритм SURF	22
1.2.3 Алгоритм ORB	26
1.3. СОПОСТАВЛЕНИЕ ОСОБЫХ ТОЧЕК	28
1.3.1 Метод k-средних	28
1.3.2 Евклидово расстояние	29
1.3.3 Расстояние Хэмминга	30
1.3.4 Расстояния Манхэттена	30
1.3.5 Расстояние Чебышева	31
1.4. СПИВАНИЕ ИЗОБРАЖЕНИЙ	31
1.4.1 Геометрические преобразования	31
1.4.2 Алгоритм RANSAC	33
1.4.3 Генерация трёхмерных объектов	34
2 РАЗРАБОТКА И РЕАЛИЗАЦИЯ АЛГОРИТМА СОЗДАНИЯ ПАНОРАМНЫХ ИЗОБРАЖЕНИЙ	38
2.1. ОПИСАНИЕ БИБЛИОТЕКИ OPENCV	38
2.1.1 Вычисление детекторов ключевых точек на изображении	39
2.1.2 Структура работы компьютерной модели	40
2.2. АЛГОРИТМ СОЗДАНИЯ ПАНОРАМНЫХ ИЗОБРАЖЕНИЙ	41
2.3. ВЫБОР СРЕДЫ И ЯЗЫКА	44
3 РЕЗУЛЬТАТЫ ЭКСПЕРИМЕНТОВ	48
3.1. ОПИСАНИЕ ТЕСТОВОЙ ВЫБОРКИ СОПОСТАВЛЕНИЕ ИЗОБРАЖЕНИЙ	48
3.2. СИНТЕЗ ИЗОБРАЖЕНИЙ В ПАНОРАМУ	50
3.3. АНАЛИЗ РЕЗУЛЬТАТОВ	51

4	ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И РЕСУРСОСБЕРЕЖЕНИЕ.....	54
4.1.	ОРГАНИЗАЦИЯ И ПЛАНИРОВАНИЕ РАБОТ.....	54
4.1.1	Продолжительность этапов работ	55
4.1.2	Расчет нарастания технической готовности работ	58
4.2.	РАСЧЕТ СМЕТЫ ЗАТРАТ НА РАЗРАБОТКУ ПО И ПРОВЕДЕНИЕ ЭКСПЕРИМЕНТА	59
4.2.1.	Расчет затрат на материалы.....	59
4.2.2.	Расчет основной заработной платы.....	59
4.2.3.	Расчет отчислений от заработной платы	60
4.2.4.	Расчет затрат на электроэнергию	61
4.2.5.	Расчет амортизационных расходов	62
4.2.6.	Расчет расходов на услуги связи	62
4.2.7.	Расчет прочих расходов.....	63
4.2.8.	Расчет общей себестоимости разработки	63
4.3.	ОЦЕНКА ЭКОНОМИЧЕСКОЙ ЭФФЕКТИВНОСТИ ПРОЕКТА	64
4.3.1	Оценка научно-технического уровня НИР	64
5	СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ.....	70
	ВВЕДЕНИЕ.....	70
5.1.	ПРОФЕССИОНАЛЬНАЯ СОЦИАЛЬНАЯ БЕЗОПАСНОСТЬ	70
5.1.1	Нарушение параметров микроклимата	71
5.1.2	Плохой освещенность рабочей зоны.....	72
5.1.3	Монотонный режим работы.....	73
5.1.4	Опасность поражения электрическим током	74
5.1.5	Опасность возникновения пожара.....	75
5.2	ЭКОЛОГИЧЕСКАЯ БЕЗОПАСНОСТЬ.....	76
5.3	БЕЗОПАСНОСТЬ В ЧРЕЗВЫЧАЙНЫХ СИТУАЦИЯХ.....	77
5.4	ПРАВОВЫЕ И ОРГАНИЗАЦИОННЫЕ ВОПРОСЫ ОБЕСПЕЧЕНИЯ БЕЗОПАСНОСТИ	80
	ЗАКЛЮЧЕНИЕ	82
	ОПУБЛИКОВАННЫЕ РАБОТЫ	83

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ	84
ПРИЛОЖЕНИЕ А	88
ПРИЛОЖЕНИЕ Б.....	99

ВЕДЕНИЕ

Десятки лет исследователи искусственного интеллекта из разных стран занимаются разработкой алгоритмов, позволяющих научить компьютер видеть так же, как видит человек. Существуют частные методы, которые позволяют получить желаемый результат в узконаправленных задачах при меньших затратах. Они, как правило, основаны не на структуре человеческого аппарата анализа и интерпретации изображений, а непосредственно на искусственно выделяемых особенностях самого изображения.

Данная работа посвящена решению задачи создания панорамных изображений. Решение задачи создания панорамных изображений требуется при разработке виртуальных туров и экскурсий, моделировании, в системах виртуальной реальности, кинематографии. Вопрос эффективности таких алгоритмов ставится наиболее остро, особенно при работе с мобильными устройствами – смартфонами, без которых сложно представить жизнь современного человека.

Актуальность темы исследования.

Панорамное изображение – фотография, полученная путем технологии сборки панорам из отдельных кадров, с большим углом обзора, превышающим возможности обычных объективов, а зачастую и больше 180 градусов, или обычная фотография «длинного» формата, с соотношением сторон 1×2, 1×3 и больше. Стремительное развитие цифровых фотографий привело к появлению панорамных изображений. Мир панорамных фотографий позволяет нам увидеть более реалистичную картину отснятых изображений. Панорамы городов, природы, космоса, панорамные карты окунают нас в новый мир цифровых фотографий. Часто в одной панораме виден больший угол обзора, чем позволяет увидеть человеческий глаз.

Цель исследования. Создание и разработка алгоритма синтеза панорамных изображений из серии фотографий.

Объект исследования. Объектом исследования является создание панорамных изображений, получаемые из последовательности картинок.

Предмет исследования. Предметом исследования являются методы, существующие алгоритмы вычисления особых точек и синтеза панорамных изображений.

Данная работа предполагает решение следующих задач: выделение особых точек на двух изображениях, вычисление признаков описаний особых точек (получение дескрипторов), сопоставление особых точек двух изображений, сшивание изображений.

В результате работы над проектом выходит нижеперечисленные решения задачи сопоставления изображений:

1. На изображениях выделяются особые точки, вычисляются их дескрипторы.
2. Происходит поиск на разных изображениях соответствующих особых точек.
3. Исходя из набора соответствующих особых точек строится модель преобразования изображений, на основе которой два исходных изображения совмещаются в панораму.

Данная работа состоит из четырех разделов. В первом разделе описана основная теоретическая часть: исследование алгоритмов поиска особых точек, осуществление сопоставления особых точек, оценка перспективного преобразования между двумя изображениями, сшивание изображений и моделирование трехмерной сцены.

Во втором разделе описан предлагаемый алгоритм и реализованное в ходе работы программное обеспечение для поиска особых точек и осуществление сопоставления особых точек. В результате сопоставления различных вариантов предложенного алгоритма, был выбран вариант, показывающий наилучшую точность сопоставления изображений. Была описана программа, которая позволяет осуществить перспективное преобразование и сшивание между двумя изображениями.

В третьем разделе приведены результаты тестирования, разработанного ПО для сопоставления особых точек и соединения изображений.

В четвертом и пятом разделах приведены материалы, связанные с ресурсоэффективностью и социальной ответственностью.

Методы исследования. Для решения поставленных в данной работе задач были использованы существующие методы поиска особых точек, сопоставления особых точек, методы оценки перспективных преобразований, метрики сопоставления векторов.

По теме диссертации были сделаны две публикации в международных научно-практических конференциях студентов, аспирантов и молодых ученых:

1. XIV Международная научно-практическая конференция студентов, аспирантов и молодых ученых «Молодежь и современные информационные технологии» 7 – 11 ноября, 2016 г., г. Томск.

2. Международная научно-техническая конференция студентов, аспирантов и молодых учёных «Научная сессия ТУСУР – 2017», посвящённая 55-летию ТУСУРа, 10 по 12 мая, 2017. За доклад «Исследование методов поиска и сопоставления особых точек для решения задачи построения панорамных изображений» был получен диплом II степени.

Научная новизна работы заключается в разработке алгоритма создания панорамных изображений.

1 ОБЗОР АЛГОРИТМОВ ОБРАБОТКИ И АНАЛИЗА ИЗОБРАЖЕНИЙ

1.1. Обзор алгоритмов поиска ключевых точек

Ключевое значение при построении признакового описания изображения имеет выделение характерных для изображения частей, в качестве которых можно рассмотреть, например, углы, ребра, регионы, соответствующие экстремумам интенсивности, и т.п. На алгоритмы, выделяющие такие особенности (ключевые точки), накладываются требования инвариантности относительно преобразований смещения и поворота, изменения масштаба и освещенности исходного изображения, а также точки положения камеры, с которой снят объект. Для выделения из изображения некоторой интерпретируемой информации необходимо привязаться к локальным особенностям изображения. На изображении можно выделить особые точки. *Особая точка m* – является точкой образа, окрестность $o(m)$ которой можно выделить из окрестности любой другой точки образа $o(n)$ в некоторой другой окрестности особой точки $o_2(m)$ [20]. В работе [21] были предложены следующие свойства ключевых точек:

Повторяемость (repeatability) – особая точка находится в том же месте сцены или объекта изображения, несмотря на изменения освещении и в точке обзора.

Отличительность / информативность (distinctiveness/informativeness) – окрестности особых точек должны иметь большие отличия друг от друга, так, чтобы возможно было выделить и сопоставить особые точки.

Локальность (locality) – для снижения чувствительности фотометрическим и геометрическим искажениям между двумя изображениями, особая точка должна занимать небольшую область изображения.

Количество (quantity) – для обнаружения небольших объектов число обнаруженных особых точек должно быть достаточно большим. Количество обнаруженных особых точек должно определяться адаптивно. Чтобы

обеспечить компактное представление особых точек должна показывать информационное содержимое изображения.

Точность (accuracy) – определенные особые точки должны точно ограничиваться, в другом масштабе, так и в исходном изображении.

Эффективность (efficiency) – время определенных особых точек на изображении должно быть удовлетворительным в критичных по времени приложениях.

Подходы к определению особых точек можно разделить на 3 категории [19]:

1. Исходя из интенсивности изображения: особые точки вычисляются непосредственно из значений интенсивности пикселей изображения.

2. Использование контуров изображений: методы извлечения контуров и поиска мест с максимальным значением кривизны или полигональной аппроксимации контуров и определения пересечений. Эти методы чувствительны к пересечениям пересечений, так как извлечение часто может быть неправильным в тех местах, где пересекаются 3 или более краев.

3. Основываясь на использовании модели: модели с интенсивностью используются в качестве параметров, которые адаптируются к изображениям до субпиксельной точности. Они имеют ограниченное применение с точками определенных типов (например, L-связанных углов), зависят от используемых шаблонов.

Нужно выбирать такие точки, которые вносят некий вклад в описание изображения, также необходимо считать особыми такие точки, которые с большой вероятностью будут найдены на другом (подобном) изображении. Каждый метод обнаружения особых точек должен гарантировать инвариантность относительно аффинных и перспективных преобразований.

Процесс нахождения особых точек осуществляется с помощью детектора, а их описание производится и дескриптором.

Детектор – это метод извлечения особых точек из изображения. Детектор обеспечивает инвариантность нахождения одних и тех же особых точек относительно различных преобразований.

Дескриптор – идентификатор особой точки, выделяющий её из остального множества особых точек. В свою очередь, дескрипторы должны обеспечивать инвариантность нахождения соответствия между особыми точками относительно преобразований изображений [20]. В результате построения дескрипторов для исходного набора особых точек вырабатывается множество векторов признаков.

Сейчас наиболее известным дескриптором ключевых точек является *SIFT* [6], в основе которого лежит идея вычисления в окрестности особой точки гистограммы ориентированных градиентов. Модификацией данного дескриптора является *PCA-SIFT* [3], который строится по той же схеме, используя окружения большего размера. В отличие от базового дескриптора *SIFT* в нем осуществляется снижение размерности векторов для результирующего набора дескрипторов посредством анализа главных компонент (PCA) [2].

Дескриптор SURF [1] также относится к числу тех дескрипторов, которые одновременно выполняют поиск особых точек и строят их описание, инвариантное к изменению масштаба и повороту изображения. Фактически, данный дескриптор представляет собой взвешенные значения градиентов и их абсолютных значений в 16 квадрантах, на которые разбита окрестность особой точки.

Цель создания дескриптора *BRIEF* [4] заключалась в обеспечении распознавания одинаковых частей изображения, полученных с разных углов обзора. Задача состояла в том, чтобы сократить количество выполненных вычислений. В окрестности точки выбирается некоторым образом множество пар пикселей, и на полученном множестве строится набор бинарных тестов, представляющих собой сравнение интенсивности в соответствующих пикселях. Результаты тестов объединяются в битовую строку, которая и является итоговым

дескриптором ключевой точки. Более эффективной альтернативой дескриптору *BRIEF* является бинарный дескриптор *ORB* [5].

Алгоритм *FAST* [13] не требует вычисления производных яркости. На рис.1 показана анализируемая точка и окружающие ее пиксели. Яркость пикселей, лежащих на окружности, сравнивается с яркостью центральной точки, и на основании ряда проверок принимается решение, является ли центральная точка особой. Последовательность проверок и их общее число подбираются и оптимизируются заранее на основе обширной обучающей выборки изображений. В результате проверки выполняются очень быстро. Для принятия решения, является точка углом или нет, требуется лишь несколько десятков операций сравнения. Алгоритм *FAST* хорошо зарекомендовал себя в приложениях, осуществляющих слежение за объектами в реальном времени [14].

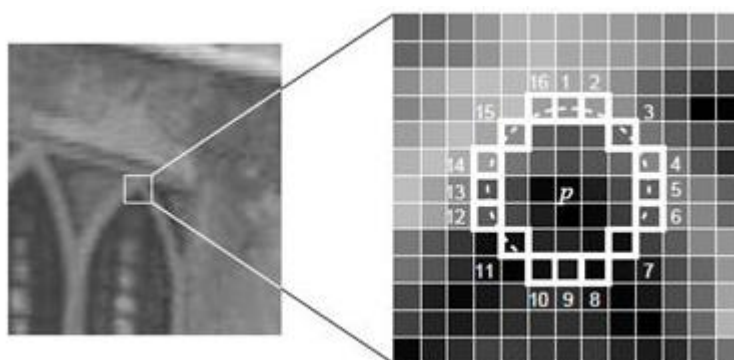


Рисунок 1 – Пиксели, используемые в быстрых проверках алгоритма *FAST*

Алгоритм *ORB* – это ориентированный *FAST* и Rotated *BRIEF* – улучшенная версия и комбинация *FAST*-точечного детектора и двоичных дескрипторов *BRIEF*. *ORB* обладает одним важным нюансом – размер дескрипторов равен 32 байтам на одну точку. Проверка совпадения представляет собой сумму расстояний Хэмминга для каждого байта дескриптора (первый сравнивается с первым, второй со вторым и т. д.).

1.2. Математическое описание дескрипторов особых точек

Результатом работы детекторов является множество особых точек, для которых необходимо построить математическое описание. В данном разделе приведен обзор некоторых существующих дескрипторов.

Входными данными дескриптора является изображение и набор особых точек, выделенных на заданном изображении. Выходом дескриптора является множество векторов признаков для исходного набора особых точек. Необходимо отметить, что какие-то дескрипторы решают одновременно две задачи – поиск особых точек и построение описателей этих точек.

Признаки (описатели) строятся на основании информации об интенсивности, цвете и текстуре особой точки. Но особые точки могут представляться углами, ребрами или даже контуром объекта, поэтому, как правило, вычисления выполняются для некоторой окрестности. В идеале хорошие признаки должны обладать рядом свойств:

Отличимость (*distinctness*) – особая точка должна явно выделяться на фоне и быть отличимой (уникальной) в своей окрестности.

Инвариантность (*invariance*) – определение особой точки должно быть независимо к аффинным преобразованиям.

Стабильность (*stability*) – определение особой точки должно быть устойчиво к шумам и ошибкам.

Уникальность (*uniqueness*) – кроме локальной отличимости, особая точка должна обладать глобальной уникальностью для улучшения различимости повторяющихся паттернов.

Интерпретируемость (*interpretability*) – особые точки должны определяться так, чтобы их можно было использовать для анализа соответствий и выявления интерпретируемой информации из изображения.

1.2.1 Алгоритм SIFT

Алгоритм *SIFT* масштабно-пространственный детектор: на первой стадии вычисляются ключевые точки на изображениях различного масштаба. После чего происходит фильтрация обнаруженных ключевых точек, в результате которой остаются наиболее стабильные из них. Для каждой найденной ключевой точки определяется одна или несколько главных ориентаций, вычисляются дескрипторы [18].

Алгоритм состоит из следующих этапов [1]:

- 1) построение пирамиды гауссианов и их разностей. На этом шаге обеспечивается инвариантность к масштабированию;
- 2) определение экстремумов;
- 3) уточнение особых точек;
- 4) построение дескрипторов (обеспечивается инвариантность к освещению, шуму, изменению положения камеры).

SIFT позволяет определять ключевые точки в виде капель (англ. blobs), так как они инвариантны ко всем видам аффинных преобразований. Капли представляют собой структуры, описываемые координатами центра, масштабом и направлением [7-9]. Этот тип структуры является наиболее сложным среди всех форм особых точек и, следовательно, обеспечивает их устойчивое обнаружение.

На первом этапе алгоритма *SIFT* [10, 11] строится масштабируемое пространство изображений – набор изображений, сглаженных Гауссовым фильтром

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}, \quad (1)$$

где (x, y) – координаты точки; σ – радиус размытия.

На основе полученного масштабируемого пространства изображений осуществляется гауссова разность $D(x, y, \sigma)$ – попиксельное вычитание изображений в одной октаве с различным радиусом размытия. Октаву формируют изображения одного масштаба, размытые фильтром Гаусса с

различным радиусом размытия (4 изображения в одной октаве). На этом этапе обеспечивается инвариантность к масштабированию. Затем определяются экстремумы, которые записываются в список потенциальных особых точек.

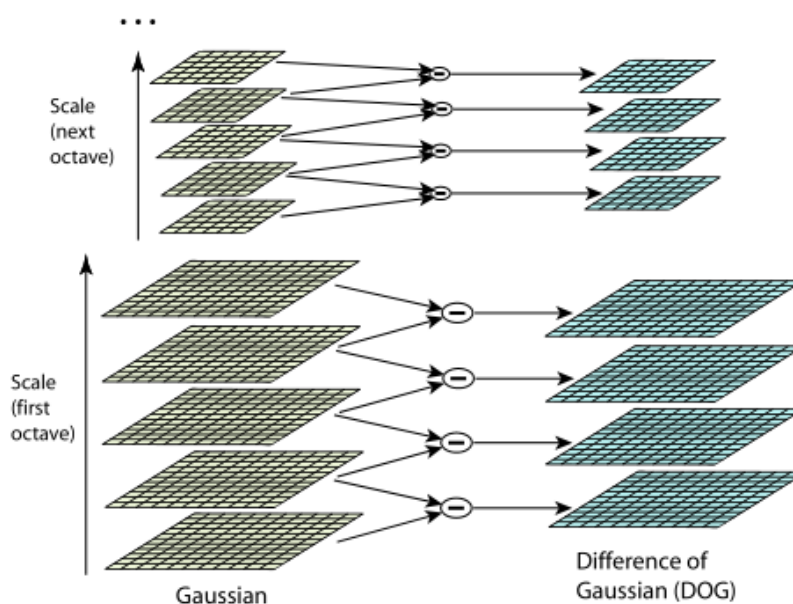


Рисунок 2 – Разность Гауссо-размытых изображений

На рис.2. слева изображена пирамида гауссианов, а справа — их разностей. Схематично показано, что каждая разность получается из двух соседних гауссианов, количество разностей на единицу меньше количества гауссианов. При переходе к следующей октаве размер изображений уменьшается вдвое.

После построения пирамид точка считается особой, если она является локальным экстремумом разности гауссианов. Для поиска экстремумов используется метод, схематично изображенный на рис.3.

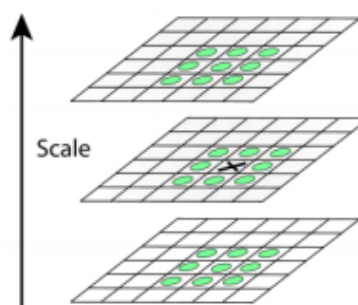


Рисунок 3 – Метод поиска экстремумов разности гауссианов

Если значение разности гауссианов в точке, помеченной крестиком, больше (меньше) всех значений в точках, помеченных окружностями, то эта точка считается точкой экстремума.

Алгоритм *SIFT* представляет наилучшие результаты обнаружения ключевых точек и получения их дескрипторов, и подходит исследуемой в данной работе предметной области [1-3]. В Приложении Б (рис.15) приведен результат применения алгоритма *SIFT* для поиска особых точек.

1.2.2 Алгоритм SURF

Альтернативным подходом является *SURF*, который в несколько раз быстрее *SIFT*. Алгоритм *SURF* осуществляет поиск ключевых точек изображения и вычисление дескрипторов, которые являются инвариантными относительно масштаба и поворота. Он хорошо обнаруживает пятна, углы и края линий. Данный алгоритм реализуется на основе интегрального представления изображения, что ускоряет поиск особых точек.

Значение в каждой точке интегрального изображения вычисляется как сумма значений яркости в данной точке и значений яркости всех точек, которые выше и левее данной точки. С помощью интегральных изображений для постоянного времени вычисляются прямоугольные фильтры, которые состоят из нескольких прямоугольных областей [1].

Определение особых точек на изображении осуществляется на основе матрицы Гессиана (FAST-Hessian detector). Применение Гессиана обеспечивает инвариантность относительно преобразования поворота, но не инвариантность относительно изменения масштаба. Поэтому *SURF* применяет фильтры разного масштаба для вычисления Гессиана. Скажем, что исходное изображение задается матрицей интенсивностей I , текущий рассматриваемый пиксель обозначается как $X = (x, y)$, а σ – масштаб фильтра. Тогда матрица Гессе имеет вид:

$$H(x, y, \sigma) = \begin{pmatrix} L_{xx}(x, y, \sigma) & L_{xy}(x, y, \sigma) \\ L_{yx}(x, y, \sigma) & L_{yy}(x, y, \sigma) \end{pmatrix} \quad (2)$$

где

$$L_{xx}(x, y, \sigma) = I(x, y) * \frac{d^2 g_\sigma}{dx^2},$$

$$L_{yy}(x, y, \sigma) = I(x, y) * \frac{d^2 g_\sigma}{dy^2},$$

$$L_{xy}(x, y, \sigma) = I(x, y) * \frac{d^2 g_\sigma}{dxdy}.$$

Здесь $I(x, y)$ – исходное изображение, $g_\sigma(x, y)$ – функция Гаусса (1), «*» – оператор свертки.

Значение гессиана используется для нахождения локального минимума или максимума яркости изображения. В этих точках значение гессиана достигает экстремума.

На рис. 4 видно, что особые точки (очерченные цветными кругами) представляют собой локальные экстремумы яркости изображения. Мелкие точки не распознаны как особые, из-за порогового отсечения по величине гессиана.

