

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ

«ТЮМЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ НАУК
Кафедра информационных систем

ДОПУЩЕНО К ЗАЩИТЕ В ГЭК
И ПРОВЕРЕНО НА ОБЪЕМ
ЗАИМСТВОВАНИЯ

Заведующий кафедрой

д.т.н., профессор

И.Н. Глухих

2016 г.



МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

ЭКСПЕРТНАЯ СИСТЕМА АНАЛИЗА ВИЗУАЛЬНОГО ВОСПРИЯТИЯ
ЗРИТЕЛЬНЫХ ОБРАЗОВ В ОБЛАСТИ НЕЙРОМАРКЕТИНГА

09.04.03 Прикладная информатика

Магистерская программа «Прикладная информатика в экономике»

Выполнила работу
Студентка 2 курса
очной формы



Ерёменко
Кристина
Вячеславовна

Научный руководитель
доцент кафедры ИС, к.т.н.



Карякин
Иван
Юрьевич

Рецензент
Генеральный директор
ООО «Синапс»

Юсупов
Дмитрий
Альбертович

АННОТАЦИЯ

Тема работы: Экспертная система анализа визуального восприятия зрительных образов в области нейромаркетинга.

Актуальность: Разрабатываемое программное приложение создавалось для учебного класса компании ТрансНефть. Суть проблемы заключается в том, что для изучения экспонатов различных насосов и т.п. требуется выезд на объекты или на места размещения этих учебных макетов. Что требует отрыва учащегося от изучения соответствующего материала. Актуальным решением в данной ситуации является совмещение печатной продукции с медиа-материалами в качестве видео и 3D-моделей.

Цель работы: увеличить эффективность обучения, информативность печатной продукции. Увеличить качество обучения.

Разрабатываемая система предназначена для воспроизведения дополненной реальности в учебных материалах. Система предоставляет возможность показать устройство сложного оборудования без его реального присутствия, его ремонт, так же разные учебные видео материалы в печатной продукции.

Задачи, решаемые в работе: 1) Изучить предметную область; 2) Обзор существующих решений дополненной реальности; 3) Построить модель данных; 4) Выбрать технологии разработки; 5) Разработать диаграмму прецедентов; 6) Разработать интерфейс; 7) Разработать модель архитектуры; 8) Разработать браузер дополненной реальности.

Полученные результаты: 1) Изучена предметная область; 2) Проведен обзор существующих библиотек дополненной реальности и выбран инструмент разработки; 3) Построены ER диаграммы, логическая и физическая диаграммы в нотации IDEF1x, диаграмма прецедентов; 4) Разработаны интерфейс и модель архитектуры; 5) Построены диаграммы деятельности описывающие, алгоритм работы браузера; 6) Разработан браузер дополненной реальности.

ОГЛАВЛЕНИЕ

| | |
|--|----|
| ВВЕДЕНИЕ..... | 5 |
| ГЛАВА 1 ДОПОЛНЕННАЯ РЕАЛЬНОСТЬ..... | 7 |
| 1.1 Определение понятия «Дополненная реальность»..... | 7 |
| 1.2 Классификация систем дополненной реальности | 8 |
| 1.3 Устройства дополненной реальности | 10 |
| 1.4 Применение дополненной реальности | 12 |
| 1.5 Маркеры дополненной реальности | 14 |
| 1.6 Компьютерное зрение..... | 15 |
| 1.7 Основные методы захвата меток..... | 17 |
| 1.8 Распознавание маркера дополненной реальности..... | 22 |
| 1.9 Построение дополненной реальности..... | 30 |
| 1.10 Будущее Augmented Reality | 33 |
| 1.11 Выводы..... | 33 |
| ГЛАВА 2 ПОСТАНОВКА ЗАДАЧИ..... | 35 |
| 2.1 Название..... | 35 |
| 2.2 Назначение..... | 35 |
| 2.3 Цель | 35 |
| 2.4 Задачи..... | 35 |
| 2.5 Список разрабатываемых алгоритмов | 35 |
| 2.6 Функциональные особенности разрабатываемой системы | 36 |
| 2.7 Выбор платформы реализации | 36 |
| 2.8 Анализ существующих решений..... | 37 |
| 2.9 Выводы..... | 42 |

| | |
|--|----|
| ГЛАВА 3 ПРОЕКТИРОВАНИЕ БРАУЗЕРА ДОПОЛНЕННОЙ РЕАЛЬНОСТИ | 43 |
| 3.1 Построение функциональной модели..... | 43 |
| 3.2 Построение моделей данных | 44 |
| 3.3 Построение модели архитектуры..... | 52 |
| 3.4 Интерфейс | 54 |
| 3.5 Выводы..... | 60 |
| ГЛАВА 4 РЕАЛИЗАЦИЯ БРАУЗЕРА ДОПОЛНЕННОЙ РЕАЛЬНОСТИ | 61 |
| 4.1 Компоненты браузера дополненной реальности | 61 |
| 4.2 Алгоритм воспроизведения дополненной реальности..... | 63 |
| 4.3 Распознавание (обнаружение) маркера | 65 |
| 4.4 Построение 3D пространства..... | 81 |
| 4.5 Отображение виртуального объекта | 82 |
| 4.6 Выводы..... | 83 |
| ЗАКЛЮЧЕНИЕ | 85 |
| СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ | 88 |
| ПРИЛОЖЕНИЕ А | 93 |

ВВЕДЕНИЕ

Цель учебного центра компании ТрансНефть – повысить качество учебного материала. Для достижения цели был рассмотрен вариант создания советующей системы, которая будет анализировать графические объекты в учебных материалах и выводить рекомендации по их корректировкам с точки зрения восприятия человеком. В результате изучения этого вопроса сделан вывод, что существует ряд барьеров, которые не позволяют создать подобный программный продукт.

Большинство учебного материала печатной продукции содержит в себе текст, рисунки и ссылки на другие источники. Учебные материалы недостаточно информативны, не создают полную визуальную картинку описываемого оборудования и вариантов сборки, ремонта и т.п. Для изучения сложного оборудования данный подход обучения может быть не достаточно эффективен. Для увеличения эффективности обучения необходимо увеличить информативность учебников. Для этого учебник помимо текста и изображений должен содержать в себе видео материалы, а также 3D модели.

Дополненная реальность позволяет ввести в поле восприятия любые сенсорные данные для дополнения сведений и улучшения восприятия информации. Она создается с использованием элементов в реальности, дополняемых с помощью объектов, созданных через компьютер.