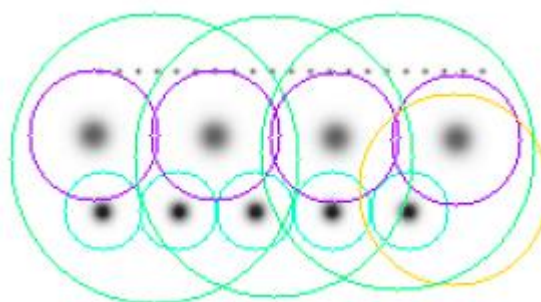


Рисунок 4 – Особые точки детектора SURF

После этого для каждой найденной ключевой точки вычисляется ориентация – преобладающее направление разности яркости. Понятие ориентации близко к понятию направления градиента, но для определения ориентации особой точки применяется фильтр Хаара. Пример фильтров Хаара показан на рис. 5.



Рисунок 5 – Фильтры Хаара (черные области имеют значения «-1», белые «+1»)

Дескриптор представляет собой массив из 64 (в расширенной версии 128) чисел, которые позволяют идентифицировать ключевую точку. Дескрипторы должны примерно совпадать в одной и той же особой точке на первом и втором изображении. Метод вычисления дескриптора не зависит от масштаба и вращения.

Для вычисления дескриптора вокруг особой точки образуется прямоугольная область размером $20s$, где s - масштаб, в котором найдена особая точка. Для первой октавы, область имеет размер 40×40 пикселей. Квадрат ориентирован вдоль приоритетного направления, вычисленного для особой точки. Дескриптор рассматривается как описание градиента для 16 квадрантов вокруг особой точки.

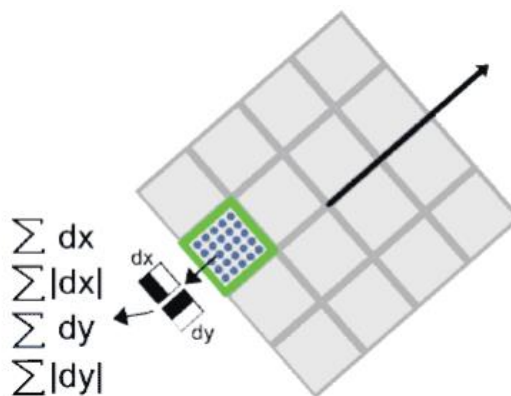


Рисунок 6 – Вычисление дескриптора особой точки

Затем квадрат делится на 16 меньших квадрантов, как показано на рис.б. В каждом квадранте берется регулярная сетка 5×5 и подбирается точки сетки с использованием фильтра Хаара. Размер фильтра Хаара принимается равным $2s$, и для первой октавы – 4×4 .

Следует отметить, что при вычислении фильтра Хаара изображение не вращается, фильтр рассматривается в обычных координатах изображения. Полученные градиентные координаты (dX , dY) поворачиваются на угол, соответствующий ориентации квадрата. В общем, для вычисления дескриптора особой точки, нужно вычислить 25 фильтров Хаара, в каждом из 16 квадрантов. Всего 400 фильтров Хаара. Учитывая, что фильтру требуется 6 операций, окажется, что дескриптор будет требовать выполнения не менее 2400 операций. После нахождения 25 градиентов точки квадранта вычисляются четыре значения, которые фактически являются компонентами дескриптора:

$$\sum dX, \sum |dX|, \sum dY, \sum |dY|.$$

Две из них – это просто полный градиент по квадранту, а две других – сумма модулей точечных градиентов. На рис.7. показаны значения рассчитываемых величин для разных изображений.

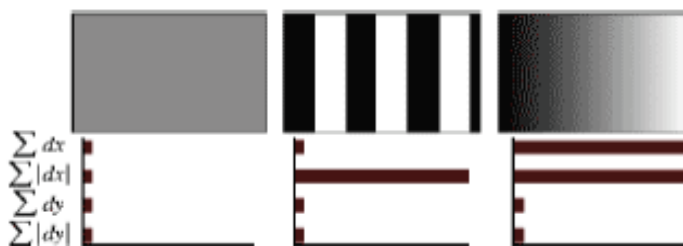


Рисунок 7 – Поведение дескриптора для разных изображений

Для однородных областей – все значения близки к нулю. Для повторяющихся вертикальных полосок – все величины, кроме второй близки к нулю. С увеличением яркости в направлении оси X , первые две компоненты имеют высокие значения.

Четыре компонента на каждый квадрант и 16 квадрантов дают 64 компонента дескриптора для области особой точки. При вводе в массив, значения дескрипторов умножаются на гауссиану, с центром в особой точке и с $\sigma = 3.3s$. Это необходимо для обеспечения большей устойчивости дескриптора к шуму в областях, удаленных от особой точки [22]. В дополнение к дескриптору, для описания точки используется знак следа матрицы Гессе, то есть величина $sign(D_{xx} + D_{yy})$. Для светлых точек на темном фоне, след отрицателен, для

темных точек на светлом фоне – положителен. Подобным образом, *SURF* различает темные и светлые пятна.

Применяя *SURF* к изображению, мы получаем набор дескрипторов, который однозначно идентифицирует стандарт на сцене. Хороший и стабильный результат наблюдается при вращении, изменении освещенности изображения или угла, под которым был сделан снимок. Неплохо метод справляется и с изменением яркости изображения. В Приложении Б (рис.16) приведен результат применения алгоритма *SURF* для поиска особых точек.

1.2.3 Алгоритм ORB

ORB - это ориентированный *FAST* и Rotated *BRIEF* - улучшенная версия и комбинация *FAST*-точечного детектора и двоичных дескрипторов *BRIEF*.

Цель создания *BRIEF*-дескриптора состояла в том, чтобы одни и те же части изображения, взятые с разных точек зрения, были распознаны. Задача состояла в том, чтобы минимизировать количество выполненных вычислений. Алгоритм распознавания сводится к построению случайного леса (*randomize classification trees*) или наивного Байесовского классификатора на некотором обучающем наборе изображений и последующей классификации тестовых областей изображения. В упрощенном варианте может использоваться метод ближайшего соседа для поиска наиболее похожего участка в обучающей выборке. Небольшое количество операций обеспечивается за счет представления характеристического вектора в виде двоичной строки и использования расстояния Хэмминга как меры сходства [15].

Схема построения векторов признаков достаточно проста. Изображение разбивается на патчи. Допустим патч P имеет размеры $S \times S$ пикселей. Из патча выбирается некоторым образом множество пар пикселей $\{(X, Y), \forall X, Y \text{ в окрестности}\}$, для которых строится набор бинарных тестов:

$$\tau(P, X, Y) = \begin{cases} 1, & I(X) < I(Y) \\ 0, & \text{иначе} \end{cases}, \quad (3)$$

где $I(Y), I(X)$ – интенсивность пикселей $Y X$ соответственно. Для каждого патча выбирается множество, содержащее n_d пар точек, которые однозначно определяют набор бинарных тестов. Далее на основании этих тестов строится бинарная строка:

$$f_{n_d}(P) = \sum_{1 \leq i \leq n_d} 2^{i-1} \tau(P, X_i Y_i). \quad (4)$$

Авторы приводят результаты экспериментов (качества распознавания) при выборе пар точек согласно закону равномерного распределения в патче, а также нормального распределения с разными значениями математического ожидания и среднеквадратичного отклонения [5]. Отметим, что при одинаковых условиях проведения экспериментов на некоторых тестовых изображениях точность детектирования с помощью *BRIEF* почти в 1.5 раза выше, чем с использованием *SURF*-дескрипторов [15].

Алгоритм *ORB* быстрее вычисляет особые точки и их дескрипторы, что позволяет использовать его в задачах, где требуется обработка изображений в реальном времени. *ORB* использует алгоритм *FAST* для нахождения ключевых точек. В Приложении Б на рис.17 приведен результат применения алгоритма *ORB* для поиска особых точек.

Краткий обзор наиболее популярных из существующих методов позволяет сделать вывод, что все дескрипторы имеют свои достоинства, проявляя слабую инвариантность к различным преобразованиям изменения масштаба, сдвига, поворота. В данной работе предполагается исследование рассмотренных алгоритмов для решения задачи создания панорамных изображений.

1.3. Сопоставление особых точек

После осуществления поиска и описания особых точек необходимо выбрать алгоритм для их сопоставления. Было решено использовать метод k -средних. Сопоставление может проводиться на основе различных метрик. В данной работе рассматривались метрики Евклида, Хэмминга, Чебышева и Манхэттена.

1.3.1 Метод k -средних

Метод k -средних [23] – это метод кластерного анализа, цель которого является разделение m наблюдений (из пространства R_n) на k кластеров, при этом каждое наблюдение относится к тому кластеру, к центру (центроиду) которого оно ближе всего.

Алгоритм требует задания числа кластеров, обозначаемых через k и $A = \{a_1, \dots, a_k\}$ – для обозначения множества классов. Алгоритм состоит из следующих шагов:

1) каждому из k кластеров произвольным образом назначаются их центры (или эталонные образы) \vec{y}_i ; в качестве этих центров обычно выступают первые k образов обучающей выборки $\vec{y}_i = \vec{x}_i, i = 1, \dots, k$;

2) каждый образ выборки относится к тому классу, расстояние до центра, которого минимально:

$$A_i = \underset{a_j \in A}{\operatorname{argmin}}(s(\vec{x}_i, \vec{y}_i)), i = 1, \dots, M, \quad (5)$$

3) центры кластеров пересчитываются исходя из того, какие образы к каждому из них были отнесены:

$$\vec{y}_i = \frac{1}{M_j} \sum_{(\forall i) A_i = a_j} \vec{x}_i, \quad (6)$$

где M_j – число образов, попавших в класс a_j ;

4) шаги 2) и 3) повторяются, пока не будет достигнута сходимость, т. е. пока классы не перестанут изменяться.

Особенностями метода k -средних являются:

1) Число кластеров заранее не известно и выбирается исследователем заранее.

2) Качество кластеризации зависит от первоначального разбиения.

Алгоритму широко отдается предпочтение из-за его простоты реализации, большой скорости. Действие алгоритма таково, что он стремится минимизировать суммарное квадратичное отклонение точек кластеров от центров этих кластеров. Иначе говоря, это итеративный алгоритм, который делит данное множество пикселей на k кластеров точки, которые являются максимально приближенными к их центрам, а сама кластеризация происходит за счет смещения этих же центров. Также следует оговорить то, что метод k -средних очень чувствительный к шуму, который может существенно исказить результаты кластеризации.

Рисунок 8 демонстрирует пример работы алгоритма.

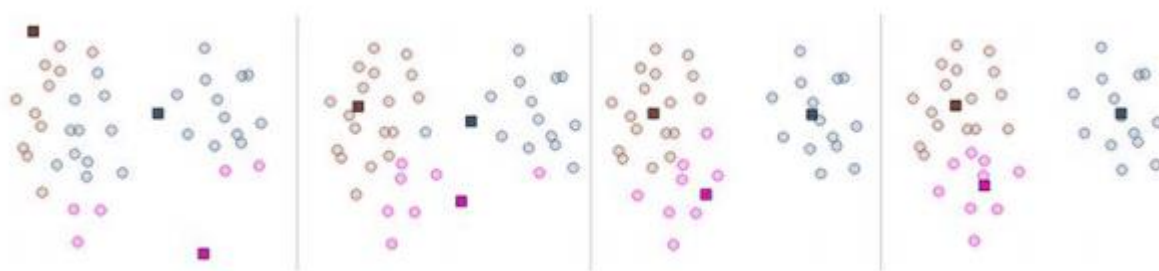


Рисунок 8 – Пример работы алгоритма k -means

1.3.2 Евклидово расстояние

Сравнивая каждый набор дескрипторов из первого изображения с каждым таким набором из второго изображения, находим наилучшее соответствие между особыми точками. Сопоставление рассчитывается на основе евклидова расстояния:

$$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}, \quad (7)$$

где d – расстояние между объектами x и y ; x_i – значение i -свойства объекта x ; y_i – значение i -свойства объекта y ; $i = (1, 2, \dots, n)$.

1.3.3 Расстояние Хэмминга

Определение 1. Мера различия объектов, задаваемых дихотомическими признаками. Определяется с помощью формулы:

$$d_H(X_i X_j) = \sum_{s=1}^p |x_i^{(s)} - x_j^{(s)}| \quad (8)$$

И, следовательно, равно числу несоответствий между значениями соответствующих признаков в рассматриваемых i -м и j -м объектах.

Определение 2. Расстоянием Хэмминга называется число отличающихся бит в двух бинарных векторах.

Расстояние Хэмминга уже довольно широко используется для различных задач, таких как поиск близких дубликатов, распознавание образов, классификация документов, исправление ошибок, обнаружения вирусов и т.д. В более общем случае расстояние Хэмминга применяется к строкам одинаковой длины любых k -ичных алфавитов и служит метрикой разности (функцией, определяющей расстояние в метрическом пространстве) объектов той же размерности. Такую метрику можно представить в виде формулы 9.

$$hamming(x, y) = \sum_{x_i \neq y_i} 1, \quad i = 1, \dots, n \quad (9)$$

где x и y – дескрипторы (бинарные векторы).

1.3.4 Расстояния Манхэттена

Это расстояние является средним разностей по координатам. В большинстве случаев эта мера расстояния приводит к таким же результатам, как и для обычного расстояния Евклида. Однако для этой меры влияние отдельных больших разностей (выбросов) уменьшается (т.к. они не возводятся в квадрат). Формула для расчета манхэттенского расстояния:

$$\rho(x, x') = \sum_i^n (x_i - x'_i)^2, \quad (10)$$

1.3.5 Расстояние Чебышева

Это расстояние может оказаться полезным, когда нужно определить два объекта как «различные», если они различаются по какой-либо одной координате. Расстояние Чебышева вычисляется по формуле:

$$\rho(x, x') = \max(|x_i - x'_i|), \quad (11)$$

1.4. Сшивание изображений

Для сшивания изображений одно из них предварительно нужно будет подвергнуть некоторому геометрическому преобразованию.

Геометрические преобразования – это преобразования графических объектов, в которых направление, масштаб и положение системы координат используются для определения положения точек в пространстве. В этом случае само пространство не подвергается никаким изменениям, то есть структура самого графического изображения сохраняется. Следовательно, для геометрических преобразований тип объекта не важен, поскольку точки участвуют в преобразованиях.

1.4.1 Геометрические преобразования

Основными операциями, связанными с геометрическими преобразованиями двумерных графических объектов, являются:

1. перенос начала координат;
2. масштабирование;
3. поворот осей координат.

Матрица, необходимая для выполнения операции переноса выглядит следующим образом:

$$T = \begin{bmatrix} 1 & 0 & -Tx \\ 0 & 1 & -Ty \\ 0 & 0 & 1 \end{bmatrix}$$

Матрица, необходимая для выполнения операции масштабирования имеет следующий вид:

$$S = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Для выполнения операции поворота используется следующая матрица:

$$R = \begin{bmatrix} \cos w & \sin w & 0 \\ -\sin w & \cos w & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Для совмещения необходимо подвергнуть одно из входных изображений геометрическому преобразованию. Для этого необходимо найти матрицу преобразования (гомографии) 3×3 . Чтобы найти матрицу преобразования, необходимо 4 точки на входном изображении и 4 соответствующих точки на выходном изображении. Среди этих 4-х точек, 3 не должны быть коллинеарными.

Чтобы учесть проективные искажения, можно использовать преобразования в матричной форме в однородных координатах

$$\begin{pmatrix} \lambda_x \\ \lambda_y \\ \lambda \end{pmatrix} = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{pmatrix} \cdot \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix}$$

Матрица гомографии $H=(h_{ij})_{3 \times 3}$ в составе формулы

$$\begin{cases} x = \frac{h_{11}x' + h_{12}y' + h_{13}}{h_{31}x' + h_{32}y' + 1} = \frac{\lambda_x}{\lambda} \\ y = \frac{h_{21}x' + h_{22}y' + h_{23}}{h_{31}x' + h_{32}y' + 1} = \frac{\lambda_y}{\lambda} \end{cases}$$

содержит 8 неизвестных параметров, чтобы ее найти,

нужно по крайней мере 4 пары ключевых на сопоставляемых изображениях. Формирование набора пар ключевых точек и выбор оптимального множества точек из этого множества, которые обеспечивают правильное определение элементов матрицы гомографии – две основные задачи поиска матрицы гомографии.

1.4.2 Алгоритм RANSAC

RANSAC – это метод, который используется для оценки параметров модели на основании случайных выборок. При сопоставлении модель представляет собой матрицу преобразования (гомография). На входе алгоритма имеется два множества дескрипторов, полученных на предыдущем и текущем изображении.

Схема работы RANSAC заключается в многократном повторении следующих этапов:

1. Выбор опорных точек и построение параметров модели, содержащей 8 неизвестных параметров. Для поиска матрицы гомографии необходимо, как минимум, 4 пары особых точек на сопоставляемых изображениях. На основании полученных наборов строится матрица преобразования.

2. Проверка построенной модели. Для каждой точки предыдущего кадра находится проекция на текущем кадре, и выполняется поиск наиболее близкого дескриптора (точки) из множества дескрипторов текущего кадра. Характерная точка помечается как «выброс», если расстояние между проекцией и соответствующим дескриптором текущего изображения больше заданного порога.

3. Замещение модели. Проверяется, является ли построенная модель лучшей среди набора предшествующих моделей.

В результате применения RANSAC строится наилучшая матрица гомографии, которая будет использоваться для поиска ложных соответствий. Схема алгоритма приведена на рис. 9.

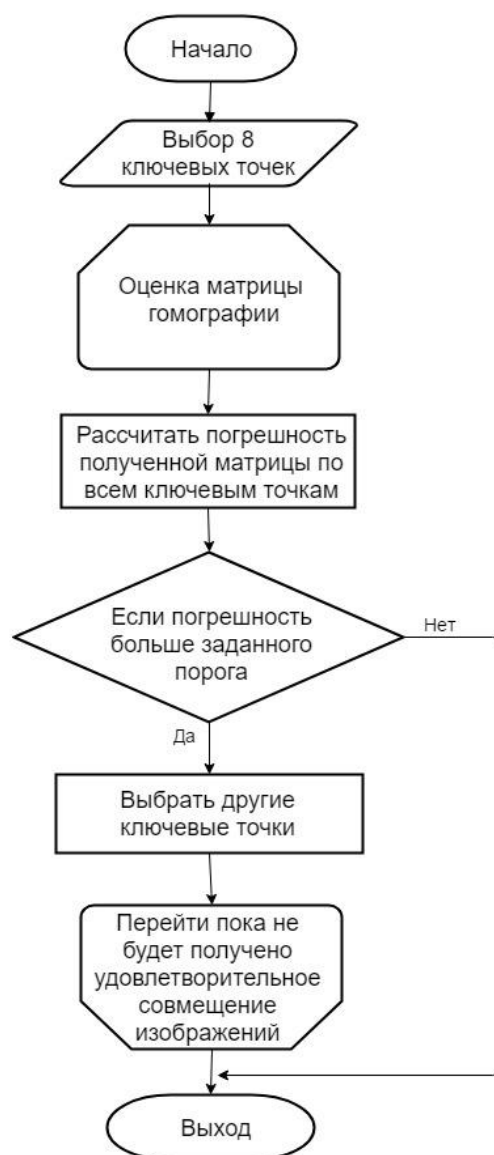


Рисунок 9 – Блок-схема

1.4.3 Генерация трёхмерных объектов

Создание трехмерной модели можно реализовать с помощью графической библиотеки. В данной работе была выбрана библиотека Unity. Подход к созданию и оптимизации трехмерной модели местности может быть применен для разработки интерактивных приложений на базе Unity3D для различного типа приложений (симуляторы, ГИС-системы), что делает его универсальным и полезным.

Unity – это инструмент для разработки двух- и трёхмерных приложений и игр. Важной особенностью данного движка является его

кроссплатформенность. Проекты, созданные в Unity, хорошо собираются под разные устройства – Android, iOS, Windows, Mac, PlayStation, Xbox, многие другие приставки и даже SmartTV. Трехмерные объекты в компьютерной графике вообще, и в Unity в частности, чаще всего представляются полигональными сетками.

Polygonal mesh (полигональная сетка) – набор вершин, граней и ребер, которые задают форму трехмерного объекта. В Unity грани составляют треугольники и четырехугольники, их называют полигонами. Полигон соединяет между собой точки фигуры, тем самым образуя единый, замкнутый объект.

Для визуализации замкнутого панорамного изображения необходимо наложить получившуюся текстуру панорамы на замкнутую фигуру: куб (Cubemap), сферу (Sphere) или цилиндр (Cylinder).

Cubemap представляет собой набор из шести квадратных текстур, которые представляют собой отражение на окружающей среде. Шесть квадратов образуют грани воображаемого куба, который окружает объект; каждая грань представляет собой вид вдоль направления мировых осей (вверх +Y, вниз -Y, вправо -X, влево +X, вперед +Z, назад -Z). Панорамные изображения находятся в этой схеме:

1. Загружается текстура с панорамным изображением;
2. Текстура разбивается на равные блоки;
3. Происходит наложение текстуры на соответствующую грань куба.

Вертикальные и горизонтальные поперечные раскладки, а также столбцы и строки граней cubemap представлена на рис.10.

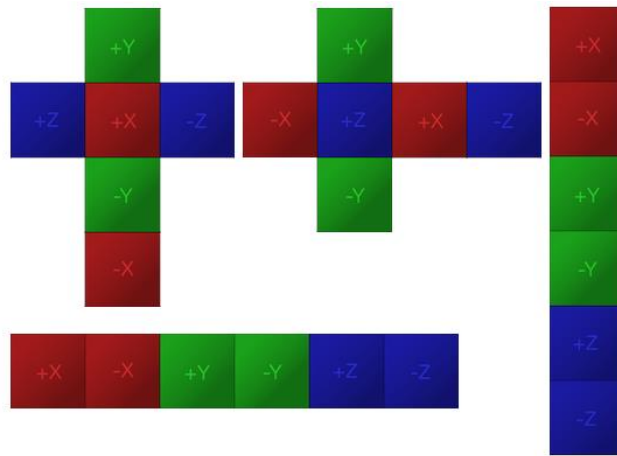


Рисунок 10 – Развёртка Subemap

Другая общая компоновка LatLong (широта-долгота, иногда называют цилиндрической). Развертка цилиндра представлена на рис.11.

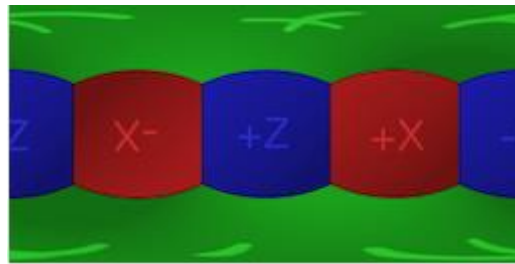


Рисунок 11 – Развёртка цилиндра

Следующая компоновка SphereMap (Сферическая карта среды) изображения. Развертка сферы представлена на рис.12.

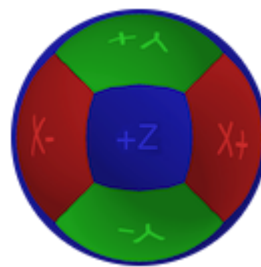


Рисунок 12 – Развёртка сферы

Выводы по Главе 1

В данном разделе сформирована область дальнейшего исследования. Приведено подробное описание трех различных методов поиска и описания особых точек: *SIFT*, *SURF* и *ORB*.

Рассмотрен алгоритм поиска ближайших соседей. Суть кластеризации состоит в том, что все исходные объекты (в данном случае пиксели) разбиваются на несколько непересекающихся групп таким образом, чтобы объекты, попавшие в одну группу, имели сходные характеристики, в то время как у объектов из разных групп эти характеристики должны значительно отличаться. Приведено описание метрик Евклида и Хэмминга.

Описан алгоритм RANSAC для оценки матрицы гомографии. Выбрана среда для создания трехмерной сцены по полученному панорамному изображению.

В данной работе было решено в качестве базового примитива выбрать куб, так как эта форма лучше соответствует форме полученного нами панорамного изображения.

2 РАЗРАБОТКА И РЕАЛИЗАЦИЯ АЛГОРИТМА СОЗДАНИЯ ПАНОРАМНЫХ ИЗОБРАЖЕНИЙ

2.1. Описание библиотеки OpenCV

Библиотека OpenCV [25] имеет модульную структуру, что означает, что пакет включает в себя несколько общих (shared) и статических (static) библиотек. Доступны следующие основные модули, которые использовались в данной работе:

- Core – модуль, определяющий основные структуры данных, в том числе многомерный массив Mat и основные функции, используемые всеми остальными модулями;

- Imgproc – модуль обработки изображений, который включает методы линейной и нелинейной фильтрации, геометрические преобразования изображений (масштабирование, аффинные и перспективные преобразования, текстурирование), преобразования цветового пространства, гистограмм и т.д.;

- Calib3d – модуль, содержащий базовые геометрические алгоритмы стереоскопической визуализации, методы калибровки обычной и стереоскопической камеры, оценки расположения объектов, алгоритмы поиска соответствий для стерео изображений, элементы 3D реконструкции; – Features2d – модуль, предназначенный для поиска ключевых точек изображений, определения и сопоставления дескрипторов;

- Objdetect – модуль, отвечающий за распознавание определенных объектов и экземпляров предопределенных классов (например, лиц, глаз, людей, автомобилей и т.д.);

- Highgui – модуль, содержащий видео кодеки, простой в использовании интерфейс для захвата видео, методы обработки изображений, а также простые возможности пользовательского интерфейса.

– `Gpu` – модуль, хранящий алгоритмы ускорения на GPU из других модулей `OpenCV`. Также присутствуют вспомогательные модули, такие как `FLANN`.

2.1.1 Вычисление детекторов ключевых точек на изображении

В библиотеке `OpenCV` реализовано вычисление следующих дескрипторов ключевых точек: *SIFT*, *SURF*, *ORB* [25] – для использования которых необходимо в исходном коде приложения подключить заголовочный файл (`features2d.hpp`), а в настройках проекта библиотеку (`features2d243(d).lib`).

Классы, содержащие реализацию алгоритмов вычисления данных дескрипторов, унаследованы от базового абстрактного класса `DescriptorExtractor`. Рассмотрим подробнее методы данного класса.

Для создания объекта класса, предназначенного для вычисления дескрипторов ключевых точек заданного типа, используется метод `create`, возвращающий указатель на созданный объект класса.

```
Ptr DescriptorExtractor::create(const string& descriptorExtractorType).
```

В качестве параметра данной функции передается строка, содержащая идентификатор типа дескриптора ("`SIFT`", "`SURF`", "`BRIEF`", "`ORB`").

Для того чтобы вычислить дескрипторы ключевых точек на изображении с использованием созданного детектора, необходимо вызвать метод `compute`.

```
void DescriptorExtractor::compute(const Mat& image, vector& keypoints, Mat& descriptors) const.
```

Рассмотрим параметры данного метода:

`image` – входное изображение.

`keypoints` – массив ключевых точек на изображении.

`descriptors` – вычисленные значения дескрипторов ключевых точек.

Необходимо учесть, что при использовании дескрипторов `SIFT` и `SURF` требуется дополнительно подключить заголовочный файл `nonfree.hpp` и библиотеку `opencv_nonfree243(d).lib`, а также вызвать функцию инициализации данного модуля: `initModule_nonfree()`.

2.1.2 Структура работы компьютерной модели

В настоящей работе рассматривается ряд методов детектирования ключевых точек и построения их дескрипторов с использованием алгоритма кластеризации k-средних.

Дескрипторы можно считать первым шагом, позволяющим выяснить связь между пикселями в цифровом изображении и тем, что человек может вспомнить через несколько минут после просмотра одного или нескольких изображений. В OpenCV используется общий интерфейс DescriptorExtractor, позволяющий переключаться между разными алгоритмами извлечения дескрипторов.

На втором шаге происходит сравнение дескрипторов, определяющих ключевые точки изображений, с целью поиска совпадений. В данной работе для выполнения этого шага применяются три способа (рисунок 13):



Рисунок 13 – Структурная схема работы компьютерной модели

1) с использованием BFMatcher, который перебирает все дескрипторы первого изображения и сравнивает их с дескрипторами второго изображения, т.е. применяет алгоритм полного перебор;

2) с использованием `FlannBasedMatcher`, который при сравнении дескрипторов изображений применяет алгоритм поиска ближайшего соседа из библиотеки `FLANN`;

3) с использованием `FlannBasedMatcher` и функций `findHomography` для нахождения и `perspectiveTransform` для выполнения перспективного преобразования ключевых точек изображений – способ позволяет реагировать и наглядно отображать поворот (или его отсутствие) одного изображения относительно другого.

2.2. Алгоритм создания панорамных изображений

Общая блок – схема алгоритма, приведенная на рис. 14, и состоит из следующих основных этапов:

1. Загрузка изображений.

Загружаются все изображения, хранящиеся в выбранной папке.

2. Для текущей пары загруженных изображений

2.1. Поиск особых точек и получения их дескрипторов на основе одного из методов: *SIFT*, *SURF* и *ORB*.

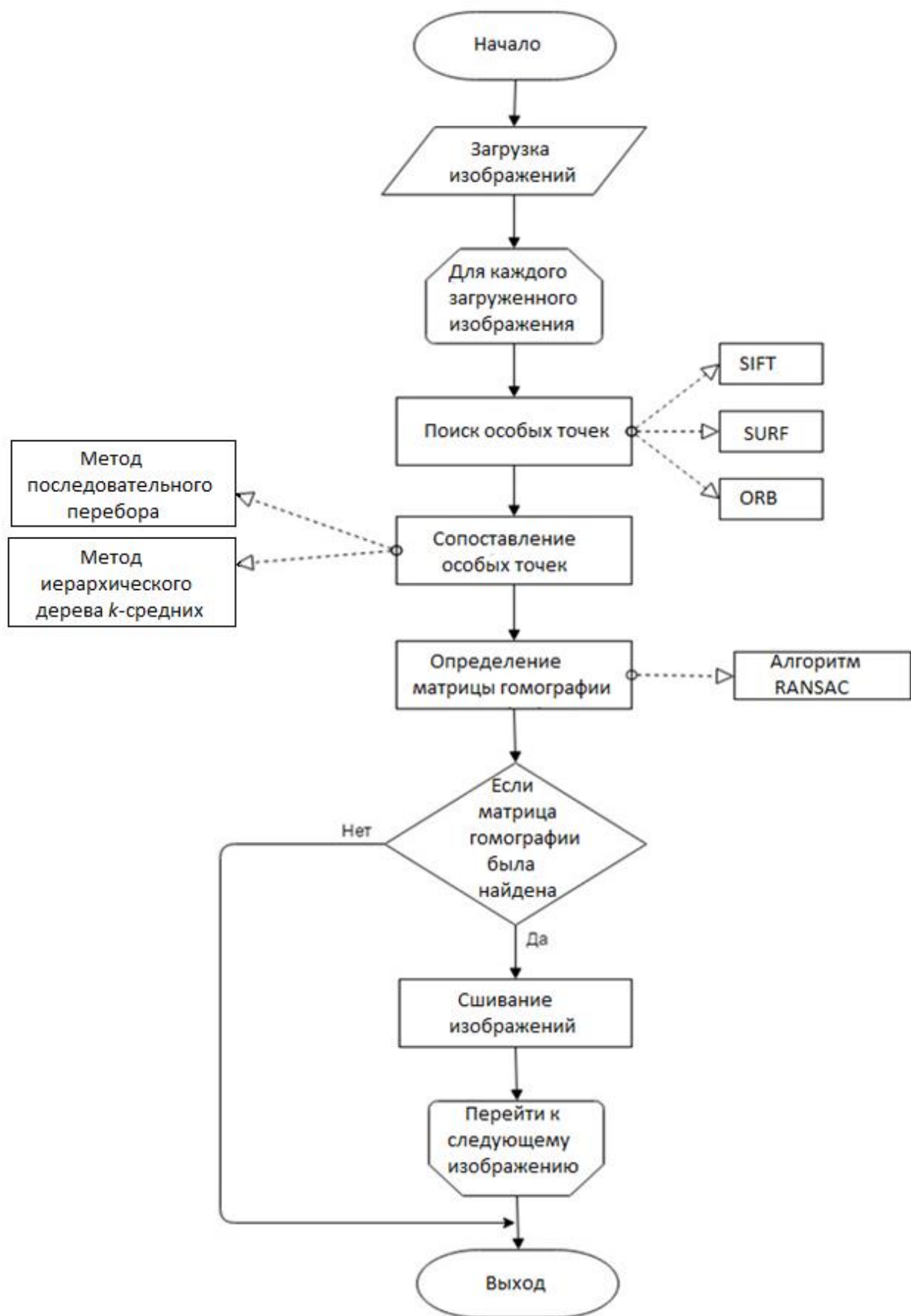


Рисунок 14 – Блок – схема

2.2. Поиск пар соответствующих особых точек.

Важным шагом является сопоставление особых точек. Это необходимо для того, чтобы найти похожие изображения или наличие объектов одного изображения на другом. Для сопоставления исследовались два метода: метод последовательного перебора и метод иерархического дерева k -средних.

Основная идея первого метода состоит в том, что последовательно каждый дескриптор из первого набора сравнивается с каждым дескриптором второго набора, причем каждый раз между ними находится расстояние, чтобы найти ближайшего соседа. Таким образом, сортируются всевозможные комбинации, что может значительно снизить скорость работы с большими наборами данных.

Представление второго метода реализовано на построении иерархического дерева k -средних. Такой ход оптимален при обработке больших объемов данных.

2.3. Оценка матрицы гомографии.

Нахождение матрицы преобразования выполнялось на основе алгоритма RANSAC.

2.4. Сшивание изображений.

После того, как матрица гомографии была найдена, на ее основе рассчитываем новые координаты ключевых точек второго изображения, объединяем изображения, в результате чего получаем панорамное изображение. Переход к следующему изображению.

2.5. Если матрица гомографии не была найдена, тогда невозможность создать панораму, выход системы.

2.3. Выбор среды и языка

Для реализации алгоритма, согласно заданию, был выбран язык программирования C++ с использованием библиотеки OpenCV и среды разработки Visual Studio 2013.

Описание структуры программы

Класс **perspective_transformer** осуществляет поиск особых точек, по найденным особым точкам строит матрицу гомографии и объединяет исходные изображения в панораму.

Класс содержит следующие поля:

```
Mat im, im_transformed, im_perspective_transformed;  
vector <Point2f> points, points_transformed;  
Mat M;
```

Здесь **Mat** является матричным классом OpenCV.

im – первое исходное изображение;

im_transformed – второе исходное изображение;

im_perspective_transformed – перспектива второго изображения.

vector<Point2f> вектор 2-мерных точек в OpenCV;

points – координаты исходных особых точек;

points_transformed – координаты преобразованных точки;

Mat M – расчетная матрица.

Класс содержит следующие методы:

Конструктор - **perspective_transformer**(std::string file_path1, std::string file_path2) – инициализирует исходные изображения.

void **addImage**(std::string file_path) – функция загружает изображение

void **estimate_perspective**() – Метод рассчитывает матрицу гомографии.

void **show_diff**() – метод показывает восстановленное перспективное преобразование действительно корректно.

Чтобы убедиться, что восстановленное преобразование правильное, функция вычисляет разницу между исходным преобразованным изображением и нетрансформированным изображением.

Конструктор **perspective_transformer** выполняет следующие этапы:

1. Преобразование изображений в оттенки серого

```
cvtColor(im, im_g, CV_BGR2GRAY);
```

2. Обнаружение ключевых точек

2.1. На основе ORB детектора

```
vector<KeyPoint> kp, t_kp;  
Mat desc, t_desc, im_g, t_im_g;  
  
OrbFeatureDetector featureDetector;  
OrbDescriptorExtractor featureExtractor;  
  
featureDetector.detect(im_g, kp);  
featureDetector.detect(t_im_g, t_kp);
```

2.2. На основе SIFT детектора

```
SiftFeatureDetector featureDetector;  
SiftDescriptorExtractor featureExtractor;  
featureDetector.detect(im_g, kp);  
featureDetector.detect(t_im_g, t_kp);
```

2.3. На основе SURF детектора

```
SurfFeatureDetector featureDetector;  
SurfDescriptorExtractor featureExtractor;  
featureDetector.detect(im_g, kp);  
featureDetector.detect(t_im_g, t_kp);
```

3. Вычисление дескрипторов (векторы признаков)

```
featureExtractor.compute(im_g, kp, desc);  
featureExtractor.compute(t_im_g, t_kp, t_desc);
```

im_g – входное конвертированное изображение.

kp – массив найденных особых точек на первом изображении.

t_im_g – второе конвертированное изображение.

t_kp – массив найденных особых точек на втором изображении.

desc – вычисленные значения дескриптора на первой изображении.

t_desc – вычисленные значения дескриптора на второй изображении.

4. Поиск соответствующих ключевых точек

4.1. На основе расстояния Евклида и метода к-ближайших соседей

```
BFMatcher(NORM_L2, true).match(descriptors1, descriptors2, matches);
```

Дескриптор из первого набора сравнивается с каждым дескриптором второго набора, причем каждый раз между ними находится расстояние, чтобы найти ближайшего соседа.

4.2. На основе расстояния Хэмминга и метода к-ближайших соседей

```
flann::Index      flannIndex(desc,      flann::LshIndexParams(12,      20,      2),  
cvflann::FLANN_DIST_HAMMING);
```

 — использовали алгоритм локального хэширования (LSH) для выполнения поиска на основе FLANN, который ускоряет поиск ближайших соседей на основе расстояния Хэмминга.

```
Mat match_idx(t_desc.rows, 2, CV_32SC1), match_dist(t_desc.rows, 2, CV_32FC1);  
flannIndex.knnSearch(t_desc, match_idx, match_dist, 2, flann::SearchParams());
```

 —

Выполняет поиск ближайших соседей К для набора точек запроса.

5. Определение «хороших» соответствий

```
vector<DMatch> good_matches;  
for (int i = 0; i < match_dist.rows; i++) {  
    if (match_dist.at<float>(i, 0) < 0.6 * match_dist.at<float>(i, 1)) {  
        DMatch dm(i, match_idx.at<int>(i, 0), match_dist.at<float>(i, 0));  
        good_matches.push_back(dm);  
        points.push_back((kp[dm.trainIdx]).pt);  
        points_transformed.push_back((t_kp[dm.queryIdx]).pt);  
    }  
}
```

match_dist — расстояние между соответствующими точками интереса.

kp — массив найденных особых точек на первой изображении.

t_kp — массив найденных особых точек на второй изображении.

6. Отобразить соответствующие точки интереса

```
Mat im_show;  
drawMatches(im_transformed, t_kp, im, kp, good_matches, im_show); //  
отобразить соответствующие точки интереса  
imshow("ORB matches", im_show); //Показать изображение
```

Отобразить изображение посредством вызова функции imshow.

7. Оценить перспективное преобразование по найденным точкам

```
M = findHomography(points, points_transformed, CV_RANSAC, 2);
```

В основе нахождения этого преобразования лежит алгоритм RANSAC в основе которого лежит итеративная оценка гомографии для случайно выбранных точек (4 на изображении и 4 в сцене).

8. Совмещение изображений, одно из которых подвергается найденному перспективному преобразованию

```
cv::Mat result;  
warpPerspective(im, result, M, cv::Size(im.cols + im_transformed.cols,  
im.rows));
```

result – результат преобразование.

warpPerspective – перспективная преобразования.

im – первое исходное изображение.

im_transformed – второе исходное изображение.

9. Копирование первого исходного изображения в первую половину полного изображения

```
cv::Mat half(result, cv::Rect(0,0,im_transformed.cols, im_transformed.rows));  
im_transformed.copyTo(half);  
im = result;  
imshow("Result", result); //Показать изображение  
waitKey(0);  
}
```

Выводы по Главе 2

В данном разделе было приведено описание основных модулей библиотеки OpenCV.

Описан алгоритм создания панорамных изображений.

Приведено подробное описание структуры разрабатываемого программного обеспечения для создания панорамных изображений.

3 РЕЗУЛЬТАТЫ ЭКСПЕРИМЕНТОВ

3.1. Описание тестовой выборки сопоставление изображений

В ходе работы было проведено исследование и сопоставление алгоритмов *SIFT*, *SURF* и *ORB* поиска особых точек. Подготовлена база изображений, состоящая из 25 различных сцен, каждая из которых содержит по 8-9 изображений. Общее количество изображений равно 200. Особые точки были сопоставлены на основе расстояния Евклида, Чебышева, Манхэттена, Хэмминга. Совмещение изображений осуществлялось по матрице гомографии, найденной на основе алгоритма RANSAC. Результаты скорости и точности работы алгоритмов приведены в таблице 1.

Таблица 1 – Описание тестовой выборки

Алгоритмы	N	%	T
SURF	165	82,5	2,342
SIFT	183	91,5	2,233
ORB	122	61,5	1,06

где N – количество правильно сопоставленных изображений, T – время обработки одного изображения, сек, % – доля правильно сопоставленных изображений от общего количества изображений.

Наибольшей времени обладает алгоритм *ORB*. Алгоритмы *SIFT* и *SURF* имеют большую точность вычисления особых точек и их дескрипторов.

Результаты сопоставления изображений приведены на рис. 18, 19, 20 Приложения Б. На изображениях присутствовали городские сцены и сцены природы. Эксперимент показал, что метод *SIFT* имеет лучшую точность вычисления особых точек и получения их дескрипторов, алгоритм *ORB* находит меньшее количество точек, и его дескрипторы менее точны, чем *SIFT*. На изображениях природы *SIFT* работает лучше: из возможных 55 изображений соединяет 45, алгоритм *ORB* из возможных 33 изображений соединяет 18. На городских сценах точность результатов алгоритмов *SIFT* и *ORB* сопоставима. В

дальнейших экспериментах для поиска особых точек решено было применять алгоритм *SIFT*.

Сравнение результатов по расстоянию Хэмминга, Евклида, Манхэттенского и Чебышева приведены в таблице 2.

Таблица 2 – Результаты сравнения

Методы Метрики	SIFT	SURF	ORB
1. Расстояние Хэмминга	2,233 с	2,342 с	1,06 с
2. Евклидово расстояние	1,832 с	1,219 с	1,24 с
3. Манхэттенское расстояния	0,863 с	1,355 с	0,16 с
4. Расстояния Чебышева	1,051 с	1,325 с	0,147 с

Расстояния Хэмминга сопоставляется на основе *SIFT*-дескриптора за 2,233 секунд, *SURF*-дескриптора за 2,342 секунд, *ORB*-дескриптора за 1,06 секунд. Евклидово расстояние сопоставляется на основе *SIFT*-дескриптора за 1,832 секунд, *SURF*-дескриптора за 1,219 секунд, *ORB*-дескриптора за 1,24 секунд. Манхэттенского расстояния сопоставляется на основе *SIFT*-дескриптора за 0,863 секунд, *SURF*-дескриптора за 1,355 секунд, *ORB*-дескриптора за 0,16 секунд. Расстояния Чебышева сопоставляется на основе *SIFT*-дескриптора за 1,051 секунд, *SURF*-дескриптора за 1,325 секунд, *ORB*-дескриптора за 0,147 секунд.

Параметры устройства, на котором проводилось тестирование. Тестирование проводилось на компьютере следующими конфигурациями приведены в таблице 3.

Таблица 3 – Описание конфигурации

Модель	HP Pavilion dm4-2102er
Экран	14", 1366x768, 16:9
Процессор	Intel Core i5-2430M (два ядра, четыре потока, 2,4/3,0 ГГц, 3072 Кбайт L3)
Память	6144 Мбайт DDR3-1333
Чипсет	Intel HM65 Express
Видеокарта	AMD Radeon HD 6470M + Intel HD Graphics 3000
Жесткий диск	640 Гбайт (5400 об/мин, SerialATA)
Блок питания	90 Вт
Размеры	341x228x32,2 мм
Вес	1,99 кг
Операционная система	Windows 10 Home Premium 64-bit

3.2. Синтез изображений в панораму

Синтез изображений в панорамы осуществлялся по найденной матрице гомографии. Общее количество возможных панорамных изображений равно 183. Число правильно соединенных изображений по 4 равно 160 из 183 возможных, т.е. для этих изображений матрица гомографии были сопоставлены на основе расстояния Хэмминга. Число правильно соединенных изображений по 4 равно 123 из 160 возможных. Результаты скорости работы алгоритма приведены в таблице 4.

Таблица 4 – Результаты соединения изображений основе алгоритма *SIFT*

Соединение изображений	Количество правильно соединенных изображений	Количество возможных соединенных изображений	Время обработки одного изображения, сек
по 2 изображения	160	183	2
по 4 изображения	123	160	3

Таблица 5 – Результаты соединения изображений основе алгоритма *ORB*

Соединение изображений	Количество правильно соединенных изображений	Количество возможных соединенных изображений	Время обработки одного изображения, сек
по 2 изображения	109	122	1
по 4 изображения	69	109	2

Количество возможных соединенных – общее количество изображений, которые могут быть сопоставлены. Количество правильно соединенных – количество правильно объединённых изображений.

Разобраны этапы создания и наполнения 3D-сцены в Unity. Было установлено, как куб использует текстуры для создания оболочки вокруг сцены. Результат панорамного изображения приведен на рисунках 22,23,25,26 Приложения Б.

3.3. Анализ результатов

Представленные результаты экспериментов показали, что алгоритм *SIFT* более точный, чем алгоритм *ORB*. Точки, полученные с помощью алгоритма, являются инвариантными к повороту и масштабированию изображения, устойчивы к изменениям положения наблюдателя, шума и изменениям освещения. Алгоритм *ORB* находит меньшее количество точек, и его дескрипторы менее точны, чем *SIFT*.

Представленные результаты метрик показали, что метрика Хэмминга может применяться для сопоставления бинарных векторов, то есть для детектора *ORB*. Для методов *SIFT*, *SURF* и *ORB* применялись метрики Евклида, Манхэттена и Чебышева. При применении метрики Евклида для сопоставления точек, найденных алгоритмом *SIFT* сопоставилось большее количество точек. Наиболее быстрое сопоставление получилось на основе метрики Манхэттена. Метрика Чебышева показала большую точность сопоставления по сравнению с метрикой Манхэттена. При применении метрики

Евклида для сопоставления точек, найденных алгоритмом *SURF* сопоставилось быстрее по сравнению алгоритмом *SIFT*. Наиболее точное сопоставление получилось на основе метрики Чебышева по сравнению с метрикой Манхэттена. Применении метрики Евклида для сопоставления точек, найденных алгоритмом *ORB* сопоставилось большее количество точек. Метрика Манхэттена показала быстрое сопоставление точек по сравнению с метрикой Чебышева.

Подготовлена база изображений общее количество изображений равно 200. Выполнена профессиональная фотосъёмка из 25 различных сцен для получения качественных изображений для панорам.

Количество изображений, которые были правильно сопоставлены на основе *SIFT*-дескриптора равно – 183, *SURF*-дескриптора – 165, *ORB*-дескриптора – 122. Матрица гомографии правильно построилась для 160 изображений из 183 на основе дескриптора *SIFT*, для 109 изображений из 122 на основе дескриптора *ORB*, для 160 изображений из 165 на основе дескриптора *SURF*.

Время обработки одного изображения методом *SIFT* – 2,233 секунд, *SURF* – 2,342 секунд, *ORB* – 1,06 секунд. По результатам экспериментов алгоритм *ORB* обладает большей скоростью, чем *SIFT*. Алгоритм *SIFT* имеет большую точность вычисления особых точек и расчета их дескрипторов.