Благодаря камерам, встроенным в устройства, можно получать изображения в реальном времени, а вычислительная мощность и набор датчиков в аппаратных платформах для смартфонов и планшетов позволяют производить наложения любых цифровых данных над этими изображениями. Все это привело к увеличению известности дополненной реальности и к распространению технологии.

Разрабатываемая информационная система позволяет добавить в печатный материал видео и 3D модели. При наведении на изображения (маркер) в печатном учебном материале система отобразит соответствующее

видео или 3D модель. Данный подход позволит увеличить содержательность учебных материалов. Повысить качество визуального восприятия графических объектов в учебном материале средствами нейромаркетинга. Сократить временные затраты на ознакомление с макетами оборудования.

Функционал разрабатываемой информационной системы:

1. Создание нового учебного материала
2. Загрузка изображения
3. Создание маркера
4. Добавление к маркеру содержимого: видео, 3D модель оборудования и т. д.
5. Распознавание маркера
6. Воспроизведение содержимого маркера
7. Просмотр списка всех учебных материалов
8. Просмотр статистики по количеству распознаваний маркера

Перечень разрабатываемых моделей и алгоритмов:

- диаграмма прецедентов (Показывает взаимодействие пользователя и программы);
- логическая модель данных IDEF1X (Описывает структуру БД);
- физическая модель данных (Описывает структуру БД применительно к выбранной СУБД MySQL);
- алгоритм распознавания (обнаружения) маркера: приводим в градации серого, определение особых точек, вычисление дескрипторов особых точек;
- захват маркера;
- построение 3D пространства;
- отображение виртуального объекта.

ГЛАВА 1 ДОПОЛНЕННАЯ РЕАЛЬНОСТЬ

1.1 Определение понятия «Дополненная реальность»

Дополненная реальность (англ. *augmented reality*, AR — «расширенная реальность») — итог внедрения в область восприятия различных сенсорных данных с целью добавления информации об окружающей среде и улучшения восприятия данных, формируется с применением «дополненных» с помощью компьютера элементов воспринимаемой реальности (когда реальные предметы монтируются в поле восприятия).

Обобщая определение можно сказать, что дополненная реальность представляет собой одновременный просмотр виртуальных объектов и реального мира, где одна информация накладывается и интегрируется с другой. Дополненная реальность находится в диапазоне от «реальности» до виртуальной реальности «полного погружения» (см. Рисунок 1).

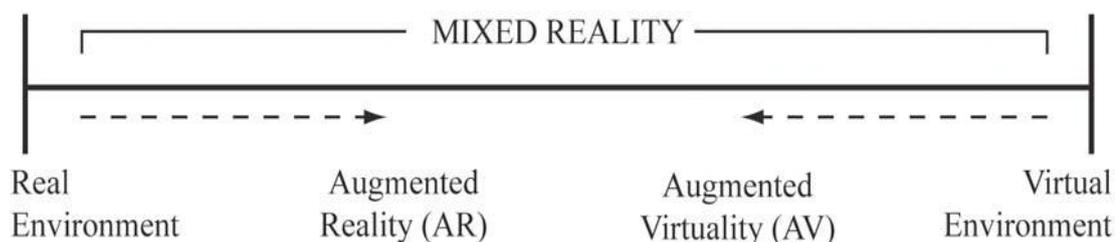


Рисунок 1. Диапазон технологий смешанной реальности

Отличительным признаком дополненной реальности от виртуальной реальности является сохранение реального мира как фона, в котором представлены виртуальные объекты. Виртуальная же реальность полностью отстранена от реального мира.

Таким образом, дополненная реальность (*augmented reality*) – это технологии, которые позволяют добавлять к живым объектам различные компьютерной графики, а также соединять изображения, полученные от разных источников компьютерной среды: видеокамер, акселерометров, компасов и т.д. Схема среды дополненной реальности представлена на рисунке (см. Рисунок 2). [1]

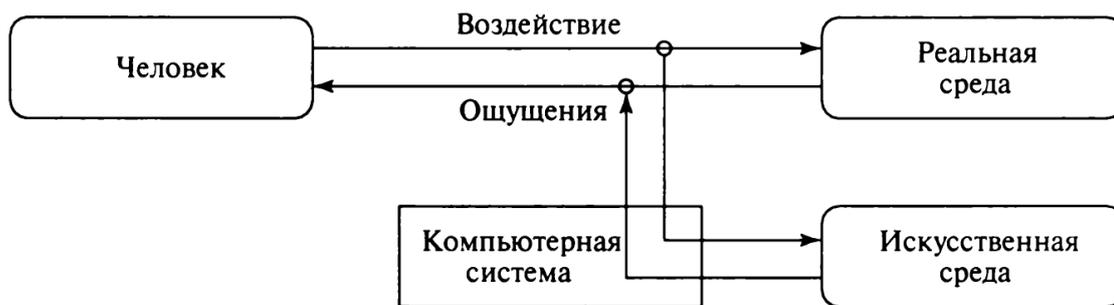


Рисунок 2. Схема среды дополненной реальности

Одним из ярких примеров дополненной реальности могут быть спортивные трансляции: во время штрафного удара на футбольном матче, когда на телевизионном экране показываются линии с уточнением расстояния от места штрафного удара до ворот и т. д.

Дополненная реальность дает возможность человеку чувствовать настоящий мир совместно с виртуальными объектами, наложенными на окружающую обстановку. Таким образом, дополненная реальность интегрируется и расширяет реальный мир вместо того, чтобы целиком его заменить.

Рональд Азума [2] выделил ряд признаков, которыми должна обладать расширенная реальность:

- 1) комбинирование реального и виртуального мира;
- 2) интерактивность;
- 3) трехмерное представление объектов.

1.2 Классификация систем дополненной реальности

AR системы можно классифицировать разными методами. Согласно виду представления информации они бывают:

- зрительные – источником информации в таких системах для человека является изображение. В основе их лежит визуальное восприятие человека. Задача такой системы – сформировать изображение для использования человеком.

Такой тип считается наиболее популярным, так как картинка для человека считается более информативным и понятным;

- аудио – информация подается в виде звука. Такие системы направлены на слуховое понимание. Характерные черты подобных систем рассматриваются в работе М. Tikander [3]. Подобные системы применяются в навигации. Например, использование звука для ориентации в пространстве, человек идет в нужном направлении, ориентируясь на звук.[4]
- аудиовизуальные – системы, соединяющие в себе аудио и визуальные типы.