Отметим, что хороших результатов удаётся добиться только при низком уровне посторонних шумов на изображении. При этом данный алгоритм достаточно сильно зависит от масштаба картинок. При сопоставлении принимаются только те результаты, когда максимальный коэффициент корреляции превышает данный порог. Некоторые хорошие соответствия имеют низкий, а некоторые ложные соответствия имеют высокий коэффициент корреляции.

**ЗАДАНИЕ ДЛЯ РАЗДЕЛА
«ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И
РЕСУРСОСБЕРЕЖЕНИЕ»**

Студенту:

Группа	ФИО
8ИМ5А	Толеген Молдир Орынбасаркызы

Институт	Кибернетики	Кафедра	Вычислительной техники
Уровень образования	магистрант	Направление/специальность	09.04.02

Исходные данные к разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»:

1. Стоимость ресурсов научного исследования (НИ): материально-технических, энергетических, финансовых, информационных и человеческих	На основании информации, представленной в научных статьях и публикациях, аналитических материалах, статистических бюллетенях и изданиях, нормативно-правовых документах, определить методику расчета экономической эффективности.
2. Нормы и нормативы расходования ресурсов	
3. Используемая система налогообложения, ставки налогов, отчислений, дисконтирования и кредитования	

Перечень вопросов, подлежащих исследованию, проектированию и разработке:

1. Оценка коммерческого и инновационного потенциала НИИ	Оценка ресурсной, социальной эффективности НИИ и потенциальных рисков.
2. Разработка устава научно-технического проекта	
3. Планирование процесса управления НИИ: структура и график проведения, бюджет, риски и организация закупок	
4. Определение ресурсной, финансовой, экономической эффективности	

Перечень графического материала (с точным указанием обязательных чертежей):

1. Перечень работ и продолжительность их выполнения
2. Трудозатраты на выполнение проекта
3. Линейный график работ
4. Нарастание технической готовности работы и удельный вес каждого этапа
5. Расчет затрат на материалы
6. Затраты на заработную плату
7. Затраты на электроэнергию технологическую
8. Смета затрат на разработку проекта
9. Оценки научно-технического уровня НИР

Дата выдачи задания для раздела по линейному графику	
--	--

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Попова Светлана Николаевна			

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8ИМ5А	Толеген Молдир Орынбасаркызы		

4 Финансовый менеджмент, ресурсоэффективность и ресурсосбережение

Технико-экономическое обоснование научно-исследовательских работ проводится с целью определения и анализа трудовых и денежных затрат, направленных на их реализацию, а также уровня их научно-технической результативности.

4.1. Организация и планирование работ

При организации процесса реализации конкретного проекта необходимо оптимально планировать занятость каждого из его участников и сроки проведения отдельных работ.

На данном этапе составляется полный перечень проводимых работ, и определяются их исполнители и оптимальная продолжительность. Результатом планирования работ является сетевой, либо реализации проекта. Наиболее удобным, простым и наглядным способом для этих целей является использование линейного графика. Для его построения составим перечень работ и соответствие работ своим исполнителям, продолжительность выполнения этих работ и сведем их в таблицу 1.

Таблица 1 – Перечень работ и продолжительность их выполнения

Этапы работы	Исполнители	Загрузка исполнителей
Постановка целей и задач, получение исходных данных	НР, И	НР – 60% И – 40%
Подбор и изучение материалов по тематике	НР, И	НР – 30% И – 100%
Разработка календарного плана	НР, И	НР – 40% И – 60%
Обсуждение литературы	НР, И	НР – 20% И – 100%
Выбор алгоритмов для решения поставленной задачи	НР, И	НР – 20% И – 100%
Планирование архитектуры программного обеспечения	НР, И	НР – 10% И – 100%
Разработка программного обеспечения	И	И – 100%
Тестирование и отладка программного обеспечения	И	И – 100%
Оформление расчетно-пояснительной записки	И	И – 100%
Подведение итогов	НР, И	НР – 70% И – 100%

НР – научный руководитель. И – исполнитель.

4.1.1 Продолжительность этапов работ

Расчет продолжительности этапов работ осуществляется двумя методами:

- технико-экономическим;
- опытно-статистическим.

В данном случае используется опытно-статистический метод, который реализуется двумя способами:

- аналоговый;
- экспертный (вероятностный).

Для определения ожидаемого значения продолжительности работ $t_{ож}$ применяется вероятностный метод – метод двух оценок t_{min} и t_{max} .

$$t_{ож} = \frac{3 \cdot t_{min} + 2 \cdot t_{max}}{5}, \quad (1)$$

где t_{min} – минимальная трудоемкость работ, чел/дн.;

t_{max} – максимальная трудоемкость работ, чел/дн.

Для выполнения перечисленных в таблице 1 работ требуются специалисты:

- инженер – в его роли действует исполнитель НИР (ВКР);
- научный руководитель.

Для построения линейного графика необходимо рассчитать длительность этапов в рабочих днях, а затем перевести в календарные дни. Расчет продолжительности выполнения каждого этапа ведется по формуле:

$$T_{РД} = \frac{t_{ож}}{K_{ВН}} \cdot K_{Д}, \quad (2)$$

где $t_{ож}$ – трудоемкость работы, чел/дн.;

$K_{ВН}$ – коэффициент выполнения работ ($K_{ВН} = 1$);

$K_{Д}$ – коэффициент, учитывающий дополнительное время на компенсации и согласование работ ($K_{Д} = 1.2$).

Расчет продолжительности этапа в календарных днях ведется по формуле:

$$T_{КД} = T_{РД} \cdot T_{К}, \quad (3)$$

где $T_{РД}$ – продолжительность выполнения этапа в рабочих днях;

$T_{КД}$ – продолжительность выполнения этапа в календарных днях;

$T_{К}$ – коэффициент календарности.

Коэффициент календарности рассчитывается по формуле:

$$T_{К} = \frac{T_{КАЛ}}{T_{КАЛ} - T_{ВД} - T_{ПД}}, \quad (4)$$

где $T_{КАЛ}$ – календарные дни ($T_{КАЛ} = 365$);

$T_{ВД}$ – выходные дни ($T_{ВД} = 52$);

$T_{ПД}$ – праздничные дни ($T_{ПД} = 10$).

$$T_{К} = \frac{365}{365 - 52 - 10} = 1,205.$$

Таблица 2 – Трудозатраты на выполнение проекта

Этап	Исполнители	Продолжительность работ, дни			Длительность работ, чел/дн.			
		t_{min}	t_{max}	$t_{ож}$	$T_{РД}$		$T_{КД}$	
					НР	И	НР	И
Постановка целей и задач, получение данных	НР, И	1	3	1,8	1,296	0,864	1,568	1,04
Подбор и изучение материалов по тематике	НР, И	10	15	12	4,32	14,4	5,2	17,35
Разработка календарного плана	НР, И	2	4	2,8	1,34	2,016	1,61	2,429
Обсуждение литературы	НР, И	2	5	3,2	0,768	3,84	0,925	4,627
Выбор алгоритмов для решения поставленных задач	НР, И	11	13	11,8	3,89	12,98	4,67	15,58
Планирование архитектуры программного обеспечения	НР, И	15	22	17,8	2,136	21,36	2,57	25,738
Разработка программного обеспечения	И	3	5	3,8	1,67	3,34	2	4
Тестирование и отладка программного обеспечения	И	2	5	3,2	1,06	3,52	1,27	4,22
Оформление расчетно-пояснительной записки	И	5	7	5,8	0,00	6,96	0	8,3868
Подведение итогов	НР, И	2	4	2,8	2,35	3,36	2,83	4,0488
Итого:				65	18,83	72,64	22,643	87,4196

4.1.2 Расчет нарастания технической готовности работ

Величина нарастания технической готовности работы показывает процент выполнения работы на каждом этапе. Расчетная формула:

$$H_i = \frac{t_{Hi}}{t_0} \cdot 100\%, \quad (5)$$

где t_{Hi} – нарастающая трудоемкость с момента начала работы i -го этапа;

t_0 – общая трудоемкость.

Общая трудоемкость вычисляется по формуле:

$$t_0 = \sum_{i=1}^n t_{0Жi}, \quad (6)$$

где $t_{0Жi}$ – ожидаемая продолжительность i -го этапа.

Удельный вес каждого этапа Y_i определяется по формуле:

$$Y_i = \frac{t_{0Жi}}{t_0} \cdot 100\% \quad (7)$$

Результаты вычислений H_i и Y_i отражены в таблице 4.3.

Таблица 3 – Нарастание технической готовности и удельный вес этапов

Этап	H_i , %	Y_i , %
Постановка целей и задач, получение исходных данных	1,51	1,51
Подбор и изучение материалов по тематике	11,59	10,08
Разработка календарного плана	13,949	2,35
Обсуждение литературы	16,63	2,689
Выбор алгоритмов для решения поставленных задач	27,39	10,756
Планирование архитектуры программного обеспечения	42,35	14,958
Разработка программного обеспечения	74,285	31,93
Тестирование и отладка программного обеспечения	92,77	18,487
Оформление расчетно-пояснительной записки	97,647	4,87
Подведение итогов	100	2,35

Таблица 4 – Линейный график работ

Этап	$T_{кд}$		февраль			март			апрель			май			Июнь		
	ННР	И	10	20	30	10	20	30	10	20	30	10	20	30	10	20	30
1	1,568	1,04															
2	5,2	17,35															
3	1,61	2,429															
4	0,925	4,627															
5	4,67	15,58															
6	2,57	25,738															
7	2	4															
8	1,27	4,22															
9	0	8,3868															
10	2,83	4,0488															



4.2. Расчет сметы затрат на разработку ПО и проведение эксперимента

В состав затрат на разработку ПО и проведение эксперимента включается стоимость всех расходов, необходимых для реализации комплекса работ, составляющих содержание данной разработки. Расчет сметной стоимости на выполнение данной разработки производится по следующим статьям затрат:

- материалы и покупные изделия;
- основная заработная плата;
- отчисления в социальные фонды;
- расходы на электроэнергию;
- амортизационные отчисления;
- работы, выполняемые сторонними организациями;
- прочие расходы.

4.2.1. Расчет затрат на материалы

К данной статье так как основным рабочим инструментом при разработке данного проекта выступал персональный компьютер, затраты на материалы и оборудование составляют 0 рублей в связи с отсутствием необходимости в расходных материалах.

4.2.2. Расчет основной заработной платы

В ТПУ оклады распределены в соответствии с занимаемыми должностями, например, ассистент, ст. преподаватель, доцент, профессор. Базовый оклад Зб определяется исходя из размеров окладов, определенных штатным расписанием предприятия. Величина месячного оклада научного руководителя (МО_{НР}) получена из открытых данных, размещенных на официальном сайте Национального исследовательского Томского политехнического университета. Величина месячного оклада инженеров (МО_И) берется как месячный оклад инженера кафедры.

Среднедневная тарифная заработная плата рассчитывается по формуле:

$$ЗП_{\text{дн-т}} = MO/N, \quad (5.6)$$

где MO – месячный оклад, руб.;

N – количество рабочих дней в месяц, при шестидневной рабочей неделе – $N = 24,91$, а при пятидневной рабочей неделе – $N = 20,58$.

Среднедневная тарифная заработная плата научного руководителя равна

$$ЗП_{\text{дн-т}} = \frac{26\,300}{24,91} = 1055,8 \frac{\text{руб.}}{\text{раб. день}}$$

А среднедневная тарифная заработная плата инженеров равна

$$ЗП_{\text{дн-т}} = \frac{16\,500}{20,58} = 801,7 \frac{\text{руб.}}{\text{раб. день}}$$

Затраты времени на выполнение работы по каждому исполнителю брались из таблицы 2: Также был принят во внимание коэффициент, учитывающий коэффициент по премиям $K_{\text{доп.ЗП}} = 1,188$, $K_{\text{ПР}} = 1,1$ и районный коэффициент $K_{\text{РК}} = 1,3$ ($K = 1 * K_{\text{ПР}} * K_{\text{РК}} * K_{\text{РК}} = 1 * 1,1 * 1,3 * 1,188 = 1,699$). Расчеты затрат на основную заработную плату приведены в таблице 5.

Таблица 5– Затраты на основную заработную плату

Исполнитель	Оклад, руб./мес.	Среднедневная ставка, руб./час	Затраты времени, дни	Коэффициент	Фонд з/платы, руб.
НР	26 300	1055,8	18	1,699	32 288,47
И	16 500	801,7	72	1,699	98 070,35
Итого:					130 358,82

4.2.3. Расчет отчислений от заработной платы

Затраты по этой статье составляют отчисления в социальные фонды.

Отчисления по заработной плате определяются по следующей формуле:

$$C_{\text{соц}} = K_{\text{соц}} * C_{\text{зп}}, \quad (10)$$

где $K_{\text{соц}}$ – коэффициент, учитывающий размер отчислений из заработной платы. Данный коэффициент составляет 30,2% от затрат на заработную плату и включает в себя:

- отчисления в пенсионный фонд;

- на социальное страхование;
- на медицинское страхование.

Итак, отчисления из заработной платы составили:

$$C_{\text{соц}} = 0,302 * 130\,358,82 = \mathbf{39\,368,36 \text{ руб.}}$$

4.2.4. Расчет затрат на электроэнергию

Данный вид расходов включает в себя затраты на электроэнергию при работе оборудования. Затраты на электроэнергию при работе оборудования для технологических целей рассчитываются по формуле:

$$\mathcal{E}_{\text{ОБ}} = P_{\text{ОБ}} \cdot C_{\mathcal{E}} \cdot t_{\text{ОБ}}, \quad (11)$$

где $\mathcal{E}_{\text{ОБ}}$ – затраты на электроэнергию, потребляемую оборудованием, руб.;

$P_{\text{ОБ}}$ – мощность, потребляемая оборудованием, кВт;

$C_{\mathcal{E}}$ – тарифная цена за 1 кВт·час, $C_{\mathcal{E}} = 5,8 \text{ руб/кВт·час}$;

$t_{\text{ОБ}}$ – время работы оборудования, час.

Время работы оборудования вычисляется на основе данных для $T_{\text{рд}}$ таблицы 4 для инженера из расчета, что продолжительность рабочего дня равна 8 часов.

Мощность, потребляемая оборудованием, определяется по формуле:

$$P_{\text{ОБ}} = P_{\text{уст.об}} \cdot K_{\text{с}}, \quad (12)$$

где $P_{\text{уст.об}}$ – установленная мощность оборудования, кВт;

$K_{\text{с}}$ – коэффициент загрузки, зависящий от средней степени использования номинальной мощности.

Для технологического оборудования малой мощности $K_{\text{с}} = 1$.

Затраты на электроэнергию для технологических целей приведены в таблице 6.

Таблица 6 – Затраты на электроэнергию для технологических целей

Наименование оборудования	Время работы оборудования $t_{об}$, час	Потребляемая мощность $P_{об}$, кВт	Затраты $\text{Э}_{об}$, руб.
Персональный компьютер	144	0,09	75,168
Итого:			75,168

4.2.5. Расчет амортизационных расходов

В статье «Амортизационные отчисления» от используемого оборудования рассчитывается амортизация за время выполнения работы для оборудования, которое имеется в наличии.

Амортизационные отчисления рассчитываются на время использования ЭВМ по формуле:

$$C_{AM} = \frac{N_A \cdot C_{об}}{F_D} \cdot t_{BT} \cdot n, \quad (4.13)$$

где N_A – годовая норма амортизации, $N_A = 25\%$;

$C_{об}$ – цена оборудования, $C_{об} = 50000$ руб.;

F_D – действительный годовой фонд рабочего времени, $F_D = 2416$ часов;

t_{BT} – время работы вычислительной техники при создании программного продукта, $t_{BT} = 984$ часа;

n – число задействованных ПЭВМ, $n = 1$.

Итак, затраты на амортизационные отчисления составили:

$$C_{ам} = 5091,06 \text{ руб}$$

4.2.6. Расчет расходов на услуги связи

Расходы на услуги связи определены наличием подключения к сети Интернет на ПК, которые использовались в данной работе.

Ежемесячная оплата, согласно тарифу TRUnet, составляет 350 рублей. В соответствии с таблицей 2, трудоемкость выполняемой задачи составляет четыре

календарных месяца. Таким образом, сумма расходов на услуги связи составляет $4 \cdot 350 = 1400$ руб. Общая сумма расходов $C_{св} = 1400$ руб.

4.2.7. Расчет прочих расходов

В статье «Прочие расходы» отражены расходы на разработку проекта, которые не учтены в предыдущих статьях.

Прочие расходы составляют 10 % от единовременных затрат на выполнение технического продукта и проводятся по формуле:

$$C_{\text{проч}} = (C_{\text{зп}} + C_{\text{соц}} + C_{\text{эл.об.}} + C_{\text{ам}}) \cdot 0,1 \quad (14)$$

$$C_{\text{проч}} = (130\,358,82 + 39\,368,36 + 75,168 + 5091,06) \cdot 0,1 = 17\,489,34$$

4.2.8. Расчет общей себестоимости разработки

Проведя расчет сметы затрат на разработку, можно определить общую стоимость разработки проекта «Алгоритм и программные средства для создания панорамных изображений».

Таблица 7 – Смета затрат на разработку проекта

Статья затрат	Условное обозначение	Сумма, руб.
1 Материалы и покупные изделия	$C_{\text{мат}}$	0
2 Основная заработная плата	$C_{\text{зп}}$	130 358,82
3 Отчисления в социальные фонды	$C_{\text{соц}}$	39 368,36
4 Расходы на электроэнергию	Э	75,168
5 Амортизационные отчисления	$C_{\text{ам}}$	5091,06
6 Расходы на услуги связи	$C_{\text{св}}$	1400
7 Прочие расходы	$C_{\text{проч}}$	17 489,34
Итого:		193 782,74

Таким образом, расходы на разработку составили $C = 193\,782,74$ руб.

4.3. Оценка экономической эффективности проекта

Выполнение научно-исследовательских работ оценивается уровнями достижения экономического, научного, научно-технического и социального эффектов.

Научный эффект характеризует получение новых научных знаний и отображает прирост информации, предназначенной для внутринаучного потребления. Научно-технический эффект характеризует возможность использования результатов в других проектах и обеспечивает получение информации, необходимой для создания новой техники. Экономический эффект характеризуется в частности выраженной в стоимостных показателях экономией живого общественного труда, а также в возможности применения полученных знаний для создания новых разработок. Социальный эффект проявляется в улучшении условий труда.

Экономическая эффективность проекта от внедрения результатов проводимых исследований связан с актуальностью и значимостью вопроса замещения импортной продукции в системах технического зрения и создания панорамных изображений.

Подводя итог вышесказанному, экономический эффект от реализации проекта может быть выражен в снижении затрат на покупку подобных систем от сторонних разработчиков, а так же повышение эффективности обработки изображений за счет экономии времени при замене ручной обработки на автоматизированную.

4.3.1 Оценка научно-технического уровня НИР

Научно-технический уровень характеризует, в какой мере выполнены работы и обеспечивается научно-технический прогресс в данной области. Для оценки научной ценности, технической значимости и эффективности, планируемых и выполняемых НИР, используется метод бальных оценок. Бальная оценка заключается в том, что каждому фактору по принятой шкале присваивается определенное количество баллов. Обобщенную оценку проводят

по сумме баллов по всем показателям или рассчитывают по формуле. На этой основе делается вывод о целесообразности НИР.

Сущность метода заключается в том, что на основе оценок признаков работы определяется коэффициент ее научно-технического уровня по формуле:

$$K_{НТУ} = \sum_{i=1}^3 R_i \cdot n_i, \quad (4.15)$$

где $K_{НТУ}$ – коэффициент научно-технического уровня;

R_i – весовой коэффициент i -го признака научно-технического эффекта;

n_i – количественная оценка i -го признака научно-технического эффекта, в баллах.

Таблица 8 – Весовые коэффициенты признаков НТУ

Признак НТУ	Примерное значение весового коэф-та n_i	
1 Уровень новизны	Систематизируются и обобщаются сведения, определяются пути дальнейших исследований	0,4
2 Теоретический уровень	Разработка способа (алгоритм, программа мероприятий, устройство, вещество и т.п.)	0,1
3 Возможность реализации	Время реализации в течение первых лет	0,5

Таблица 9 – Баллы для оценки уровня новизны

Уровень новизны	Характеристика уровня новизны	Баллы
Принципиально новая	Новое направление в науке и технике, новые факты и закономерности, новая теория, вещество, способ	8 – 10
Новая	По-новому объясняются те же факты, закономерности, новые понятия дополняют ранее полученные результаты	5 – 7
Относительно новая	Систематизируются, обобщаются имеющиеся сведения, новые связи между известными факторами	2 – 4
Не обладает новизной	Результат, который ранее был известен	0

Таблица 10 – Баллы значимости теоретических уровней

Теоретический уровень полученных результатов	Баллы
1 Установка закона, разработка новой теории	10
2 Глубокая разработка проблемы, многоспектральный анализ, взаимодействия между факторами с наличием объяснений	8
3 Разработка способа (алгоритм, программа и т. д.)	6
4 Элементарный анализ связей между фактами (наличие гипотезы, объяснения версии, практических рекомендаций)	2
5 Описание отдельных элементарных факторов, изложение наблюдений, опыта, результатов измерений	0,5

Таблица 11 – Возможность реализации научных, теоретических результатов по времени и масштабам

Время реализации	Баллы
В течение первых лет	10
От 5 до 10 лет	4
Свыше 10 лет	2

Результаты оценок признаков научно-технического уровня приведены в таблице 12.

Таблица 12 – Количественная оценка признаков НИОКР

Признак научно-технического эффекта НИР	Характеристика признака НИОКР	R_i
Уровень новизны	Систематизируются, обобщаются имеющиеся сведения, новые связи между известными факторами	00,4
Теоретический уровень	Разработка способа (алгоритм, программа мероприятий, устройство, вещество и т.п.)	00,1
Возможность реализации	Время реализации 5-10 лет	00,5

Обоснование оценки признаков НИОКР приводится в таблице 13.

Таблица 13 – Оценки научно-технического уровня НИР

Фактор НТУ	Значимость	Уровень фактора	Выбранный балл	Обоснование выбранного балла
Уровень новизны	0,4	Относительно новая	4	Облегчит обработку построения панорамных изображений
Теоретический уровень	0,1	Разработка способа	6	Разработана и реализован алгоритм, решающий поставленные в рамках задания задачи
Возможность реализации	0,5	В течение первых лет	7	Полученный продукт находится на стадии тестирования и внедрения

Исходя из оценки признаков НИОКР, показатель научно-технического уровня для данного проекта составил:

$$K_{\text{нту}} = 0,4*4 + 0,1*6 + 0,5*7 = 1,6 + 0,6 + 3,5 = 5,7$$

Таким образом, исходя из данных в таблице 14, проект «Алгоритм и программные средства для создания панорамных изображений» имеет средний уровень научно-технического эффекта.

Таблица 14– Оценка уровня научно-технического эффекта

Уровень НТЭ	Показатель НТЭ
Низкий	1-4
Средний	4-7
Высокий	8-10

**ЗАДАНИЕ ДЛЯ РАЗДЕЛА
«СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»**

Студенту:

Группа	ФИО
8ИМ5А	Толеген Молдир Орынбасаркызы

Институт	ИК	Кафедра	Информационные системы и технологии
Уровень образования	Магистр	Направление/специальность	09.04.02 «Информационные системы и технологии»

Исходные данные к разделу «Социальная ответственность»:

<p>1. <i>Характеристика объекта исследования (вещество, материал, прибор, алгоритм, методика, рабочая зона) и области его применения.</i></p>	<p>1. <i>Объектом исследования является задача создания панорамных изображений. Алгоритм создание панорамных изображений позволяет создавать трехмерные модели окружающего пространства, что может быть применено в сферах туризма, картографии, космосе и т.д..</i></p>
---	--

Перечень вопросов, подлежащих исследованию, проектированию и разработке:

<p>1. <i>Профессиональная социальная безопасность.</i></p> <p>1.1 <i>Анализ вредных и опасных факторов, которые может создать объект исследования.</i></p> <p>1.2 <i>Анализ вредных и опасных факторов, которые могут возникнуть на рабочем месте при проведении исследований.</i></p> <p>1.3 <i>Обоснование мероприятий по защите исследователя от действия опасных и вредных факторов.</i></p>	<p>1.1. <i>В качестве вредных факторов при разработке и при реализации выделены:</i></p> <ul style="list-style-type: none"> - <i>Нарушение параметров микроклимата;</i> - <i>Недостаточная освещенность рабочей зоны;</i> - <i>Монотонный режим работы.</i> <p>1.2. <i>В качестве опасных факторов при разработке и при реализации выделены:</i></p> <ul style="list-style-type: none"> - <i>Опасность поражения электрическим током;</i> - <i>Опасность возникновения пожара.</i> <p>1.3 <i>Обоснование мероприятий по защите исследователя от действия опасных и вредных факторов.</i></p>
<p>2. <i>Экологическая безопасность.</i></p> <p>2.1 <i>Анализ влияния объекта исследования на окружающую среду.</i></p> <p>2.2 <i>Анализ влияния процесса исследования на окружающую среду.</i></p> <p>2.3 <i>Обоснование мероприятий по защите окружающей среды.</i></p>	<p>2.1 <i>Программное обеспечение не связана с производством, следовательно, влияние на окружающую среду минимально.</i></p> <p>2.2. <i>Рассмотрена утилизация бумажных отходов и неисправных комплектующих.</i></p> <p>2.3 <i>Обоснование мероприятий по защите окружающей среды.</i></p> <p>2.4 <i>Специфика программы для защиты окружающей среды:</i></p> <ul style="list-style-type: none"> - <i>Получить панораму с мест биоразнообразии;</i> - <i>Получить общую панораму мусорных свалок.</i>
<p>3. <i>Безопасность в чрезвычайных ситуациях.</i></p> <p>3.1 <i>Анализ вероятных ЧС, которые может инициировать объект исследований.</i></p>	<p>3.1 <i>Вероятные ЧС, инициируемые объектом исследования:</i></p> <ul style="list-style-type: none"> - <i>Пожар;</i> - <i>Социальные ЧС(терроризма);.</i>

<p>3.2 Анализ вероятных ЧС, которые могут возникнуть на рабочем месте при проведении исследований.</p> <p>3.3 Обоснование мероприятий по предотвращению ЧС и разработка порядка действия в случае возникновения ЧС.</p>	<ul style="list-style-type: none"> - Обрушение зданий. <p>3.2 Вероятные ЧС, возникающие на рабочем месте:</p> <ul style="list-style-type: none"> - Пожары и взрывы; - Получить общую панораму СЧС. - Внезапное обрушение зданий. <p>3.3 Мероприятия по предотвращению наиболее типичной ЧС – пожара, согласно нормативным документам:</p> <ul style="list-style-type: none"> - НПБ 105-03; - ППБ 01–03.
<p>4 Правовые и организационные вопросы обеспечения безопасности.</p> <p>4.1 Специальные (характерные для проектируемой рабочей зоны) правовые нормы трудового законодательства.</p> <p>4.2 Организационные мероприятия при компоновке рабочей зоны.</p>	<p>4.1 Описание правовых норм для работ, связанных с работой за ПЭВМ согласно следующим документам:</p> <ul style="list-style-type: none"> - Трудовой кодекс Российской Федерации" от 30.12.2001 N 197-ФЗ (ред. от 30.12.2015). <p>4.2 Специфика применение:</p> <ul style="list-style-type: none"> - Позволяет экономить время и усилия, затрачиваемые на распознавания данных вручную; - экономия средств бюджета.

Дата выдачи задания для раздела по линейному графику	
---	--

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент каф. ЭБЖ	Акулов Петр Анатольевич			

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8ИМ5А	Толеген Молдир Орынбасаркызы		

5 СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ

Введение

Разработанный в рамках магистерской диссертации проект является задачей создания панорамных изображений. Алгоритм создания панорамных изображений позволяет создавать трехмерные модели окружающего пространства, что может быть применено в сферах туризма, картографии, космосе и т.д. Разработка программы велась исключительно при помощи компьютера. Сферы применения разработки очень широки: от нужд обычных пользователей до специализированных задач.

Раздел включает в себя анализ опасных и вредных факторов создаваемых алгоритмом влияния на окружающую среду на рабочем месте при проведении исследований и эксплуатации. А также организационные по защите окружающей среды, правовые и организационные вопросы обеспечения безопасности, мероприятий по предотвращению ЧС.

5.1. Профессиональная социальная безопасность

Работа связана с анализом и теоретической разработкой алгоритмов поиска особых точек, сопоставление особых точек и сшивание изображений. В ходе выполнения работы и при дальнейшей эксплуатации разработанных программных средств могут возникнуть следующие вредные и опасные факторы:

- 1) несоответствия нормам условий освещения рабочего места;
- 2) несоответствия нормам условий температуры рабочего места;
- 3) возможности возникновения пожара
- 4) монотонность работы при разработке алгоритма.
- 5) повышенное напряжение электрической цепи.

Воздействие неблагоприятных факторов приводит к снижению работоспособности, вызываемому развивающимся утомлением работника.

5.1.1 Нарушение параметров микроклимата

Проанализируем микроклимат в здании, где расположено рабочее место. Микроклимат производственных зданий описывает следующие параметры: скорость движения воздуха, относительная влажность, температура. Эти факторы определяя его самочувствие, влияют на организм человека.

Допустимые и оптимальные значения параметров микроклимата приведены в таблице 1.

Параметры микроклимата в здании, где распаложено рабочее место, приточно-вытяжной вентиляцией и регулируются системой центрального отопления [1].

Таблица 1

Допустимые и оптимальные нормы микроклимата.

Период года	Температура, °С					Относительная влажность, %		Скорость движения воздуха, м/с	
	Оптимальная	Допустимая на рабочих местах				Оптимальная	Допустимая	Оптимальная, не более	Допустимая, не более
		Верхняя		Нижняя					
		Пост.	Не пост.	Пост.	Не пост.				
Холодный	21 – 23	24	25	22	19	30 - 50	65	0,2	0,2
Теплый	22 – 24	27	29	23	21	30 - 50	60	0,2	0,2

Меры по улучшению воздушной среды в производственном помещении включают: правильную организацию вентиляции и кондиционирования воздуха, отопление помещений. Вентиляция может осуществляться естественным и механическим способом. В здании должны подаваться следующие объёмы наружного воздуха: при объёме помещения более 45 м³ на человека– не менее 35 м³ в час на человека; при объёме помещения до 25 м³ на человека и отсутствии выделения вредных веществ разрешается естественная вентиляция.

Присутствует в аудитории 407 принудительная вентиляция. Её наличие обусловлено требованиями по объёму воздуха в помещении приходящегося на одного работника – более 40 м³.

В зимнее время в помещении предусмотрена система водяного отопления со встроенными нагревательными элементами и терморегуляторами.

5.1.2 Плохой освещенность рабочей зоны

Важность освещения очень велико в производственной сфере и особенно в процессе жизнедеятельности. При длительной работе плохой освещенность рабочей зоны приводит ухудшению зрения и к ослаблению зрительной активности работающего. Правильно сделанная система освещения имеет большое значение уменьшая потенциальную опасность многих производственных факторов в снижении производственного травматизма, повышает общую работоспособность организма и для работы органам зрения создает нормальные условия.

Рабочее место разработчика согласно санитарно-гигиеническим требованиям должно освещаться искусственным и освещением естественным.

Рабочее место или рабочая зона освещается в такой степени, чтобы неплохо видеть процесс работы, исключалось прямое попадание лучей источника света в глаза и чтобы не напрягая зрения. Кроме того, уровень освещения определяется степенью точности зрительных работ. Наименьший размер объекта различения составляет 0.5 - 1 мм. Кроме того в помещении отсутствует естественное освещение. По нормам освещенности СанНиП 23-05-95 (СанНиП 23-05-2010) и отраслевым нормам, работа разработчика относится к четвертому разряду зрительной работы средней точности. Для этого разряда рекомендуется освещенность 400 лк.

Искусственное освещение осуществляется с использованием газоразрядных люминесцентных ламп низкого давления типа ЛБ-40, в количестве 12 светильников в каждом по 3 лампы.

Правильно спроектированное и рационально выполненное освещение производственных помещений оказывает положительное воздействие на работающих, способствует повышению эффективности и безопасности труда, снижает утомление и травматизм, сохраняет высокую работоспособность.

5.1.3 Монотонный режим работы

При работе с ПЭВМ основным фактором, влияющим на нервную систему программиста или пользователя, является огромное количество информации, которое он должен воспринимать. Это является сложной задачей, которая очень сильно влияет на сознание и психофизическое состояние из-за монотонности работы. Поэтому меры, позволяющие снизить воздействие этого вредного производственного фактора, которые регулируются СанПиН 2.2.2/2.4.1340-03, являются важными в работе оператора ПЭВМ. Они позволяют увеличить производительность труда и предотвратить появление профессиональных болезней.

Организация работы с ПЭВМ осуществляется в зависимости от вида и категории трудовой деятельности. Виды трудовой деятельности разделяются на 3 группы: группа А – работа по считыванию информации с экрана с предварительным запросом; группа Б – работа по вводу информации; группа В – творческая работа в режиме диалога с ПЭВМ. Работа программиста-разработчика рассматриваемой в данной работе системы относится к группам А и Б, в то время, как деятельность врача-специалиста, который будет использовать систему в профессиональной деятельности, относится к группе В. Категории трудовой деятельности различаются по степени тяжести выполняемых работ. Для снижения воздействия рассматриваемого вредного фактора предусмотрены регламентированные перерывы для каждой группы работ – таблица 2.

Таблица 2 – Суммарное время регламентированных перерывов в зависимости от продолжительности работы, вида категории трудовой деятельности с ПЭВМ [2]

Таблица 2

Категория работы с ПЭВМ	Уровень нагрузки за рабочую смену при видах работ с ПЭВМ			Суммарное регламентированных перерывов, мин. время	
	группа А, количество знаков	группа Б, количество знаков	группа В, ч	при 8-часовой смене	при 12- часовой смене
I	до 20 000	до 15 000	до 2	50	80
II	до 40 000	до 30 000	до 4	70	110
III	до 60 000	до 40 000	до 6	90	140

5.1.4 Опасность поражения электрическим током

Поражение электрическим током является опасным производственным фактором и, поскольку оператор ПЭВМ имеет дело с электрооборудованием, то вопросам электробезопасности на его рабочем месте должно уделяться много внимания. Нормы электробезопасности на рабочем месте регламентируются СанПиН 2.2.2/2.4.1340-03, вопросы требований к защите от поражения электрическим током освещены в ГОСТ Р 12.1.019-2009 ССБТ.

Электробезопасность – система организационных и технических мероприятий и средств, обеспечивающих защиту людей от вредного и опасного воздействия электрического тока, электрической дуги, электромагнитного поля и статического электричества.

Опасность поражения электрическим током усугубляется тем, что человек не в состоянии без специальных приборов обнаружить напряжение дистанционно.

Помещение, где расположено рабочее место оператора ПЭВМ, относится к помещениям без повышенной опасности ввиду отсутствия следующих факторов: сырость, токопроводящая пыль, токопроводящие полы, высокая

температура, возможность одновременного прикосновения человека к имеющим соединение с землей металлоконструкциям зданий, технологическим аппаратам, механизмам и металлическим корпусам электрооборудования.

Для оператора ПЭВМ при работе с электрическим оборудованием обязательны следующие меры предосторожности:

- Перед началом работы нужно убедиться, что выключатели и розетка закреплены и не имеют оголённых токоведущих частей;
- При обнаружении неисправности оборудования и приборов необходимо, не делая никаких самостоятельных исправлений, сообщить человеку, ответственному за оборудование [2, 3].

5.1.5 Опасность возникновения пожара

Возникновение пожара является опасным производственным фактором, т.к. пожар на предприятии наносит большой материальный ущерб, а также часто сопровождается травмами и несчастными случаями. Регулирование пожаробезопасности производится СНИП 21-01-97.

В помещениях с ПЭВМ повышен риск возникновения пожара из-за присутствия множества факторов: наличие большого количества электронных схем, устройств электропитания, устройств кондиционирования воздуха; возможные неисправности электрооборудования, освещения, или неправильная их эксплуатация может послужить причиной пожара.

Возможные виды источников воспламенения:

- Искра при разряде статического электричества;
- Искры от электрооборудования;
- Несоблюдение правил пожарной безопасности;
- Открытое пламя [4].

5.2 Экологическая безопасность

В данном разделе рассматривается воздействие на окружающую среду деятельности по разработке проекта, а также самого продукта в результате его реализации на производстве.

Разработка программного обеспечения и работа за ПЭВМ не являются экологически опасными работами, потому объект, на котором производилась разработка продукта, а также объекты, на которых будет производиться его использование операторами ПЭВМ относятся к предприятиям пятого класса, размер селитебной зоны для которых равен 50 м [5].

Непосредственно программный продукт, разработанный в ходе выполнения магистерской диссертации, не наносит вреда окружающей среде ни на стадиях его разработки, ни на стадиях эксплуатации. Однако, средства, необходимые для его разработки и эксплуатации могут наносить вред окружающей среде.

Современные ПЭВМ производят практически без использования вредных веществ, опасных для человека и окружающей среды. Исключением являются аккумуляторные батареи компьютеров и мобильных устройств. В аккумуляторах содержатся тяжелые металлы, кислоты и щелочи, которые могут наносить ущерб окружающей среде, попадая в гидросферу и литосферу, если они были неправильно утилизированы. Для утилизации аккумуляторов необходимо обращаться в специальные организации, специализировано занимающиеся приемом, утилизацией и переработкой аккумуляторных батарей [6].

Люминесцентные лампы, применяющиеся для искусственного освещения рабочих мест, также требуют особой утилизации, т.к. в них присутствует от 10 до 70 мг ртути, которая относится к чрезвычайно-опасным химическим веществам и может стать причиной отравления живых существ, а также загрязнения атмосферы, гидросферы и литосферы. Сроки службы таких ламп составляют около 5-ти лет, после чего их необходимо сдавать на

переработку в специальных пунктах приема. Юридические лица обязаны сдавать лампы на переработку и вести паспорт для данного вида отходов [6-7].

Возле каждого города с населением 1 млн. человек ежегодно образуется до двух миллионов кубических метров твердых бытовых отходов (ТБО). То есть, каждые 15-20 лет вокруг каждого города миллионника образуется свалка площадью 30-40 гектаров.

Положение усугубляется тем, что вместе с пищевыми отходами, бумагой, стеклянной, полимерной, и металлической тарой, выбрасываются разбитые ртутьсодержащие люминесцентные лампы и термометры, лекарства с просроченным сроком годности, тара с остатками лаков, красок, ядохимикатов и т.п. Все это под видом обычных малоопасных бытовых отходов вывозится на свалки, которые обычно устраивают в оврагах, выработанных карьерах, и прочих непригодных для этого местах, что совершенно недопустимо с эколого-гигиенической точки зрения. Разработанным алгоритмом можем сделать панорамную визуализацию для того чтобы информировать окружающих. Также во многих местах изменяются природные комплексы, связанное с массовой гибелью живых организмов. С этой программой можем сфотографировать таких места и сделать панорамную визуализацию для того чтобы определить принципы и способы сохранения биоразнообразия на разных иерархических уровнях.

5.3 Безопасность в чрезвычайных ситуациях

В рабочей среде оператора ПЭВМ возможно возникновение следующих чрезвычайных ситуаций техногенного характера:

- Пожары и взрывы в зданиях и на коммуникациях;
- Социальные ЧС(терроризма);
- Внезапное обрушение зданий.

Экологические чрезвычайные ситуации могут быть вызваны изменениями состояния, литосферы, гидросферы, атмосферы и биосферы в результате деятельности человека [8].

Наиболее характерной для объекта, где размещаются рабочие помещения, оборудованные ПЭВМ, чрезвычайной ситуацией является пожар.

Помещение для работы операторов ПЭВМ по системе классификации категорий помещений по взрывопожарной и пожарной опасности относится к категории Д (из 5-ти категорий А, Б, В1-В4, Г, Д), т.к. относится к помещениям с негорючими веществами и материалами в холодном состоянии [9].

Каждый сотрудник организации должен быть ознакомлен с инструкцией по пожарной безопасности, пройти инструктаж по технике безопасности и строго соблюдать его.

Запрещается использовать электроприборы в условиях, не соответствующих требованиям инструкций изготовителей, или имеющие неисправности, которые в соответствии с инструкцией по эксплуатации могут привести к пожару, а также эксплуатировать электропровода и кабели с поврежденной или потерявшей защитные свойства изоляцией. Электроустановки и бытовые электроприборы в помещениях по окончании рабочего времени должны быть обесточены (вилки должны быть вынуты из розеток). Под напряжением должны оставаться дежурное освещение и пожарная сигнализация. Недопустимо хранение легковоспламеняющихся, горючих и взрывчатых веществ, использование открытого огня в помещениях офиса.

Перед уходом из служебного помещения работник обязан провести его осмотр, закрыть окна, и убедиться в том, что в помещении отсутствуют источники возможного возгорания, все электроприборы отключены и выключено освещение. С периодичностью не реже одного раза в три года необходимо проводить замеры сопротивления изоляции токоведущих частей силового и осветительного оборудования.

Повышение устойчивости достигается за счет проведения соответствующих организационно-технических мероприятий, подготовки персонала к работе в ЧС [8].

Работник при обнаружении пожара или признаков горения (задымление, запах гари, повышение температуры и т.п.) должен:

- Немедленно прекратить работу и вызвать пожарную охрану по телефону «01», сообщив при этом адрес, место возникновения пожара и свою фамилию;
- Принять по возможности меры по эвакуации людей и материальных ценностей;
- Отключить от сети закрепленное за ним электрооборудование;
- Приступить к тушению пожара имеющимися средствами пожаротушения;
- Сообщить непосредственному или вышестоящему начальнику и оповестить окружающих сотрудников;
- При общем сигнале опасности покинуть здание согласно «Плану эвакуации людей при пожаре и других ЧС».

Для тушения пожара применять ручные углекислотные огнетушители (типа ОУ-2, ОУ-5), находящиеся в помещениях офиса, и пожарный кран внутреннего противопожарного водопровода. Они предназначены для тушения начальных возгораний различных веществ и материалов, за исключением веществ, горение которых происходит без доступа воздуха. Огнетушители должны постоянно содержаться в исправном состоянии и быть готовыми к действию. Категорически запрещается тушить возгорания в помещениях офиса при помощи химических пенных огнетушителей (типа ОХП-10) [10].

Разработанный алгоритм может помочь для обеспечения безопасности в чрезвычайных случаях как часть программного продукта. Программы такого рода служат для получения панорамную изображение и применить для проведении следственных действий, оперативных мероприятий и криминалистических экспертиз с целью расследования преступлений и представления в суд наглядного доказательственного материала.

5.4 Правовые и организационные вопросы обеспечения безопасности

Регулирование отношений между работником и работодателем, касающихся оплаты труда, трудового распорядка, особенности регулирования труда женщин, детей, людей с ограниченными способностями и проч., осуществляется законодательством РФ, а именно трудовым кодексом РФ.

Продолжительность рабочего дня не должна быть меньше указанного времени в договоре, но не больше 40 часов в неделю. Для работников до 16 лет – не более 24 часов в неделю, от 16 до 18 лет и инвалидов I и II группы – не более 35 часов.

Возможно установление неполного рабочего дня для беременной женщины; одного из родителей (опекуна, попечителя), имеющего ребенка в возрасте до четырнадцати лет (ребенка-инвалида в возрасте до восемнадцати лет). Оплата труда при этом производится пропорционально отработанному времени, без ограничений оплачиваемого отпуска, исчисления трудового стажа и других прав.

При работе в ночное время продолжительность рабочей смены сокращается на один час. К работе в ночную смену не допускаются беременные женщины; работники, не достигшие возраста 18 лет; женщины, имеющие детей в возрасте до трех лет, инвалиды, работники, имеющие детей-инвалидов, а также работники, осуществляющие уход за больными членами их семей в соответствии с медицинским заключением, матери и отцы-одиночки детей до пяти лет.

Организация обязана предоставлять ежегодный отпуск продолжительностью 28 календарных дней. Дополнительные отпуска предоставляются работникам, занятым на работах с вредными или опасными условиями труда, работникам имеющими особый характер работы, работникам с ненормированным рабочим днем и работающим в условиях Крайнего Севера и приравненных к нему местностях.

В течение рабочего дня работнику должен быть предоставлен перерыв для отдыха и питания продолжительностью не более двух часов и не менее 30 минут, который в рабочее время не включается. Всем работникам предоставляются выходные дни, работа в выходные дни осуществляется только с письменного согласия работника.

Организация-работодатель выплачивает заработную плату работникам. Возможно удержание заработной платы только в случаях установленных ТК РФ ст. 137. В случае задержки заработной платы более чем на 15 дней, работник имеет право приостановить работу, письменно уведомив работодателя.

Законодательством РФ запрещена дискриминация по любым признакам и принудительный труд [11].

Влияние реализации программы на работу компаний:

- Позволяет экономить время и усилия, затрачиваемые на распознавания данных вручную;
- Экономия денежных средств;
- Требуется оснастить рабочее место и обучить рабочий персонал.

ЗАКЛЮЧЕНИЕ

В результате работы над диссертацией была подготовлена база данных изображений, состоящая из 25 различных сцен, каждая из которых содержит по 8-9 изображений. Общее количество изображений в базе равно 200.

Был предложен и разработан алгоритм создания панорамных изображений, состоящий из следующих этапов:

1. Были исследованы существующие методы поиска ключевых точек: *SIFT*, *SURF* и *ORB*.

2. Для сопоставления особых точек было проведено сравнение четырех метрик: Расстояния Хэмминга, Евклидова расстояния, Манхэттенского расстояния и Расстояния Чебышева. Выполняется поиск ближайших соседей k для набора точек запроса.

3. Нахождение матрицы преобразования осуществляется на основе алгоритма RANSAC. В результате применения RANSAC строится наилучшая матрица гомографии.

4. Сшивание изображений осуществляется на основе найденной матрицы преобразования. Были рассчитаны новые координаты ключевых точек второго изображения для его объединения с первым.

5. Построение трехмерной сцены полученной панорамы происходит в графической среде UNITY на основе примитива куб.

6. В результате работы предложен алгоритм создания панорамных изображений. На основе предложенного алгоритма реализовано программное обеспечение.

Новые возможности по построению панорамных изображений позволяют эффективно использовать эти изображения для туризма, панорамных картах, космосе и т.д.

ОПУБЛИКОВАННЫЕ РАБОТЫ

1. Разработка алгоритма построения панорам [Электронный ресурс] / М.О.Толеген, науч. рук. Ю. А. Болотова// Молодежь и современные информационные технологии: сборник трудов XIII Международной научно-практической конференции студентов, аспирантов и молодых ученых, г. Томск, 7-11 октября 2016 г.в 2 т. / Национальный исследовательский Томский политехнический университет (ТПУ), Институт кибернетики (ИК) ; ред. кол. В. С. Аврамчук [и др.]. — 2017. — Т. 2. — [С. 195-96]. — Заглавие с титульного экрана. — Свободный доступ из сети Интернет. — Adobe Reader.