Виртуальные объекты дополненной реальности строятся на основе информации полученной из физического мира. Для получения этой информацию используются разные устройства. Согласно виду приборов, с которых AR система получает информацию о реальном мире:

- геопозиционные – подобные системы ориентируются на сигналы систем позиционирования GPS или ГЛОНАСС [5], также могут применять дополнительно компас и акселерометр для определения угла поворота относительно вертикали и азимута;
- оптические – для таких систем источником информации является изображения. Устройства позволяющие получать изображения – различные камеры.

По уровню мобильности системы классифицируют:

- стационарные – транспортировка систем данного вида приводит к сбою деятельности, вследствие чего их нельзя перемещать;
- мобильные – подобные системы возможно транспортировать.

Системы AR можно различать согласно уровню взаимодействия с пользователем. В одних системах пользователь активно управляет работой

системы, в других системах является лишь наблюдателем. По этому признаку системы можно разделить на:

- автономные – системы, цель которых состоит в том, чтобы обеспечить пользователя нужной информацией. Для работы системы не требуется вмешательства пользователя. Задача таких систем сводится к предоставлению данных о предмете. Системы подобного вида используют в медицине;
- интерактивные – системы, в которых совершается интенсивное взаимодействие с пользователем, который с ней взаимодействует. Подобные системы базируются на взаимодействии с пользователем. На свои действия пользователь получает ответ. В таких системах используется прибор для ввода информации. В качестве такого прибора может использоваться сенсорный дисплей планшета, телефона или специальное устройство.

1.3 Устройства дополненной реальности

Основными устройствами, применяемые в системах дополненной реальности являются дисплеи, устройства ввода, устройства отслеживания и компьютер

Дисплеи

Имеется три главных вида дисплеев, применяемых в дополненной реальности: ручные дисплеи, пространственные и head mounted displays (HMD).

Ручной дисплей – небольшой вычислительный прибор с экраном, который пользователь способен удерживать в руках. Они применяют видео-прозрачные способы для наложения графики на реальную среду, а так же применяют вспомогательные измерители (GPS, компас). Для реализации AR систем зачастую применяются системы распознавания меток, такие как

ARToolKit, либо способы компьютерного зрения, наподобие SLAM. Подобными приборами могут быть: телефоны, планшеты. Телефоны переносные и широко распространённые приборы, сочетающие мощнейшие процессор, камеру, набор необходимых сенсоров, что делает их очень многообещающей платформой для систем дополненной реальности. Планшеты также являются перспективной платформой для работы с AR.. [6]

Пространственная дополненная реальность (SAR) с применением зрительных компонентов, видеопроекторов, голограмм, и иных технологий предоставляет графические данные напрямую на физические предметы, не призывая пользователя держать или носить дисплей. Пространственные дисплеи отделяют большую часть технологий от пользователя и интегрируют их в окружающую среду. Это дает возможность SAR естественно масштабироваться до групп пользователей, что дает возможность реализовать их совместную работу, увеличивая заинтересованность к подобным AR систем в лабораториях, институтах, музеях.

HMD является устройством, фиксируемым на специальном шлеме или на голове у пользователя и которое размещает перед глазами пользователя изображения действительной и виртуальной среды.

Устройства ввода

Существует большое количество видов устройств ввода для систем дополненной реальности. Многие системы применяют перчатки. Другие - применяют беспроводные браслеты. В случае смартфонов, телефон сам по себе является указательным прибором. Подбор прибора ввода в большей степени зависит от вида разрабатываемой системы и вида выбранного дисплея.

Устройства отслеживания

Приборами слежения считаются цифровые камеры либо прочие оптические датчики, компасы, GPS, акселерометры, беспроводные датчики и

т.д. Любая из этих технологий обладает разным уровнем точности и сильно зависит от вида системы.

Компьютер

AR система должна владеть мощным процессором и обладать необходимым объемом оперативной и видео памяти, с целью обработки изображений с камеры. С формированием технологий и появлением новых переносимых и в то же время мощных устройств, подобных смартфонам и планшетами, проблемы недостатка мощности отошли на второй план.

1.4 Применение дополненной реальности

Разработка и производство

Применение AR технологии увеличивает эффективность разработки нового продукта. Взамен того, чтобы вновь и вновь создавать физические прототипы, дополненная реальность дает возможность фирмам применять виртуальные модели САПР, соединенные с реальными приборами. Это экономит время и может помочь выявить ошибки на ранних стадиях проектирования, дает понять как будет функционировать очередное улучшение продукта.

Дополненная реальность упрощает работу между различными отделами, задействованными в разработке продукта, сокращает временные и материальные затраты. [7]

Обслуживание и ремонт

Автомобильная индустрия была первопроходцем в использовании решений дополненной реальности для сервиса и ремонта. В настоящее время уже реализованы такие сценарии, что технику достаточно лишь посмотреть через смарт-очки под крышку машины, и он уже знает, что в данный момент нужно сделать. Ему, шаг за шагом, показываются рабочие процессы и он может уверенней выполнять свою работу.

Обучение персонала

AR, как показано в предыдущих примерах, облегчает прототипирование, производство и техническое сервис. Дополненная реальность кроме того гарантирует продуктивный способ обучения персонала с целью сервиса и ремонтных работ техники.

При применении в обучении, на семинарах либо презентациях, AR упрощает восприятие сборочной документации, инструкции по эксплуатации, чертежей. Преподаватели могут лучше объяснить стажерам механизм внутренних компонентов и машин.

Маркетинг и продажи

Применяя дополненную реальность в маркетинге можно улучшить взаимодействие с продуктом и общее впечатление о продукте, а также узнаваемость бренда.

С помощью дополненной реальности можно накладывать дополнительную информацию непосредственно на упаковку или на сам продукт. Все это помогает привлечь клиента, вызвать интерес к продукту и к его детальному рассмотрению.

AR для обслуживания клиентов

Уже после совершения приобретения, покупатели продолжают производить оценку продукта и услуги в соответствии с их начальными ожиданиями. Немаловажно свести к минимуму уровень неопределенности, чтобы не вызвать ощущения расстройства у покупателя.

Руководство пользователя

Дополненная реальность может помочь пояснить, как правильно применить продукт.

Навигация

Обслуживание – это не только послепродажная помощь, но также вежливое отношение к возможным покупателям. В парках развлечений, на пути к туристическим достопримечательностям, или выставкам, хороший сервис начинается с удобных и интересных указателей местности.

Примером использования может служить аквариум в Токио, который применил технологии AR для своего приложения. Приложение, используя GPS координаты, показывало пингвинов, идущих к аквариуму и указывающих дорогу.[8] [9] [10]

1.5 Маркеры дополненной реальности

Возможно ответить два основных принципа построения дополненной реальности:

- на основе маркера;
- на базе координат местоположения пользователя.

Без маркерные технологии нередко используются в мобильных аппаратах, и строятся с помощью специализированных датчиков: гироскоп, акселерометр, GPS-приёмник, магнетометр.

Под маркером подразумевается предмет, находящийся в окружающем пространстве, который располагается и анализируется особым программным обеспечением с целью дальнейшей отрисовки виртуальных объектов. На базе информации о состоянии маркера в пространстве, программа способна довольно точно спроецировать на него виртуальный объект, от чего будет достигнут эффект его физического присутствия в окружающем пространстве. Применяя вспомогательные графические фильтры и высококачественные модели, виртуальный предмет способен быть почти настоящим и сложно непохожим с другими компонентами интерьера или экстерьера.

Нередко в роли маркера представляют лист бумаги со особым рисунком. Вид изображения может колебаться достаточно сильно и зависит от алгоритмов распознавания изображений. Вообще говоря, большое число

маркеров довольно широко: ими могут быть и геометрические фигуры простой формы (например, квадрат, треугольник), и объекты в форме прямоугольного параллелепипеда, и даже глаза и лица людей.

1.6 Компьютерное зрение

Концепция компьютерного зрения (англ. computer vision) считается основополагающей для развития технологий дополненной реальности, и в первую очередь в области применения маркеров. Главное направление данной дисциплины – это исследование и обработка изображений (в том числе и видеопотока). Алгоритмы компьютерного зрения дают возможность выделять ключевые особенности на изображении (границы, области, углы), осуществлять поиск фигур и предметов в реальном времени, осуществлять 3D реконструкцию по нескольким фотографиям и многое другое.

В сфере дополненной реальности методы компьютерного зрения применяются с целью поиска в видеопотоке специализированных маркеров. В зависимости от задачи в качестве маркера могут выступать как специально сформированные рисунки, так и лица людей. Уже после обнаружения маркера в видеопотоке и вычисления его месторасположение, возникает возможность построения матрицы проекции и позиционирования виртуальных моделей. С помощью них возможно наложить виртуальный предмет на видеопоток таким образом, что станет достигнут эффект присутствия. Главная трудность как раз и заключается в том, чтобы найти маркер, установить его местоположение в кадре и спроецировать соответствующим образом виртуальную модель.

Генетические алгоритмы

Генетические алгоритмы – это эвристические алгоритмы поиска, используются с целью решения задач оптимизации и моделирования путём случайного подбора, комбинирования и вариации искомым параметров с применением механизмов, напоминающих биологическую эволюцию.

В компьютерном зрении они применяются с целью поиска предмета некоторого установленного класса на неподвижном изображении либо видеопотоке. Сначала следует осуществить подготовку алгоритма при помощи двух различных наборов изображений:

1. “Хорошие” – включают необходимый объект.
2. “Плохие” – ошибочные изображения без желаемого объекта.

При этом для обучения применяется огромное число изображений, и чем их больше – тем лучше будет работать сам алгоритм. Для каждой картинке выполняется выделение разных основных особенностей: линии, центральные элементы, границы.

Согласно им выполняется создание статистической модели, которая затем и применяется для поиска предмета на изображении.

Примером применения данного метода может служить алгоритм обнаружения глаз и лиц на видеопотоке. Постепенно обучая алгоритм, можно достичь значительных результатов нахождения заданного класса объектов. Однако необходимость обучения как раз и создает использование генетических алгоритмов достаточно проблемным. С целью их оптимальной деятельности необходимо значительное количество разных изображений (как “хороших”, так и “плохих”), и создание классификатора для каждого предмета может занимать длительный период.

Feature detection

Теория feature detection в компьютерном зрении принадлежит к способам, которые нацелены на расчет абстракций изображения и выделения на нем основных отличительных черт. Данные характерные черты могут быть как в виде изолированных точек, так и кривых или связанных областей. Не существует жестко установленного того, что такое основная отличительная черта изображения. Любой алгоритм устанавливает под этим своё (углы, грани, области и т.п.).

Нередко с целью поиска маркеров применяются алгоритмы, которые осуществляют поиск и сопоставляют изображений по основным точкам. Основная точка – это определенный участок картинка, который является характерным для установленного изображения. Что именно берется за данную точку – непосредственно зависит от применяемого алгоритма.

С целью их нахождения и дальнейшего сопоставления применяется три составляющие:

- Детектор (англ. feature detector) – реализовывает поиск основных точек на изображении.
- Дескриптор (англ. descriptor extractor) – совершает описание обнаруженных основных точек, оценивая их позиции посредством описание окружающих областей.
- Матчер (англ. matcher) – реализовывает создание соответствий среди двух наборов точек.

Вначале при помощи детектора выполняется поиск основных точек шаблонного изображения. Полученные точки потом описываются с помощью дескриптора. Эти сведения сохраняются в отдельный файл (или базу данных), чтобы не осуществлять этот процесс снова. Для поиска меток выполняется обработка видео для каждого кадра. С целью определения соотношения среди основных точек между дескрипторами применяется матчер.

1.7 Основные методы захвата меток

В машинном зрении и робототехнике имеется класс задач: обнаружение заранее известных маркеров. Сюда возможно причислить всё: дополненная реальность, детектирование объектов по меткам, систематизация объектов в робототехнике, помощь автоматическим системам в позиционировании, трекинг объектов, задачи позиционирования объектов, QR-коды, и.т.д. [11]

По дальности работы выделяют три метода: ближней зоны, средней зоны и дальней зоны (см. Таблица 1).

Таблица 1. Классификация методов по дальности работы

| Дальность | Методы | Особенность |
|--------------|---------------------------------------|---------------------------------------|
| Ближняя зона | QR-Code, Aztec code, MaxiCode и т. д. | Возможно считывать информацию с меток |
| Средняя зона | AR-методы, SURF, SIFT, HOG и т. д. | Возможно считывать ориентацию с меток |
| Дальняя зона | Активные и пассивные метки | Можно считать только положение метки |

Ближняя зона

Ближняя зона – это область, в которой допустимо обнаружение с высоким разрешением, необходимым, для того чтобы заметить штрих-код. Методы ближней зоны дают возможность не только детектировать метку (см. Рисунок 3), но и обнаружить довольно огромное число данных с данной метки. Значительный недостаток подобных методов – требуется высокое разрешение, а следовательно, метка должна быть большой. Положительный момент – автоматическое чтение значительного числа информации, в некоторых случаях это удобно.