Режим доступа: <http://earchive.tpu.ru/handle/11683/17096>

2. Исследование методов поиска и сопоставления особых точек для решения задачи построения панорамных изображений [Электронный ресурс] М.О.Толеген, науч. рук. Ю. А. Болотова // XXII Международной научно-технической конференции студентов, аспирантов и молодых ученых «Научная сессия ТУСУР 2017», посвященной 55-летию ТУСУРа. г. Томск, 10–12 мая 2017 г.: в 8 частях. – Томск: В-Спектр, 2017 – Ч. 5. – 244 с. / Томский государственный университет систем управления и радиоэлектроники (ТУСУР).

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Дж. Стокман, Л. Шапиро. Компьютерное зрение. Пер. с англ. — М.: БИНОМ. Лаборатория знаний, 2006.
2. Shi and C. Tomasi. Good Features to Track. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 593-600, June 1994.
3. Bishop C. M. Pattern Recognition and Machine Learning. Springer, 2006.
4. Canny. A Computational Approach to Edge Detection. IEEE Trans. on Pattern Analysis and Machine Intelligence, 8(6), pp. 679-698 (1986).
5. Friedman J. H. Stochastic Gradient Boosting. Technical report. Dept. of Statistics, Stanford University, 1999.
6. Breiman L., Friedman J.H., Olshen R.A., Stone C.J. Classi Classification and Regression Trees Wadsworth & Brooks, 1984.
7. Juan L., Gwun O. A Comparison of SIFT, PCA-SIFT and SURF // International Journal of Image Processing (IJIP). 2009.
8. Tran P. Pyramid Matching Using SURF and SIFT Descriptors for SVM Classification. University of Kansas.—2010. Режим доступа:http://www.ittc.ku.edu/potetz/EECS_841_Fall09/Lectures/Final%20Projects/Paper_04.pdf (дата обращения 12.04.2017);
9. Valgren C., Lilienthal A.J. SIFT, SURF & Seasons: Appearance-based Long-term Localization in Outdoor Environments // Robotics and Autonomous Systems. 2010. Vol. 58. No. 2. PP. 149–156. DOI: <http://dx.doi.org/10.1016/j.robot.2009>.
10. Meng Y., Tiddeman B. Implementing the Scale Invariant Feature Transform (SIFT) Method. Citeseer. – 2008. Режим доступа: http://www.cs.st-andrews.ac.uk/~yumeng/yumeng-SIFTreport-5.18_bpt.pdf (дата обращения 10.03.2017);
11. Гонсалес Р., Вудс Р. Цифровая обработка изображений. Пер. с англ. под ред. П.А. Чочиа. М.: Техносфера, 2005.
12. <https://habrahabr.ru/post/152679/>

13. Rosten E., Drummond T. Machine learning for high-speed corner detection // Proc. European Conference on Computer Vision. – 2006.
14. Taylor S., Rosten E., Drummond T. Robust feature matching in 2.3 μ s // Proc. IEEE CVPR Workshop on Feature Detectors and Descriptors: The State Of The Art and Beyond. – 2009.
15. Rublee E., Rabaud V., Konolige K., Bradski G. ORB: an efficient alternative to SIFT or SURF. IEEE International Conference on Computer Vision, 2011.
16. Гомография, <http://en.wikipedia.org/wiki/Homography>.
17. Pollefeys M. Visual 3D Modelling from Images, 2002. <http://www.cs.unc.edu/~marc/tutorial/node3.html>
18. Lindeberg T. Feature Detection with Automatic Scale Selection // IJCV(30). – 1998.
19. V. Rodehorst, A. Koschan. Comparison and evaluation of feature point detectors, 2006.
20. Конушин А.С. Слежение за точечными особенностями сцены (Point feature tracking) // Компьютерная графика и мультимедиа. Электрон. журн. 2003. №1(5). Режим доступа <http://cgm.computergraphics.ru/content/view/54> (дата обращения 16.05.2017);
21. Tinne Tuytelaars, Krystian Mikolajczyk. Local Invariant Feature Detectors: A Survey, 2008.
22. Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-Up Robust Features (SURF).
23. Сойфер В.А. Методы компьютерной обработки изображений. – 2-е изд., испр. – М.: ФИЗМАТЛИТ, 2003.
24. Bradsky G., Kaehler A. Learning OpenCV — O'Reilly, 2008.
25. ГОСТ Р 17.0.0.06-2000 «Охрана природы. Экологический паспорт природопользователя. Основные положения. Типовые формы».
26. СанПиН 2.2.2/2.4.1340-03. Санитарно-эпидемиологические правила и нормы. Гигиенические требования к персональным электронно-

вычислительным машинам и организации работы // Библиотека гостей и нормативов. 2016.

URL:http://www.ohranatruda.ru/ot_biblio/normativ/data_normativ/39/39082/#i72870
(дата обращения: 08.05.2017).

27. ГОСТ Р 12.1.019-2009 ССБТ. Электробезопасность. Общие требования и номенклатура видов защиты // Электронный фонд правовой и нормативно-технической документации. 2010. URL:
<http://docs.cntd.ru/document/gost-r-12-1-019-2009-ssbt> (дата обращения: 08.05.2017).

28. СНиП 21-01-97. Пожарная безопасность зданий и сооружений // Библиотека гостей и нормативов. 2016. URL:
http://www.ohranatruda.ru/ot_biblio/normativ/data_normativ/2/2107/ (дата обращения: 08.05.2017).

29. СанПиН 2.2.1/2.1.1.1200-03. Санитарно-эпидемиологические правила и нормативы. Санитарно-защитные зоны и санитарная классификация предприятий, сооружений и других объектов // Библиотека гостей и нормативов. 2016. URL: http://ohranatruda.ru/ot_biblio/normativ/data_normativ/11/11774/ (дата обращения: 08.05.2017).

30. СанПиН 2.1.7.1322-03. Санитарно-эпидемиологические правила и нормативы. Гигиенические требования к размещению и обезвреживанию отходов производства и потребления. 2.1.7. Почва, очистка населённых мест, бытовые и промышленные отходы, санитарная охрана почвы // Библиотека гостей и нормативов. 2016. URL:
http://ohranatruda.ru/ot_biblio/normativ/data_normativ/11/11774/ (дата обращения: 08.05.2017).

31. Энергосбережение в компьютерном мире // НВП. 2008. URL:
http://www.hwp.ru/articles/Energoberezhenie_v_kompyuternom_mire_CHast_1____osnovnie_tendentsii/?SHOWALL_1=1 (дата обращения: 08.05.2017).

32. Назаренко О. Б. Безопасность жизнедеятельности: учебное пособие / О. Б. Назаренко, Ю. А. Амелькович; Томский политехнический университет. –

3-е изд., перераб. и доп. – Томск: Изд-во Томского политехнического университета, 2013. – 178 с.

33. НПБ 105-03 Определение категорий помещений, зданий и наружных установок по взрывопожарной и пожарной опасности // Электронный фонд правовой и нормативно-технической документации. 2016. URL: <http://docs.cntd.ru/document/1200032102> (дата обращения: 24.04.2017).

34. ППБ 01–03. Правила пожарной безопасности в Российской Федерации. – М.: Министерство Российской Федерации по делам гражданской обороны, чрезвычайным ситуациям и ликвидации последствий стихийных бедствий, 2003.

35. Трудовой кодекс Российской Федерации" от 30.12.2001 N 197-ФЗ (ред. от 24.04.2017) // Консультант Плюс. 2015. URL: http://www.consultant.ru/document/cons_doc_law_34683/?utm_campaign=law_doc&utm_source=google.adwords&utm_medium=cpc&utm_content=Labor%20Code&gclid=CjwKEAjwgPe4BRCB66GG8PO69QkSJAC4EhHhU-5yAFZCJfmzkTLNGnrpgHHAyFPhhPzRo-sZGWmqnBoCPynw_wcB (дата обращения: 25.04.2017).

ПРИЛОЖЕНИЕ А

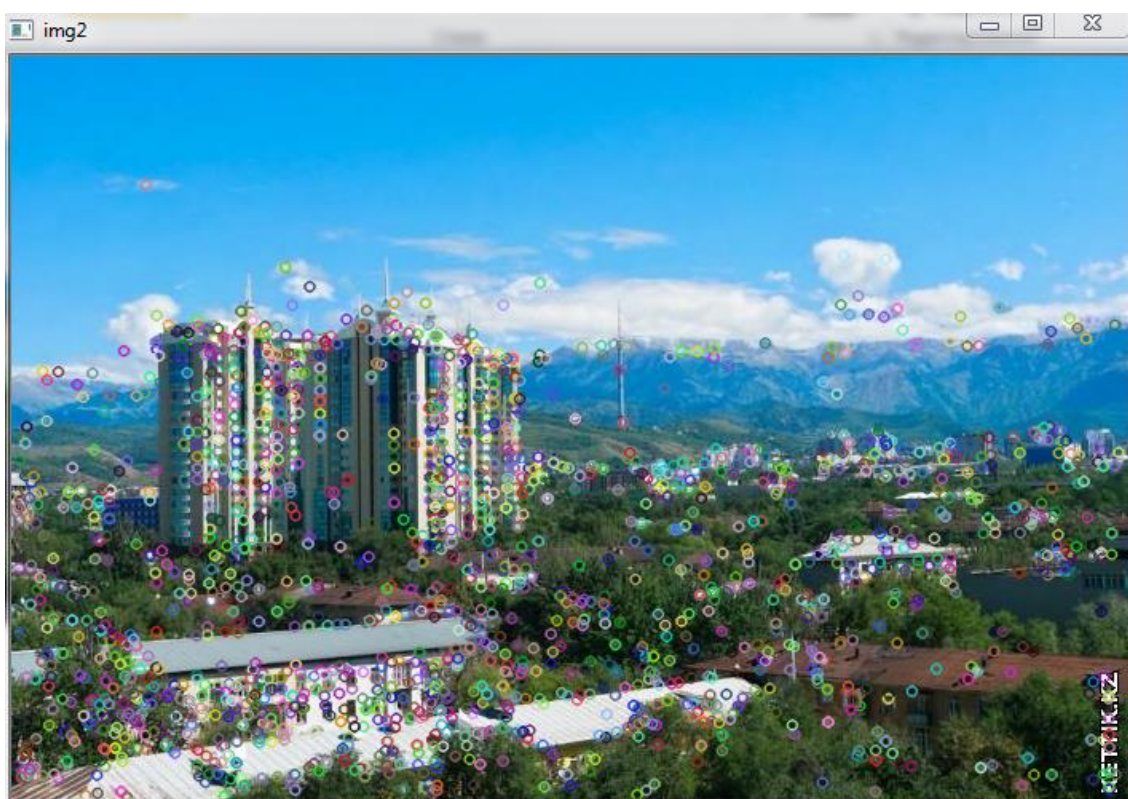
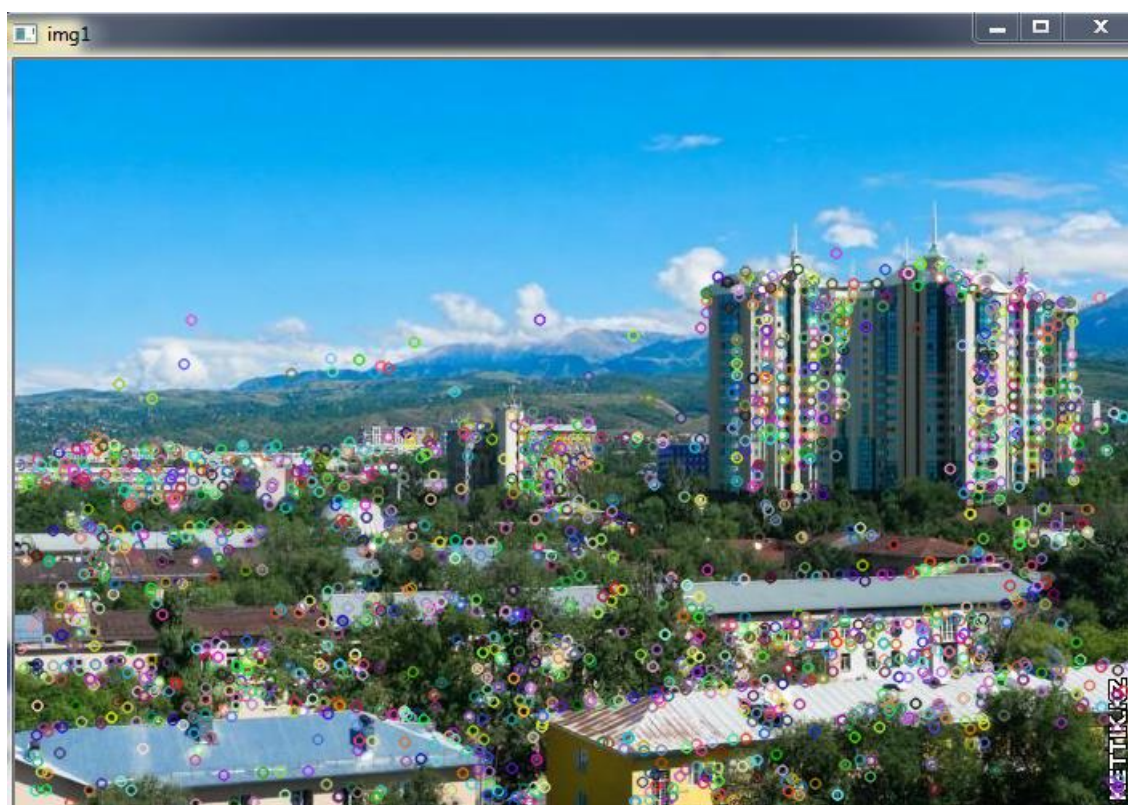


Рисунок 15 – Найденные особые точки алгоритма *SIFT*

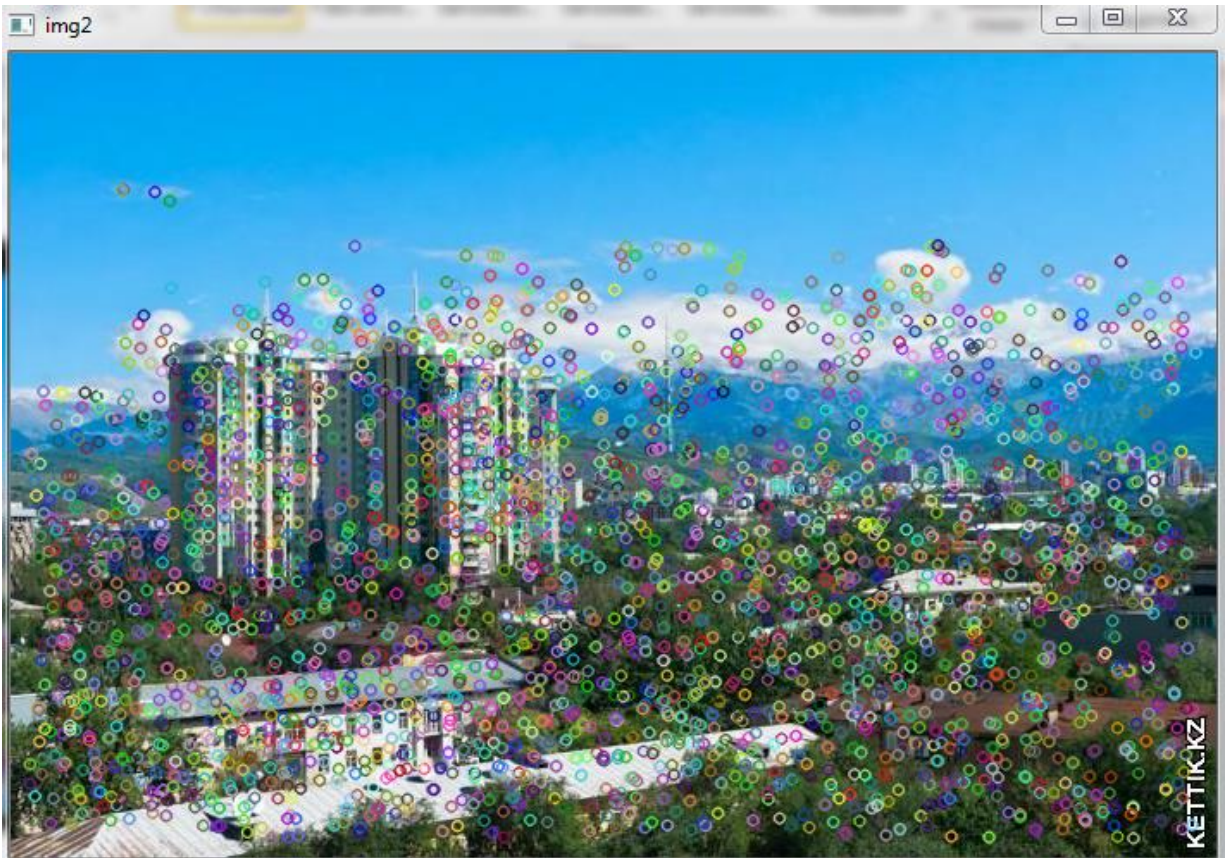
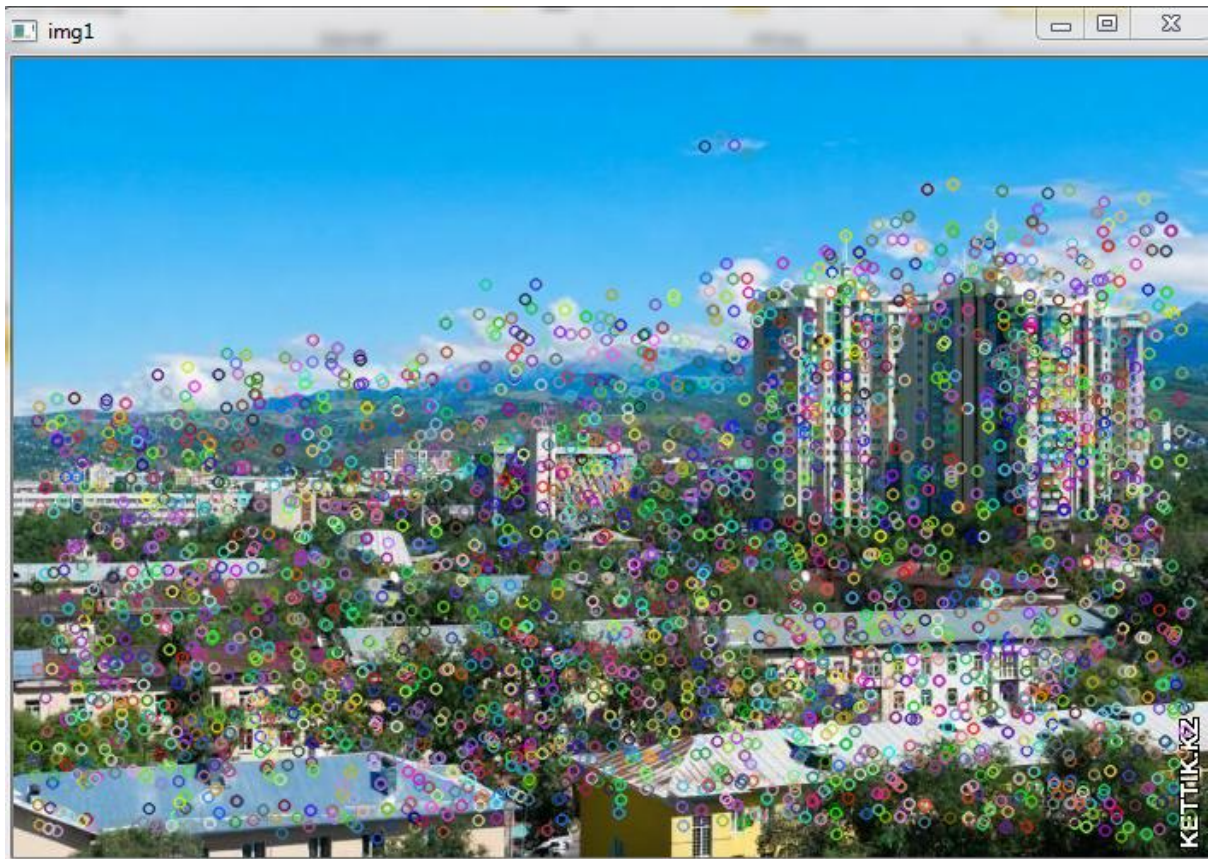