Рисунок 3. Метки ближней зоны

К методам ближней зоны принадлежит всё разнообразие штрих-кодов, QR-код, Aztec code, MaxiCode, и т.д.

Методы ближней зоны:

1. Выделение главных направлений преобразованием Хафа. Вначале на изображении выделяются все границы. Для изображения строится гистограмма направлений границ. Два основных максимума будут определять ориентацию маркера в пространстве. Строится преобразование Хафа в окрестности обоих максимумов. Выделяется 16 главных максимумов, по которым находится уточнённая ориентация паттерна QR-кода, а так же размер строки. После осуществляется натягивание маски QR-кода и точное считывание данных. [12] [13]
2. Поиск посредством каскада Хаара. Обученным каскадом ищется угловая метка QR-кода. После нахождения кандидата ищутся оси симметрии, которые задают ориентацию самого кода.
3. Берётся комплект испытательных прямых, для которых ищутся места, где происходит чередование черного и белого в отношении (1:1:3:1:1). Данное отношение справедливо для любого поворота метки. Места, в которых происходит выделение тестового сегмента, рассматриваются как кандидаты на месторасположение опорных символов QR-кода (см. Рисунок 4). [14]

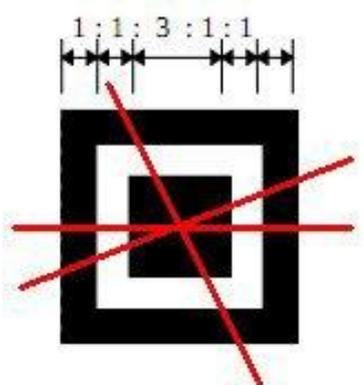


Рисунок 4. Метка QR-кода

Отмеченные три способа обладают много вариаций. Имеются и другие способы, однако все они менее распространены. Главная цель обнаружения штрих-кодов — чтение с них данных. В случае если данные с них недостаточно заметны и не считываются — от метода уже не требуется

работоспособность, даже если сам QR-код хорошо виден. По этой причине все без исключения методы, заточенные под QR-коды как правило не выделяют положение кода маленького размера, даже если его видно. [15]

Средняя зона

Средняя зона – где разрешения достаточно для пространственного разделения внешнего контура метки. Методами средней зоны считаются те методы, где нет необходимости получать данные из образца, а просто необходимо его обнаружение и поиск его ориентации. В последнее время наиболее распространенной проблемой задачи такого плана является задача дополненной реальности.

Методы средней зоны:

1. Фреймворк ARToolKit, открытый под GNU GPL. Он обнаруживает черные квадраты, в середине которых стоит 1-2 знака (см. Рисунок 5). Фреймворк детектирует связанные области тёмного цвета и осуществляет для них контурный анализ.

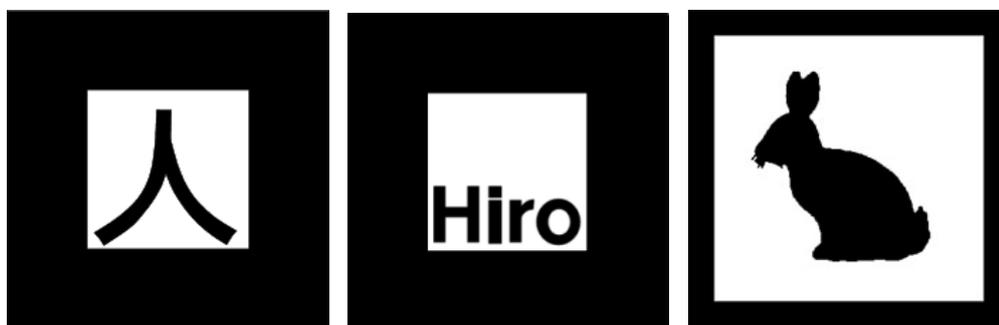


Рисунок 5. AR метки

2. Использование SURF /SIFT для поиска AR-маркера. Шаблоном считается не прямоугольник с меткой, а каждая картинка (см. Рисунок 6). [16]

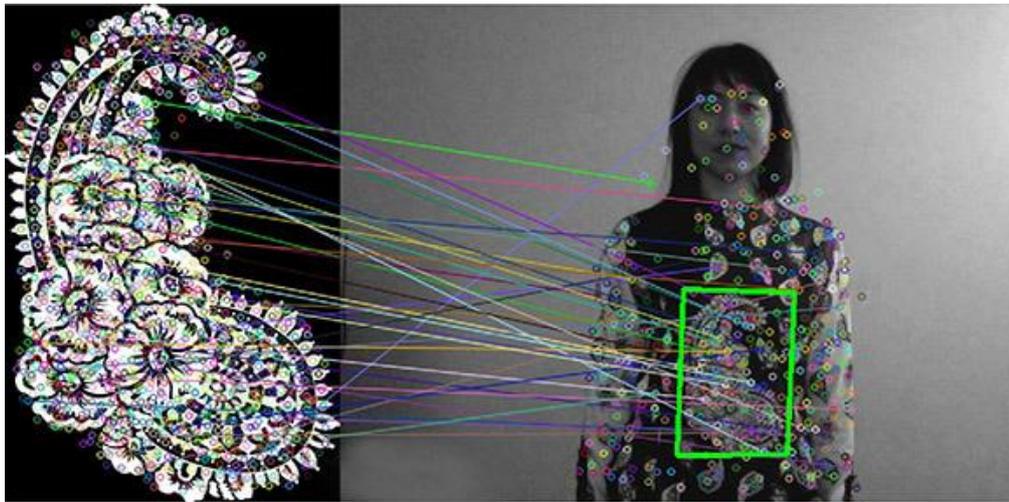


Рисунок 6. AR метка для метода SURF /SIFT

Минусом двух методов считается потребность достаточного разрешения. К тому же для обоих алгоритмов достаточно важна априорная анализ того, какого размера будет маркер на экране. Например, детекторы границ, применяемые в первом методе и настроенные на разные размеры объектов будут выдавать принципиально разные итоги (см. Рисунок 7).

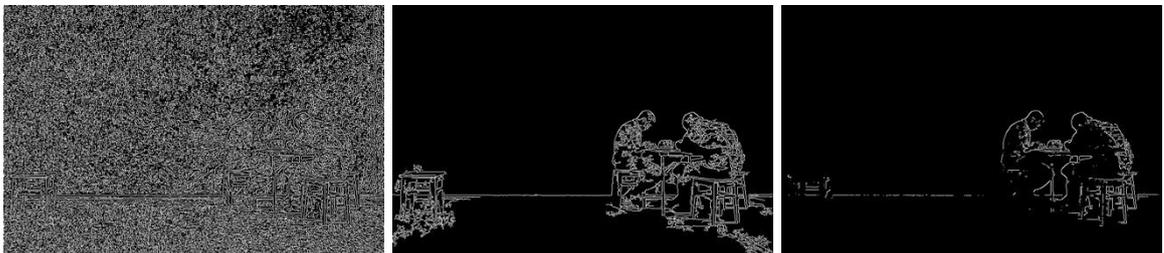


Рисунок 7. Результаты первого метода

Безусловно, настроить детектор особенностей, будь то SURF, SIFT, Haar, HOG, или что-то другое, можно на довольно небольшой картинке, но тогда работать он будет долгое время, а ориентацию восстановить будет всё равно невозможно. [17] [18] [19]

Дальняя зона

Дальняя зона – где маркер сравним с размерами пикселя. Дальнебойные способ можно применять в ближней и средней зоне.

Методы, разрешающие обнаружение в дальней зоне, менее известны, существует наименьшее число OpenSource решений. Классической задачей

для дальнейшей зоны возможно именовать motion capture (способ анимации персонажей и предметов), хотя это лишь наиболее известный из применений. Маркеры могут помогать роботам при позиционировании себя на местности, решать проблемы дополненной реальности, наблюдать за объектами, и.т.д. В каком-то смысле даже задача привязки к звёздному небу может расцениваться задачей детектирования маркеров.

1.8 Распознавание маркера дополненной реальности

Существуют два подхода для формирования дополненной реальности: с применением предварительно изготовленного маркера, который необходимо распечатать и без такого. Оба подхода распознают предметы в кадре и дополняют их, применяя алгоритм «компьютерного зрения». [20] [21]

Что может быть маркером?

В теории маркером может быть любой объект. Но на практике мы ограничены особенностями цветопередачи, освещения, вычислительной мощностью оборудования, разрешением веб-камеры(телефона). Поскольку это происходит в реальном времени, то все должно делаться стремительно, а вследствие чего выбирается, как правило, черно-белый маркер простой формы. Как правило, это четырехугольник со вписанным во внутрь идентификатором-образом.

Основные типы маркеров (см. Рисунок 8 – Рисунок 11):

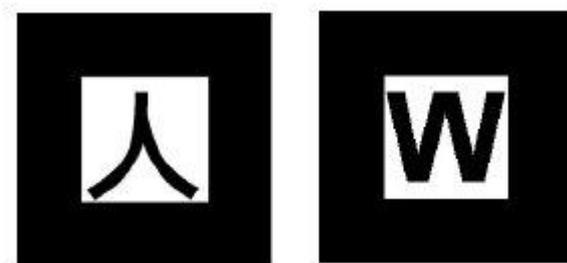


Рисунок 8. ArToolKit(АТК) marker system



Рисунок 9. Institut Graphische Datenverarbeitung (IGD) marker system

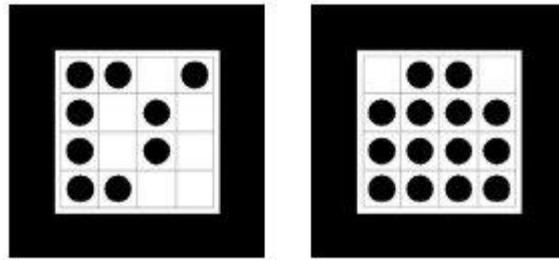


Рисунок 10. Siemens Corporate Research (SCR) marker system

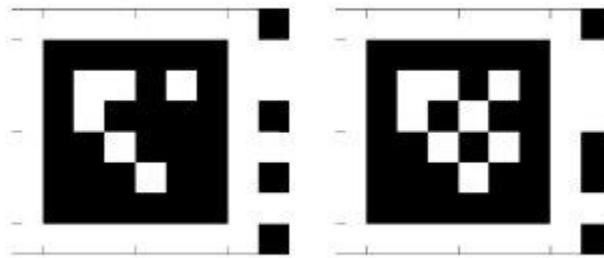


Рисунок 11. Hoffman marker system (HOM) used by SCR and Framatome ANP

Алгоритм распознавания маркера

Шаги алгоритма (см. Рисунок 12): [22]

- a) Приводим в градации серого
- b) Бинаризация изображения (порог)
- c) Определение замкнутых областей
- d) Выделяем контуры [23]
- e) Выделяем углы маркера
- f) Преобразуем координаты