Рисунок 16 – Найденные особые точки алгоритма *SURF*

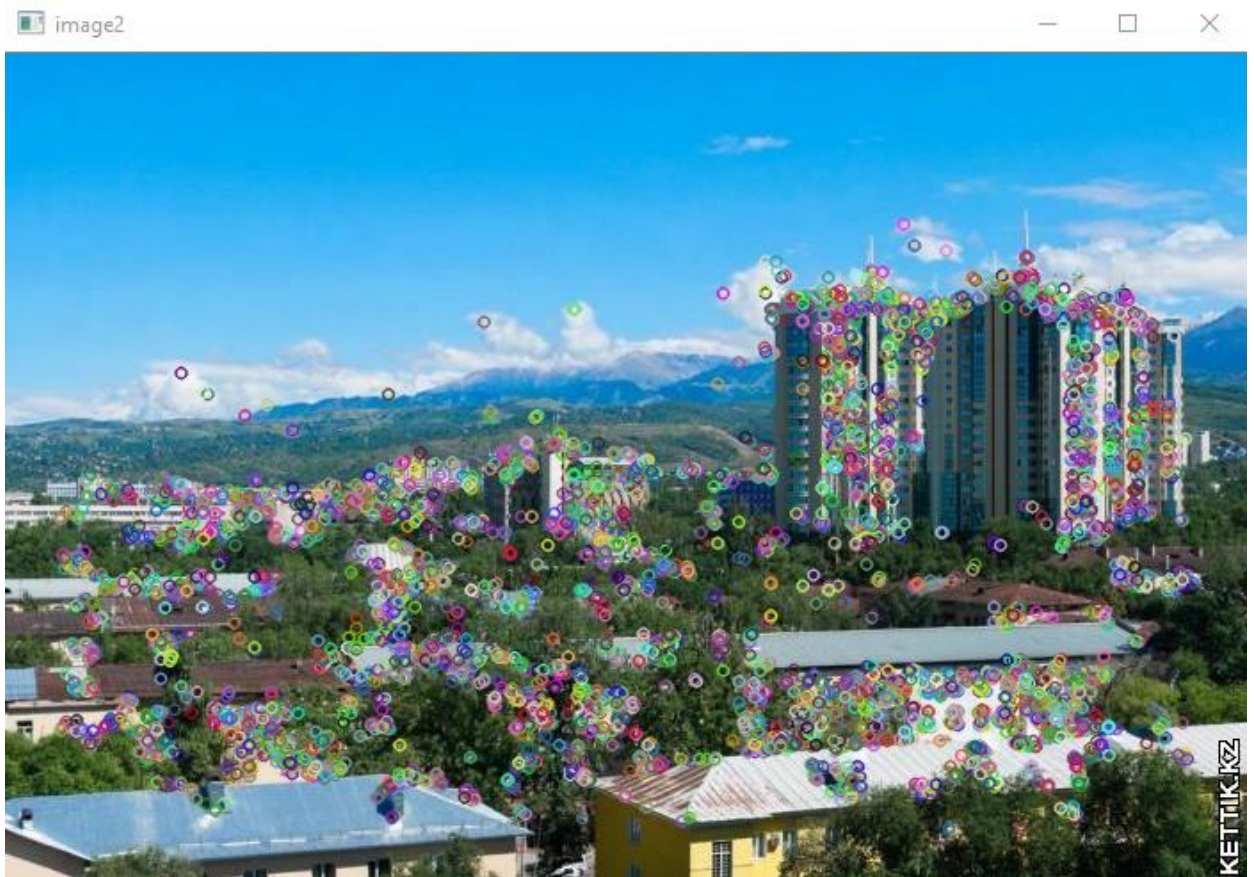
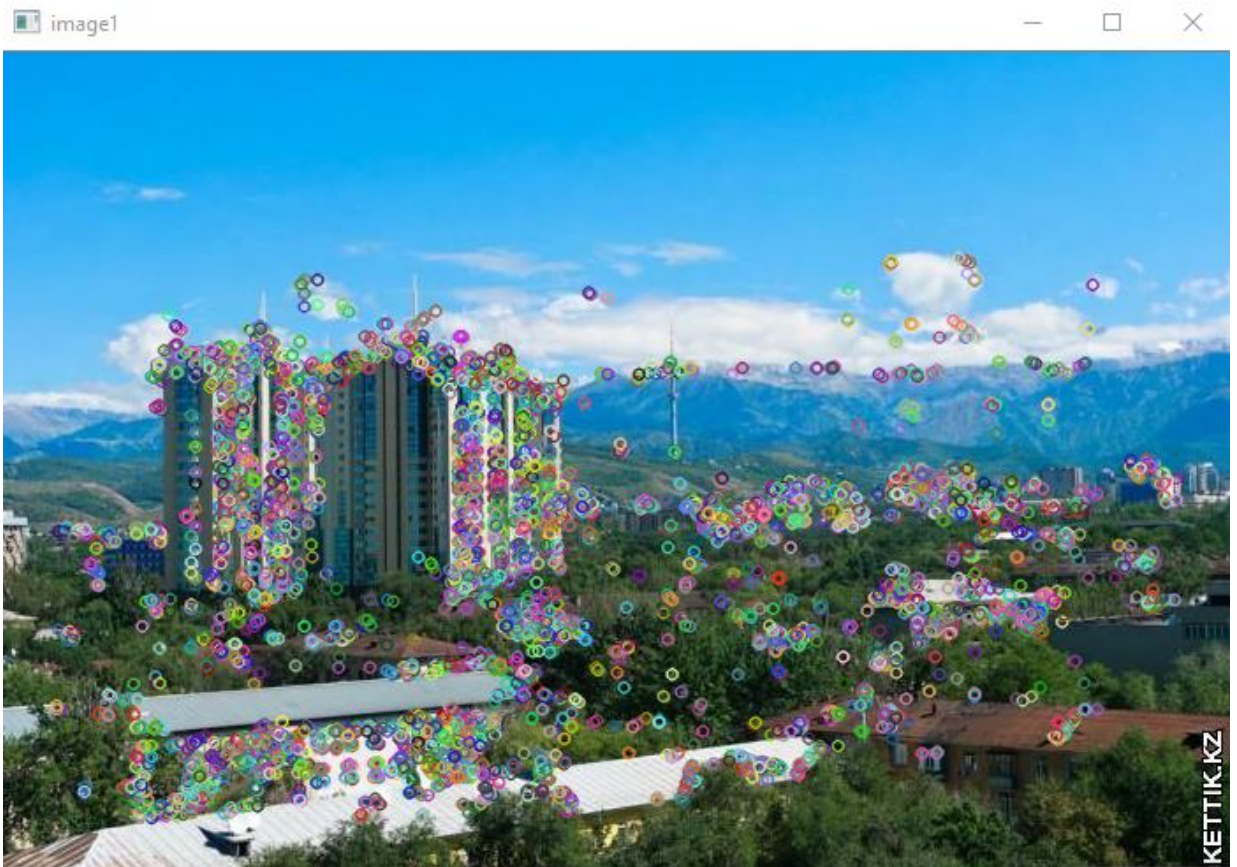


Рисунок 17 – Найденные особые точки алгоритма *ORB*

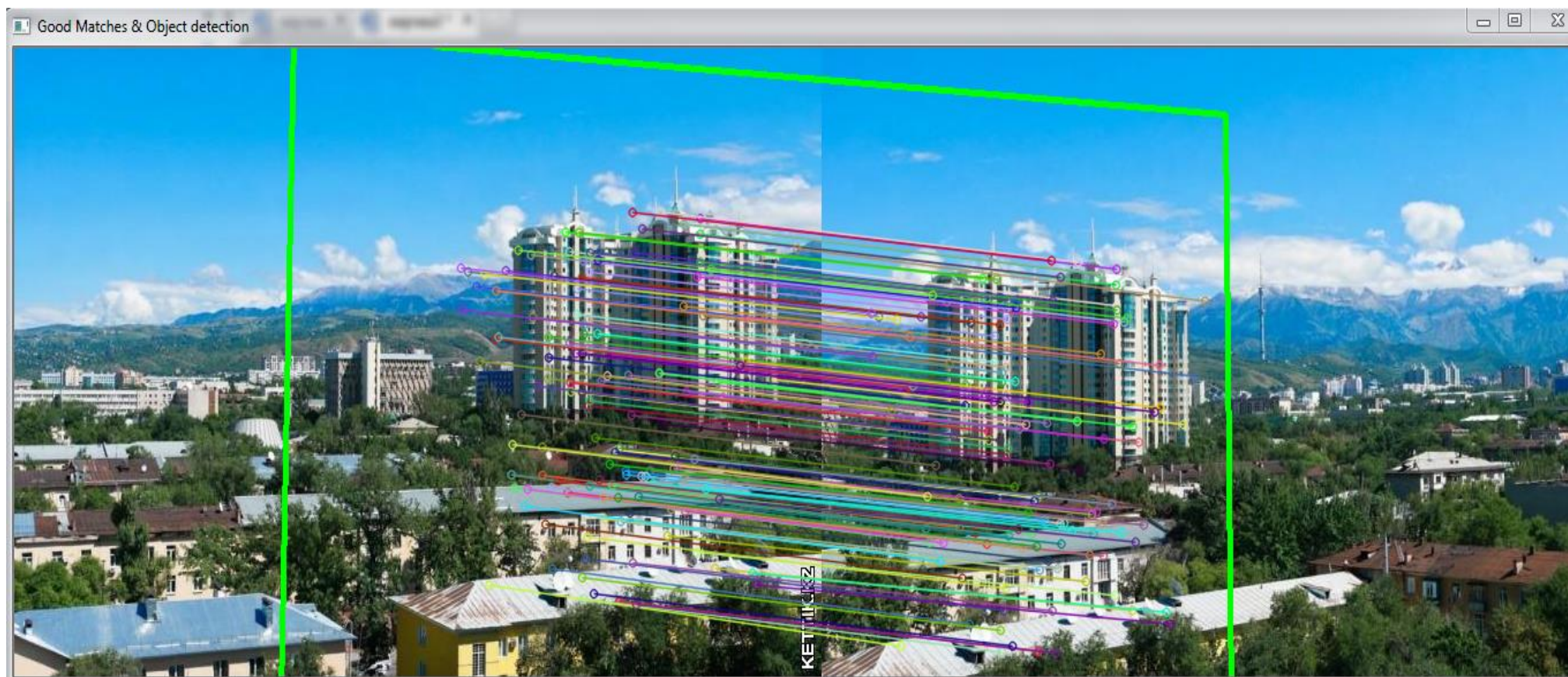


Рисунок 18 – Сопоставление изображений одинаковой ориентации на основе ключевых точек, полученных с помощью дескриптора *SURF*

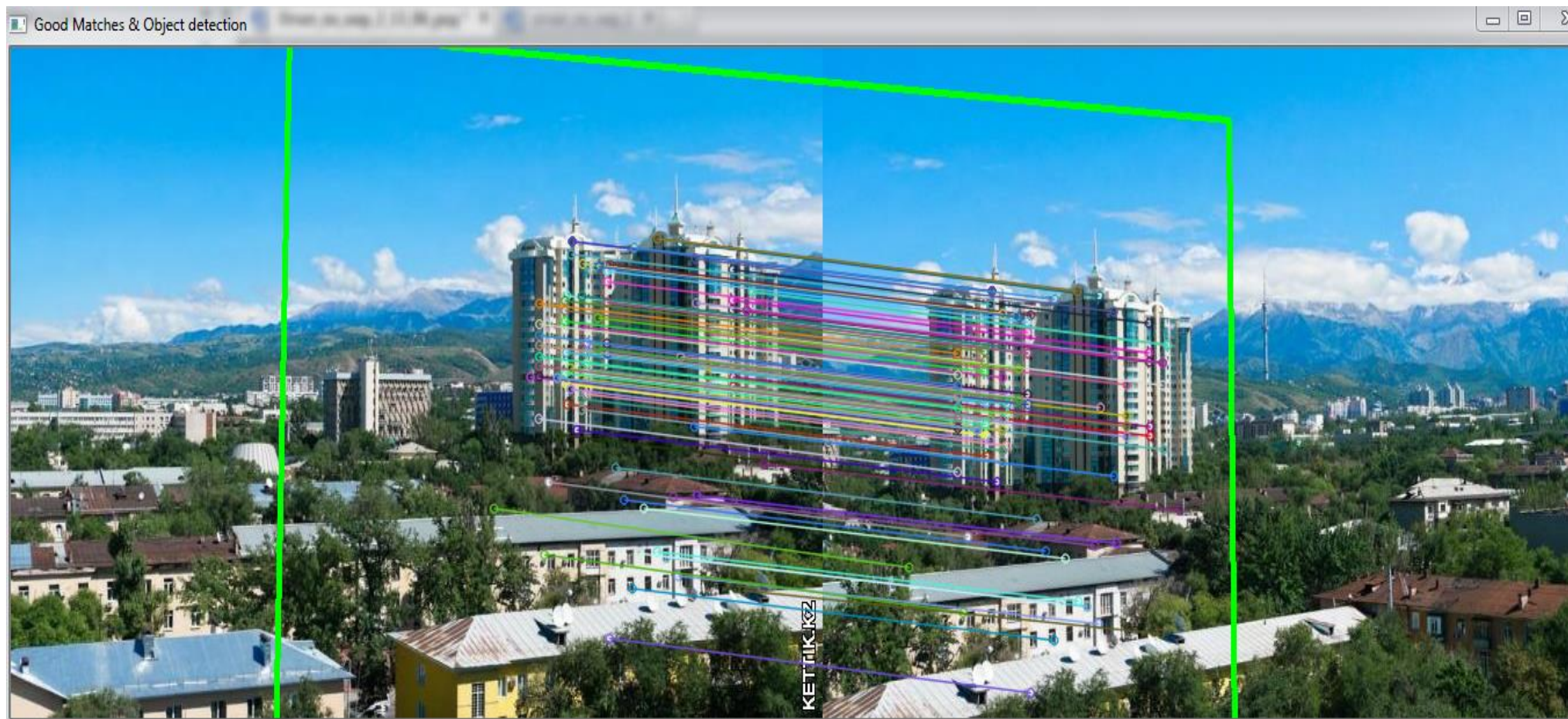


Рисунок 19 – Сопоставление изображений одинаковой ориентации на основе ключевых точек, полученных с помощью дескриптора *SIFT*

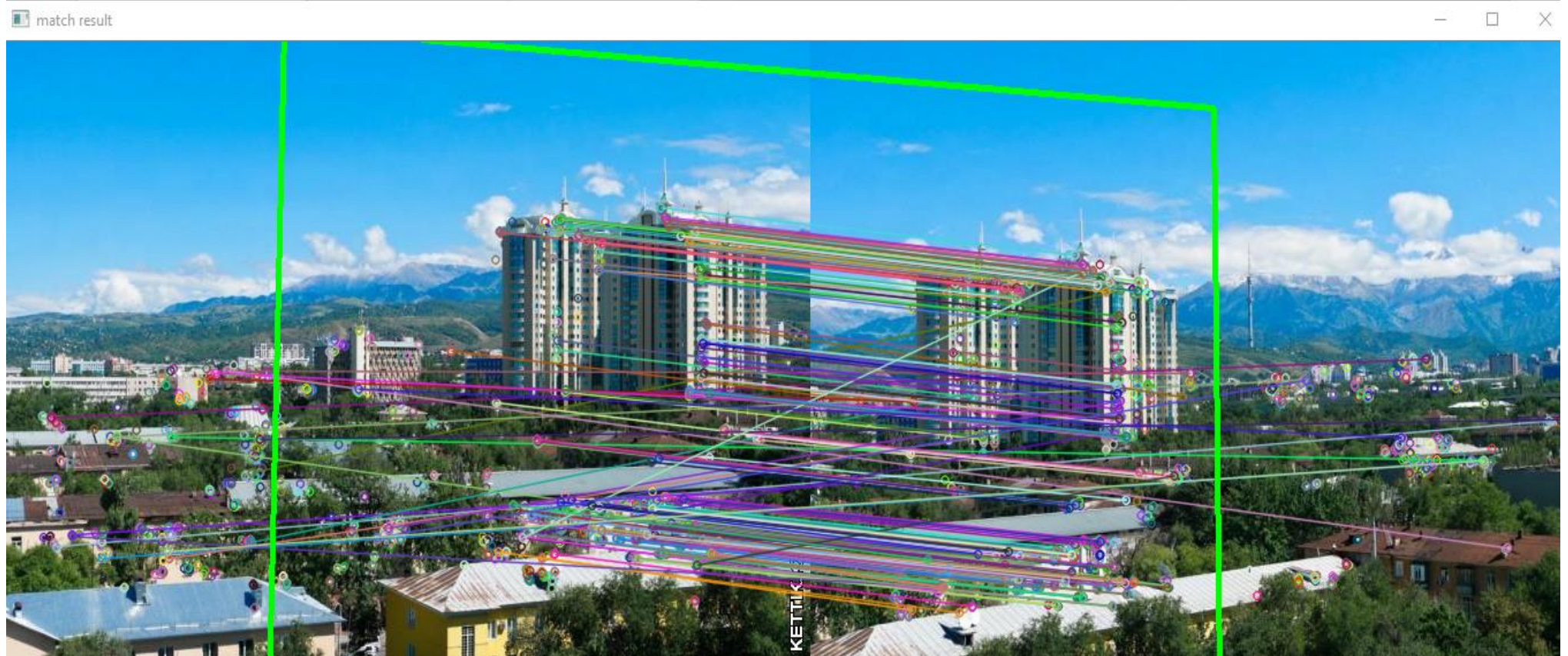


Рисунок 20 – Сопоставление изображений одинаковой ориентации на основе ключевых точек, полученных с помощью дескриптора *ORB*

Result



Рисунок 21 – Совмещение изображений полученных с помощью дескриптора *SIFT*.



Рисунок 22 – Панорамное изображения



Рисунок 23 – Панорамное изображения

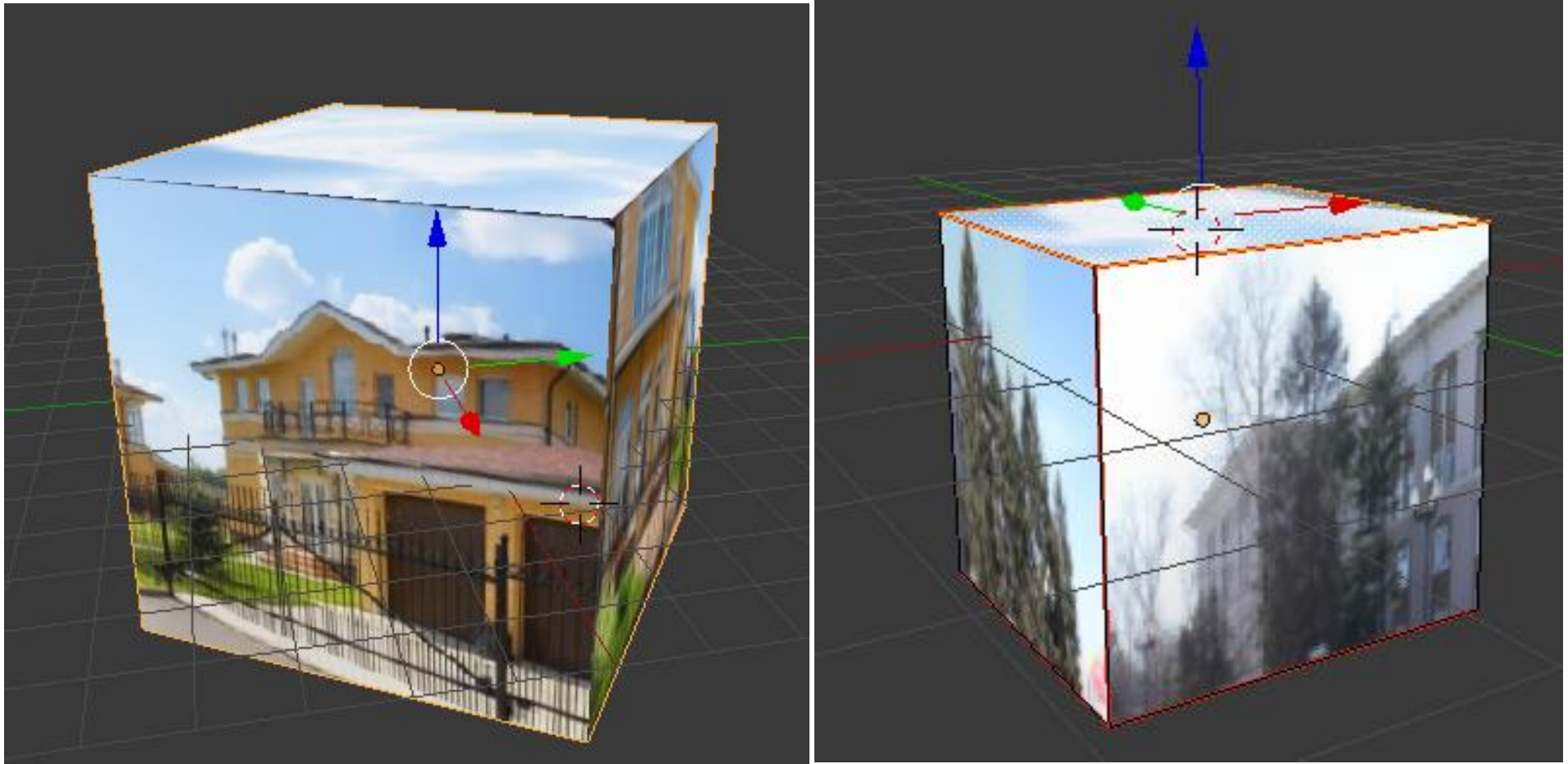


Рисунок 24 – Текстурирование

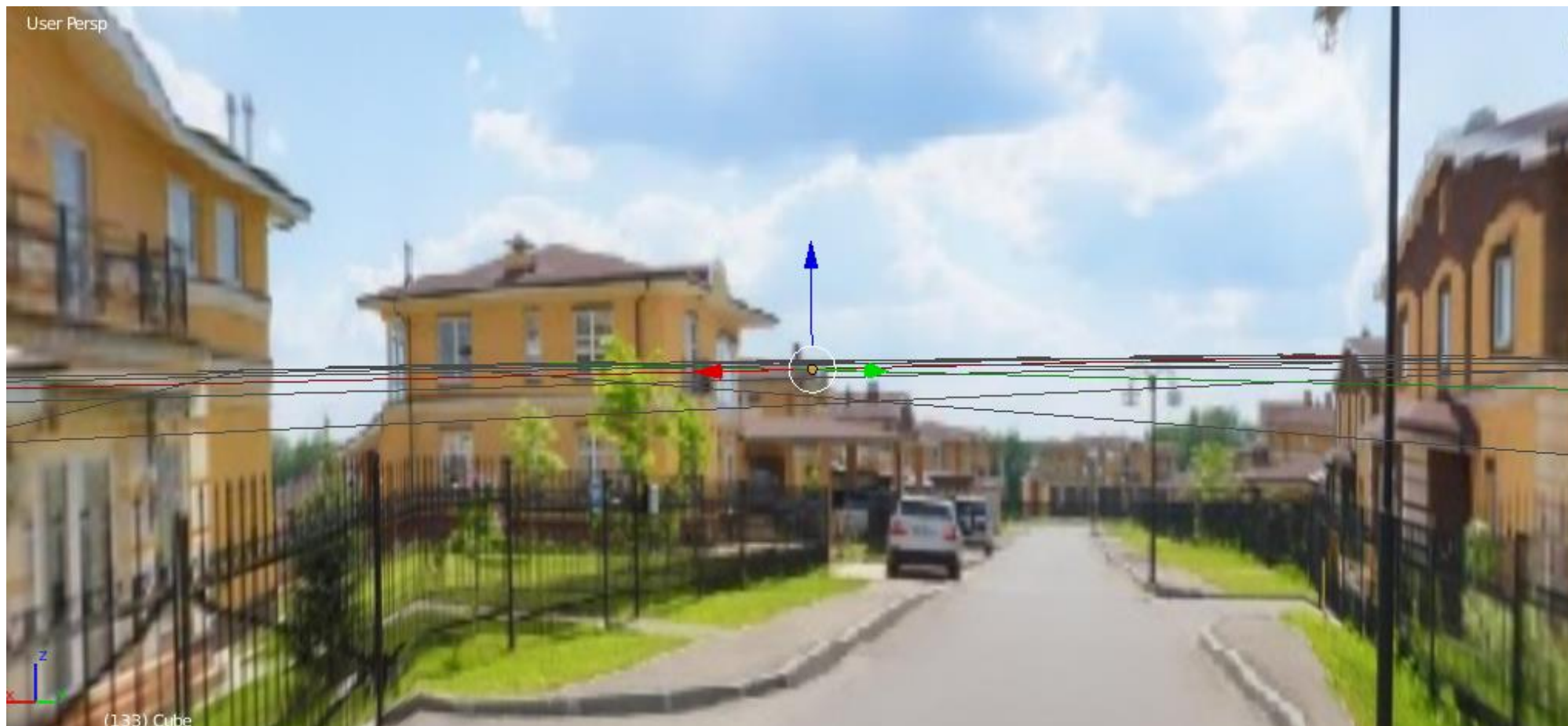


Рисунок 25 – CubeMap



Рисунок 26 – СибМар

ПРИЛОЖЕНИЕ Б

Раздел 2

ALGORITHMS FOR CREATING PANORAMIC IMAGES

Студент:

Группа	ФИО	Подпись	Дата
8ИМ5А	Толеген Молдир Орынбасаркызы		

Консультант проф. кафедры:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент каф. ИСТ	<u>Мирошниченко</u> <u>Евгений</u> <u>Александрович</u>	к.т.н.		

Консультант – лингвист кафедры ИЯИК:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Старший преподаватель	<u>Горбатова Татьяна</u> <u>Николаевна</u>			

INTRODUCTION

For decades, scientists from different countries have been working on the development of algorithms that make it possible for computers to acquire visual information in the way people do. It is normal state of things for human beings to receive the information visually. However the mission to train computers to do the same is still impossible today. Many IT corporations are working to solve this issue, but it requires a lot of efforts, financial expenses and computing power. There are methods that allow to get the desired result in narrowly focused tasks at a lower cost. They, as a rule, are based not on the structure of the human apparatus which is responsible for the analysis and interpretation of images, but directly on the artificially isolated features of the image. This work is devoted to solving the task of creating panoramic images. The solution of the task of creating panoramic images is required when developing virtual tours and excursions, modeling, virtual reality systems, cinematography. Some of these methods are based on finding singular points and their numerical description. Key points are those which are used to classify an image, recognize it and its features. Based on these data, it is possible to achieve a higher accuracy of automatic image content analysis with less time and resource costs. The effectiveness of such algorithms is significant when working with mobile devices such as smartphones, which are widely used today.

Panoramic image is a photograph obtained by the technology of assembling panoramas from individual frames, with a large viewing angle exceeding the capabilities of conventional lenses, and often more than 180 degrees, or a conventional photograph of a "long" format, with a 1×2 , 1×3 aspect ratio and more. The rapid development of digital photos has given birth to panoramic images. The world of panoramic photos allows us to see a more realistic picture of the captured images. Panoramas of cities, nature, space, panoramic maps immerse us in a new world of digital photos. In one panorama you can often see a larger angle of view than the human eye can see. The aim of this work is to create and develop an algorithm for the synthesis of panoramic images from a series of photographs.

Obtaining of panoramic images includes the following steps:

1. key points allocating;
2. non-key points screening off;
- 3 key points matching;
4. images stitching.

2 REVIEW OF ALGORITHMS FOR CREATING PANORAMIC IMAGES

2.1. Overview of the algorithms used to search key points

The key point \mathbf{m} is the point of the image which neighborhood $\mathbf{o}(\mathbf{m})$ can be distinguished from the neighborhood of any other point of the image $\mathbf{o}(\mathbf{n})$ in some other neighborhood of the key point $\mathbf{o}_2(\mathbf{m})$. Tuytelaars and Mikolajczyk (2006) defined the following properties for the key points:

Repeatability is a key point located in the same place of the scene or object of the image, despite of the changes in lighting and at the viewpoint.

Distinctiveness is the neighborhoods of key points must have great differences from each other, so that it is possible to identify and compare the singular points.

Locality is a key point should occupy a small area of the image in order to reduce the likelihood of sensitivity to geometric and photometric distortions between two images taken at different points of view.

Quantity is used for the detection of small objects. The number of detected key points must be enough. The number of detected key points should be determined adaptively by using an intuitive and simple threshold. To ensure the density of the location of key points, a compact representation should show the information content of the image.

Accuracy means that the found key points should be precisely localized, both in the initial image and in another scale.

Efficiency is the time of finding the key points on the image should be acceptable in time-critical applications.

Approaches to the definition of key points can be divided into 3 categories:

1. Based on the intensity of the image: the singular points are calculated directly from the pixel intensity values of the image.
2. Using image contours: methods for extracting contours and finding places with the maximum curvature or polygonal contour approximation and determining

intersections. These methods are sensitive to intersections of intersections, since extraction can often be incorrect in those places where 3 or more edges intersect.

3. Based on the use of the model: intensity models are used as parameters that adapt to images up to sub-pixel accuracy. They have limited use with key points of special types (for example, L-connected angles), depend on the templates used.

It is necessary to choose the points that contribute to the description of the image; also it is necessary to consider the points that are very likely to be found on another (similar) image as very important ones. Each method for detecting key points must guarantee invariance with respect to affine and perspective transformations.

The *descriptor* of a key point is a numerical vector that characterizes the features of the image in the vicinity of a given point. Initially, the procedure for constructing descriptors was used to find the correspondence between key points on different images. As a result of constructing descriptors for the initial set of key points, a set of characteristic vectors is generated.

Now the most common descriptors of key points is SIFT, which is based on the idea of calculating a histogram in the neighborhood of a key point of oriented gradients. The modification of this descriptor is PCA-SIFT [3], which is built in the same way, using larger environments. The dimensionality of vectors for the resulting set of descriptors is reduced by analyzing the main components (PCA) [2].

The SURF descriptor [1] also refers to the number of descriptors that simultaneously search for key points and construct their description, which is invariant to changing the scale and rotating the image. In fact, this descriptor is the weighted values of the gradients and their absolute values in 16 quadrants to which the neighborhood of the key point is broken.

The purpose of creating the BRIEF descriptor was to provide recognition of the same parts of the image obtained from different viewing angles. The task was to reduce the number of calculations performed. In a neighborhood of a point, a set of pairs of pixels is selected in a certain way, and a set of binary tests is constructed on this set, which is a comparison of the intensity in the corresponding pixels. The test results are

combined into a bit string, which is the final key point descriptor. A more efficient alternative to the BRIEF descriptor is the binary ORB descriptor.

FAST algorithm does not require the calculation of the brightness derivatives. Figure 1 shows the analyzed point and the surrounding pixels. The brightness of the pixels lying on the circle is compared with the brightness of the central point and on the basis of a series of checks a decision is made whether the central point is characteristic. The sequence of checks and their total number are selected and optimized in advance based on an extensive training sample of images. As a result, the checks are carried out very quickly. To decide whether the point is an angle or not, only a few dozen comparison operations are required. The FAST algorithm has proven itself in applications that track objects in real time.

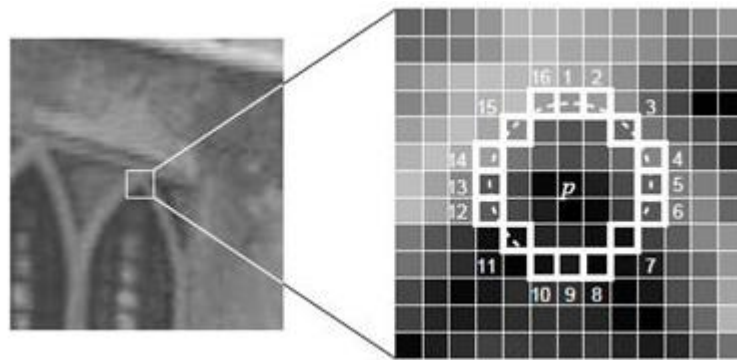


Figure 1 - Pixels used in fast tests of the FAST algorithm

ORB is an oriented FAST and Rotated BRIEF is upgrade version and a combination of a FAST-point detector and BRIEF binary descriptors. ORB has one important nuance is the size of the descriptors is 32 bytes per point. The coincidence test is the sum of the Hamming distances for each byte of the descriptor (the first is compared with the first, the second with the second, and so on).

Next, we will consider the following methods for extracting key points and obtaining their descriptors: SIFT [6], SURF [1], and ORB [5].

2.1.1 The SIFT algorithm

The SIFT algorithm is a scale-spatial detector: in the first stage, key points are calculated on images of different scales. After that, the detected key points are filtered,

after which the most stable ones remain. For each key point found, one or more principal orientations are determined, and descriptors are calculated [6].

The algorithm consists of the following steps [1]:

- 1) the construction of the pyramid of Gaussians and their differences. This step ensures invariance to scaling;
- 2) the declaration of extrema;
- 3) qualification of key points;
- 4) the construction of descriptors (ensuring the invariance to lighting, noise, changing the position of the camera).

SIFT allows you to define key points in the form of blobs, since they are invariant to all transformations. Blobs are structures described by the coordinates of the center, scale and direction. This type of structure is the most complex, high-level among all forms of key points and, therefore, ensures their stable detection.

At the first stage of the SIFT algorithm, a scalable image space is constructed is a set of images smoothed by a Gaussian filter

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}, \quad (1)$$

where (x, y) are the coordinates of the point; σ is the radius of the feathering.

They have a Gaussian difference $D(x, y, \sigma)$ is subtraction by pixels of images in one octave with alternative feathering radius. Octave is formed by images of the same scale, feathering by a Gaussian filter with an alternative feathering radius (4 images in one octave). At this stage, invariance to scaling is ensured. Then extremes are defined, which are recorded in the list of potential key points.

The SIFT algorithm represents the best results for detecting key points and obtaining their descriptors, and approaches the application environment. In Applications B of fig.9. The result of applying the SIFT algorithm for searching for key points is given.

2.1.2 The SURF algorithm

An alternative approach is SURF, which is several times faster than SIFT. The SURF algorithm solves two problems are searching for key image points and calculating descriptors that are invariant with respect to scale and rotation. This means that the description of the key points will be the same, even if the sample changes its size and is rotated. It well detects spots, corners and edges of lines. In this approach, integrated images are used to speed up the search for points of interest.

The value at each point of the integral image is calculated as the sum of the values at a given point and the values of all points that are higher and to the left of the point. With the help of integral images for constant time, the so-called rectangular filters are computed, which consist of several rectangular regions [1].

The determination of the key points in the image is carried out on the basis of the Hessian matrix (FAST-Hessian detector). The application of Hessian ensures invariance with respect to the "rotating" type transformation, but not invariance with respect to the scale change. Therefore, SURF applies filters of different scale to calculate Hessian. Let's say that the original image is given by the intensity matrix I , the current pixel under consideration is denoted by $X = (x, y)$ and σ is the filter scale. Then the Hessian matrix has the following form:

$$H(x, y, \sigma) = \begin{pmatrix} L_{xx}(x, y, \sigma) & L_{xy}(x, y, \sigma) \\ L_{yx}(x, y, \sigma) & L_{yy}(x, y, \sigma) \end{pmatrix} \quad (2)$$

where

$$L_{xx}(x, y, \sigma) = I(x, y) * \frac{d^2 g_\sigma}{dx^2},$$

$$L_{yy}(x, y, \sigma) = I(x, y) * \frac{d^2 g_\sigma}{dy^2},$$

$$L_{xy}(x, y, \sigma) = I(x, y) * \frac{d^2 g_\sigma}{dx dy}.$$

Here $I(x, y)$ is the source image, $g_\sigma(x, y)$ is the Gaussian function (1), «*» stands for the convolution operator.

After that, for each found key point, the orientation is calculated as the predominant direction of the brightness difference. The concept of orientation is close to the concept of the direction of the gradient, but the Haar filter is used to determine the orientation of the key point. An example of Haar filters is shown in Fig. 2.



Figure 2 - Haar filters (black areas have the values «-1», white «+1»)

The descriptor is a set of lattice points of 64 (in the extended version 128) numbers that allow you to identify the key point. The descriptors must roughly coincide with the same key point on the stage and on the sample. The method of calculating the descriptor is independent of scale and rotation.

To calculate the descriptor around a key point, a rectangular area of size $20s$ is formed, where s is the scale in which a key point is found. For the first octave, the area has a size of 40×40 pixels. The square is oriented along the priority direction calculated for the key point. The descriptor is considered as a description of the gradient for 16 quadrants around a key point.

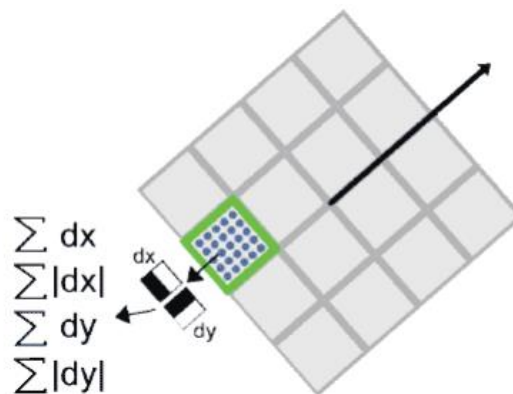


Figure 3 - Calculation of the descriptor of the key point

Then the square is divided into 16 smaller quadrants, as shown in fig.3. In each quadrant, a regular grid of 5×5 is taken and grid points are selected using the Haar filter. The size of the Haar filter is assumed to be $2s$, and for the first octave it is 4×4 .

It should be noted that in the calculation of the Haar filter the image does not rotate, the filter is viewed in the usual image coordinates. The resulting gradient coordinates (dX , dY) are rotated by an angle corresponding to the orientation of the square. In general, to calculate the descriptor of a key point, you need to calculate 25 Haar filters, in each of 16 quadrants. A total of 400 Haar filters. Given that the filter requires 6 operations, it turns out that the descriptor will cost at least 2,400 operations. After finding 25 gradients of the quadrant point, four values are calculated that are actually components of the descriptor:

$$\sum dX, \sum |dX|, \sum dY, \sum |dY|.$$

Two of them are just a full gradient on the quadrant, and the other two are the sum of the brick of point gradients.

In fig.4. The behavior for different parts of the images of these data point is shown.

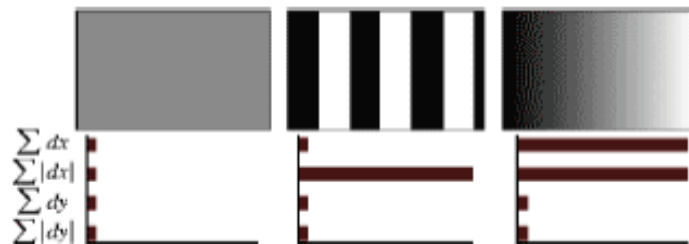


Figure 4 - Descriptor behavior for different images

For homogeneous 18 regions, all values are close to zero. For repeating vertical strips, all quantities except the second are close to zero. With increasing brightness in the direction of the X axis, the first two components have high values.

Four components per quadrant, and 16 quadrants, yield 64 descriptor components for the region of the key point. When entering a set of lattice points, the descriptor values are multiplied by Gaussian, with the center at a key point and with the $\sigma = 3.3s$. This is necessary for greater stability of the descriptor to noise in regions remote from a key point. In addition to the descriptor, the sign of the trace of the Hessian matrix is used to describe the point, that is $\text{sign}(D_{xx} + D_{yy})$. For light

points on a dark background, the trace is negative, for dark dots on a light background is positive. Similarly, SURF distinguishes between dark and light spots.

Applying SURF to the image, we get a set of descriptors that uniquely identifies the standard on the stage. A good and stable result is observed when rotating, changing the illumination of the image or the angle at which the snapshot was taken. A good method of coping with changing the brightness of the image. Applications B of fig.10. The result of applying the SURF algorithm for finding key points is given.

2.1.3 The ORB algorithm

ORB is an oriented FAST and Rotated BRIEF are upgrade version and a combination of a FAST-point detector and BRIEF binary descriptors.

The purpose of creating the BRIEF descriptor was to make sure that the same parts of the image were recognized, which were taken from different points of view. The task was to minimize the number of calculations performed. The algorithm of recognition is reduced to the construction of a randomize classification trees or a naive Bayesian classifier on some training set of images and subsequent classification of test areas of the image. In a simplified version, the nearest neighbor method can be used to find the most similar section in the training selection. A small number of operations are provided by representing the characteristic vector in the form of a binary string and using the Hamming distance as a measure of similarity [7].

The scheme for constructing feature vectors is quite simple. The image is broken into patches. Suppose the patch P has coordinates of $S \times S$ pixels. From the patch, the set of pairs of pixels $\{(X, Y), \forall X, Y \text{ in the neighborhood}\}$ for which a set of binary tests is constructed is selected in some way:

$$\tau(P, X, Y) = \begin{cases} 1, I(X) < I(Y) \\ 0, \text{ иначе} \end{cases} \quad (3)$$

where $I(Y), I(X)$ is the intensity of the pixels Y, X , respectively. For each patch, a set is selected that contains n_d pairs of points that uniquely determine the set of binary tests. Further on the basis of these tests a binary string is constructed:

$$f_{n_d}(P) = \sum_{1 \leq i \leq n_d} 2^{i-1} \tau(P, X_i Y_i) \quad (4)$$

The authors give the results of experiments (clarification quality) in the selection of pairs of points according to the law of uniform distribution in the patch, as well as the normal distribution with different values of the mathematical expectation and standard deviation [5]. Note that under identical experimental conditions for some test images, the detection accuracy with BRIEF is almost 1.5 times higher than using SURF descriptors [7].

The ORB algorithm quickly calculates the key points and calculates their descriptors, which makes it possible to use it in tasks where real-time image processing is required. ORB uses FAST to find key points. Applications B of fig.11. The result of applying the ORB algorithm for searching for key points is given.

A brief overview of the most popular existing methods makes it possible to conclude that all descriptors have their merits, showing a weak invariance to various transformations of scale, shift, and rotation. This calls for the improvement of existing methods or the development of new ones.

2.2. Key points matching

After carrying out the search and description of the key points, it is necessary to choose an algorithm for their matching. Matching can be carried out on the basis of various metrics. In this paper we consider the Euclidean and Hamming metrics.

2.2.1 The Euclidean distance

Comparing each set of descriptors from the first image with each such set from the second image, we find the best correspondence between the key points. Matching up is calculated on the basis of the Euclidean distance:

$$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}, \quad (5)$$

where d is the spacing x and y ; x_i is the value of the i -object properties x ; y_i is the value of the i - object properties y ; $i - (1, 2, \dots n)$.

2.2.2 Hamming distance

Declaration 1. A metric of the difference in objects given by dichotomous tags.

It is determined using the formula:

$$d_H(X_i X_j) = \sum_{s=1}^p |x_i^{(s)} - x_j^{(s)}| \quad (6)$$

And, consequently, is equal to the number of discrepancies between the values of the corresponding characteristics in the considered i -th and j -th objects.

Declaration 2. The Hamming distance is the number of distinct bits in two binary vectors.

The Hamming distance is already widely used for various tasks, such as finding duplicates, artificial cognition, classifying documents, purification, detecting viruses and etc. In a more general case, the Hamming distance is applied to strings of the same length of any k -ary alphabets and serves as the metric of the difference (a function that determines the distance in a metric space) of objects of the same nullity. Such a metric can be represented in the form of formula 7.

$$hamming(x, y) = \sum_{x_i \neq y_i} 1, \quad i = 1, \dots, n \quad (7)$$

where x and y are descriptors (binary vectors).

2.3. Stitching images

For stitching images, one of them must first be subjected to some geometric transformation.

Geometric transformations are transformations of graphic objects in which the direction, scale and position of the coordinate system are used to determine the position of points in space. In this case, the space itself is not subjected to any changes, i.e. The structure of the graphic image itself is saved. Therefore, for geometric transformations, the type of the object is not important, since the points participate in the transformations.

2.3.1 Geometric transformations

The main operations associated with geometric transformations of two-dimensional graphic objects are:

1. Change of the origin of coordinates
2. Scaling
3. Rotation of the coordinate axes

The matrix required to perform the carry operation is as follows:

$$T = \begin{bmatrix} 1 & 0 & -Tx \\ 0 & 1 & -Ty \\ 0 & 0 & 1 \end{bmatrix}$$

The matrix required to perform the scaling operation is as follows:

$$S = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The following matrix is used to perform the rotation operation:

$$R = \begin{bmatrix} \cos w & \sin w & 0 \\ -\sin w & \cos w & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

For overloading, one of the input images must be subjected to a geometric transformation. To do this, we need to find the transformation matrix 3×3 . To find the transformation matrix, you need 4 points on the input image and 4 corresponding points on the output image. Among these 4 points, 3 should not be collinear.

To take into account projective tampering, one can use transformations in the matrix form in homogeneous coordinates

$$\begin{pmatrix} \lambda_x \\ \lambda_y \\ \lambda \end{pmatrix} = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{pmatrix} \cdot \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix}$$

The homography matrix $H=(h_{ij})_{3 \times 3}$ in the formula

$$\begin{cases} x = \frac{h_{11}x' + h_{12}y' + h_{13}}{h_{31}x' + h_{32}y' + 1} = \frac{\lambda_x}{\lambda} \\ y = \frac{h_{21}x' + h_{22}y' + h_{23}}{h_{31}x' + h_{32}y' + 1} = \frac{\lambda_y}{\lambda} \end{cases} \text{ contains 8 unknown parameters to find that you need at}$$

least 4 key pairs on the mapped images. The formation of a set of pairs of key points, on the one hand, and the selection of the optimal set of points from this set that provide

a correct definition of the elements of the homography matrix, on the other hand, the two main tasks in the problem of combining images with the help of the homography matrix.

2.3.2 The RANSAC algorithm

RANSAC is a method that is used to evaluate model parameters based on random samples. When matching, the model is a transformation matrix (homography). At the input of the algorithm, there are two sets of descriptors obtained on the previous and current image.

The scheme of RANSAC consists in repeating the following stages:

1. Select points and construct model parameters containing 8 unknown parameters. To search for the homography matrix, at least 4 pairs of key points on the images to be compared are necessary. Based on the obtained sets, a transformation matrix is constructed.

2. Verification of the constructed model. For each point of the previous frame, the projection on the current frame is located, and the closest descriptor (point) from the plurality of descriptors of the current frame is searched. A characteristic point is marked as an "ejection" if the distance between the projection and the corresponding descriptor of the current image is greater than the specified cutoff.

3. Override model. It is checked whether the model constructed is the best among the set of previous models.

As a result of applying RANSAC, the best homography matrix is constructed, which will be used to search for false matches.

2.3.3 Generation of three-dimensional objects

Creating a three-dimensional model can be implemented by using a graphical library. In this work, the Unity library was chosen. The approach to creating and optimizing a 3D terrain model can be applied to the development of interactive applications based on Unity3D for various types of applications (simulators, GIS systems), which makes it universal and useful.

Unity is a tool for developing two- and three-dimensional applications and games. An important feature of this engine is its cross-platform. Projects created in Unity are well-going for different devices such as Android, iOS, Windows, Mac, PlayStation, Xbox, many other consoles and even SmartTV. Three-dimensional objects in computer graphics in general, and in Unity in particular, are very often represented by polygonal meshes.

Polygonal mesh is a set of vertices, faces and edges that define the shape of a three-dimensional object. Triangles and quadrangles are called polygons and they make the edges of Unity. The polygon connects the points of the figure, forming a single, closed object.

To visualize a closed panoramic image, you must superimpose the resulting panorama texture on a closed shape: Cubemap, Sphere, or Cylinder.

Cubemap is a set of six square textures that represent a reflection on the environment. Six squares form the faces of the imaginary cube that surrounds the object; each face is a view along the direction of the world axes (up + Y, down -Y, right-X, left + X, forward + Z, back -Z).

Vertical and horizontal transverse layouts, as well as the column and rows of cubemap faces are presented in fig.5.

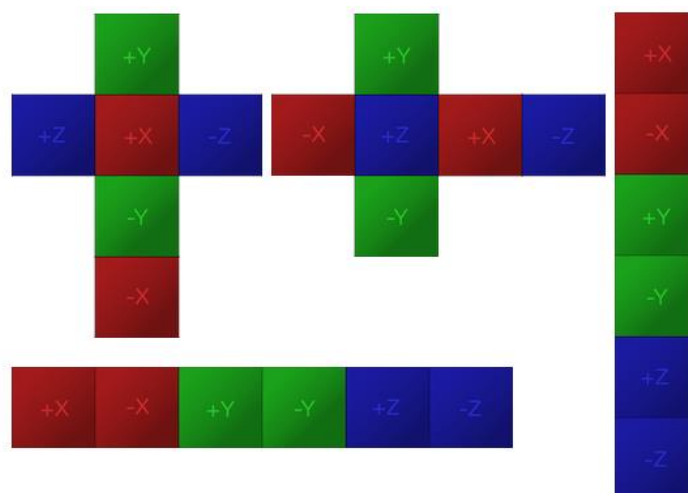


Figure 5 - Cubemap reamer

Another common layout of LatLng (latitude-longitude, sometimes called cylindrical). The reamer of the cylinder is shown in fig.6. Panoramic images are located in this scheme:

1. Loading a texture with a panoramic image;
2. The texture is divided into equal blocks;
3. The texture overlaps the corresponding face of the cylinder.

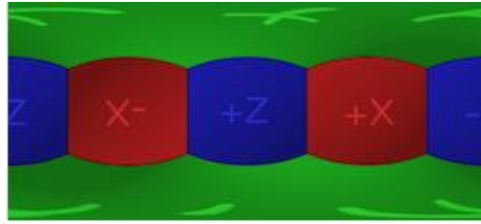


Figure 6 - Cylinder reamer

The next one is the layout of the SphereMap (Spherical Environment Map) image. The reamer of the sphere is shown in fig.7.

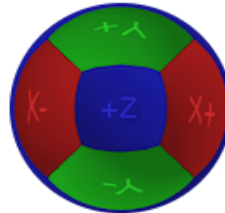


Figure 7 - Sphere reamer

In this paper, it was decided to use the cylinder as a basic primitive, since this form better corresponds to the shape of the panoramic image we obtained.