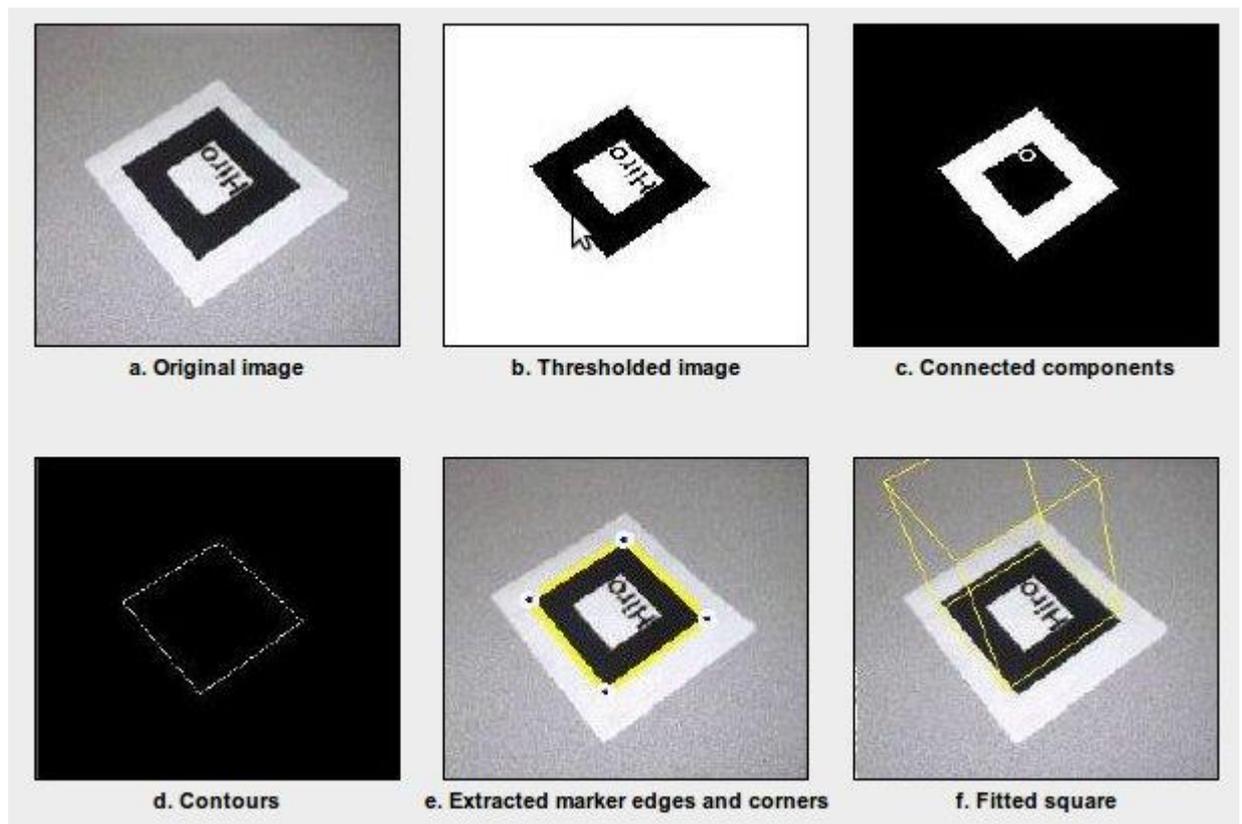


Рисунок 12. Шаги алгоритма распознавания маркера

Перевод цветного изображения в градации серого

Приведем три алгоритма перевода цветного изображения в градации серого. [24]

1. Светлота(Lightness)

$$GS = (\max(R,G,B) + \min(R,G,B))/2$$

2. Светимость(Luminosity)

$$GS = 0.21 \times R + 0.72 \times G + 0.07 \times B$$

3. Среднее(Average)

$$GS = (R + G + B) / 3$$

Вот пример, как выглядят эти три способа (см. Рисунок 13, Рисунок 14):

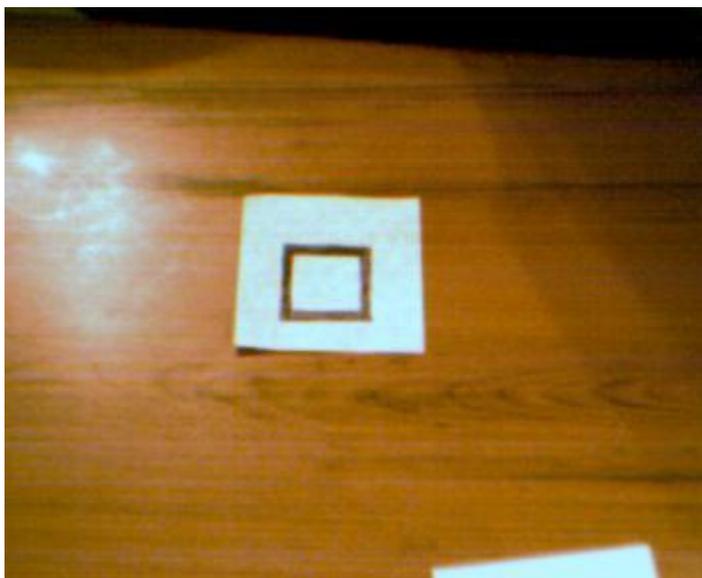


Рисунок 13. Исходное изображение

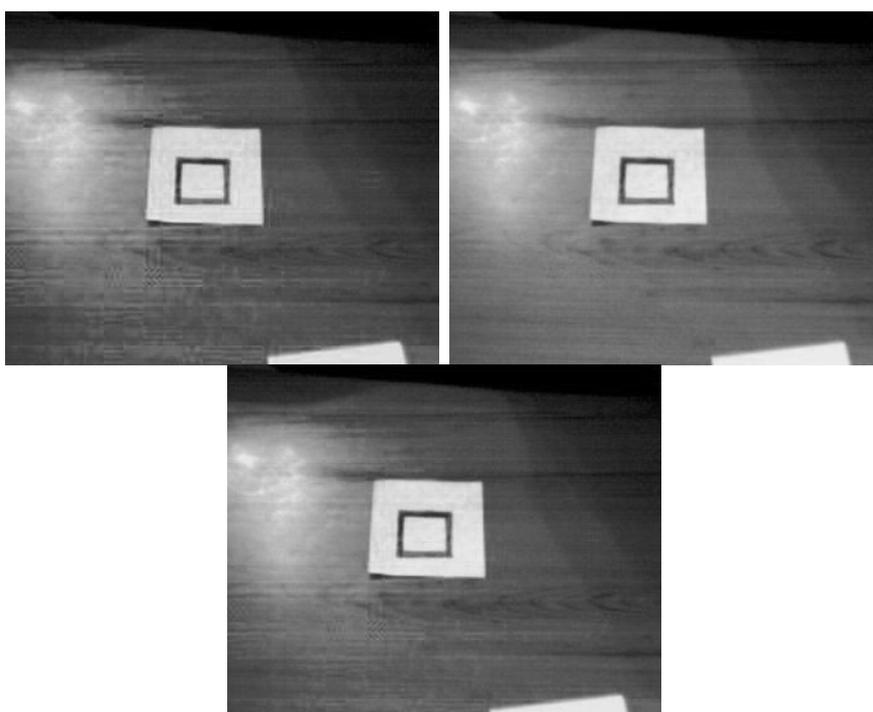


Рисунок 14. Применение алгоритмов: светлота, светимость, среднее

Бинаризация изображения. Порог

С целью перевода изображения в двухцветное состояние применяется определенный порог. Проблема в том, как и к чему данный порог использовать. Наиболее простой способ — установить порог. К примеру,

если у нас 256 цветов, то можем задать порог 128. Если немного усложнить — то можем подобрать порог воспользовавшись гистограммой цвета. Все способы обращения изображения в чёрно-белый вид можно поделить на шесть больших групп: [25]

- 1) Методы, базирующиеся на “форме“ гистограммы;
- 2) Методы на основе кластеризации;
- 3) Методы на основе исследования энтропии;
- 4) Методы базирующиеся на отыскании подобия среди серым и ч/б изображением;
- 5) Методы использующие корреляционные зависимости и особенности статистического распределения среди пикселами в областях изображения;
- 6) Методы базирующиеся на локальной адаптации порога для каждого пиксела изображения.

Больше всего применяются методы на основе локальной адаптации. Всего методов бинаризации приблизительно около сорока. Образец бинаризации изображения с установленным порогом, адаптивного, а также метода Otsu (см. Рисунок 15):

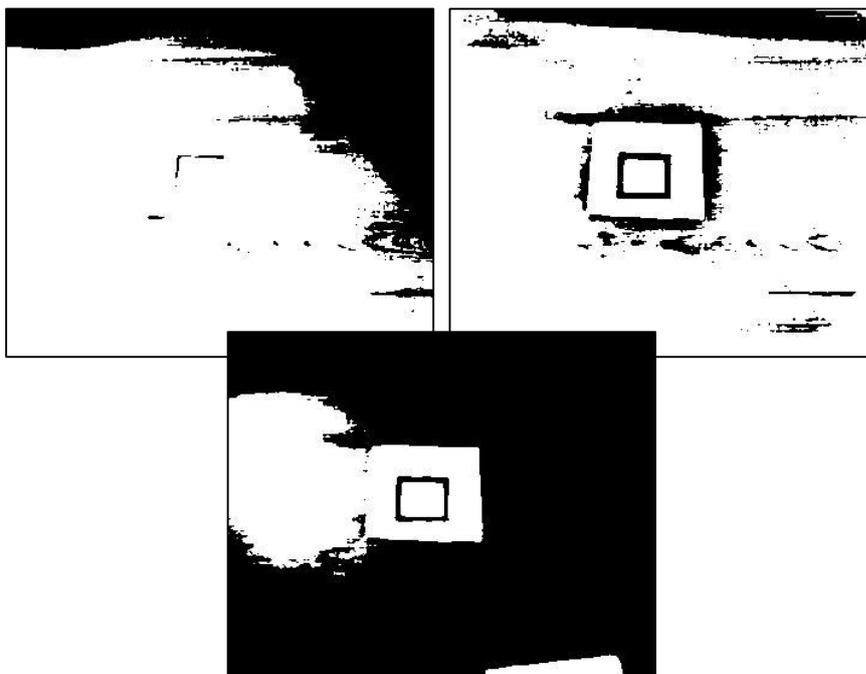


Рисунок 15. Жестко заданный порог, адаптивный, метод Otsu

Определение замкнутых областей

Следует установить замкнутые области на белом фоне. Как правило, здесь проходит сочетание алгоритмов, однако в общем случае — используют алгоритмы “заливки” белых областей и выделяют замкнутые области (см. Рисунок 16). В данном месте возможны вариации. Вначале выделяются контуры, а затем проверяются на замкнутость (см. Рисунок 17).

Выделение контуров

В настоящее время для выделения контуров используют около 6 основных методов: [26] [27] [28] [29]

- 1) Marr-Hildreth Edge Detector;
- 2) Canny Edge Detector;
- 3) Boolean function based Edge Detector;
- 4) Euclidian distance and Vector Angle based Edge Detector;
- 5) Depth Edge Detection using Multi-Flash Imaging;
- 6) Sobel Edge Detector.

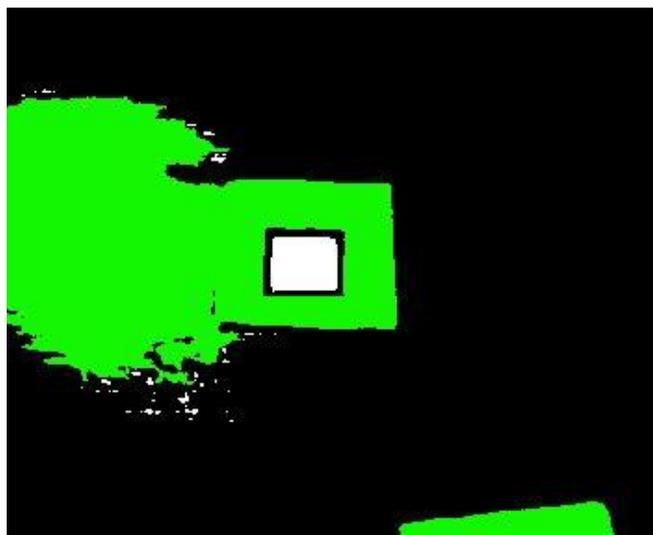


Рисунок 16. Заливка белых областей и выделение замкнутых областей

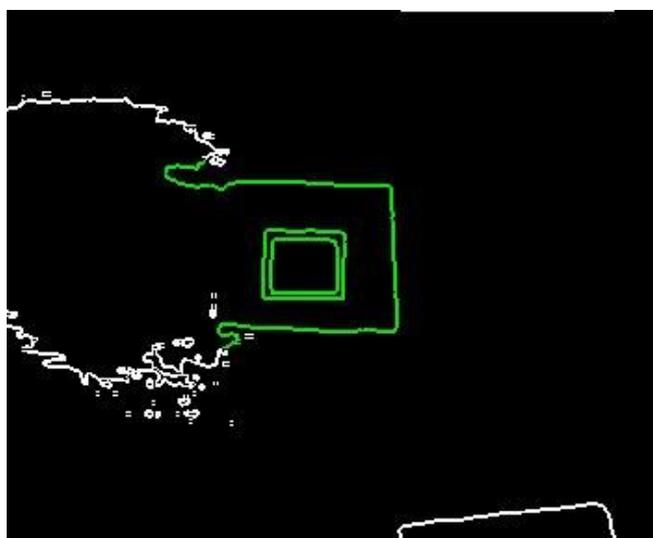


Рисунок 17. Выделили контуры, потом проверили на замкнутость

Выделение углов маркера

Выделив контур, его необходимо сопоставить с маркером. Может быть выделено много разных контуров на изображении, необходимо найти что-то «похожее» на четырехугольник.

Для решения этой задачи можно использовать алгоритм Дугласа-Пекера, позволяющий сократить количество точек кривой, аппроксимированной большей серией точек. [30]

Например, в `opencv` есть функция `approxPolyDP`, которая уже это делает. Если ее результат подвергнуть обработке, в таком случае удастся

вполне оптимальный итог (см. Рисунок 18). Таким образом получаются координаты углов маркера.

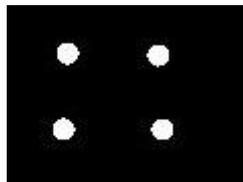


Рисунок 18. Выделенные углы маркера

Преобразование координат

Уже существуют координаты углов маркера, которые по сути, в идеале являются перпендикулярными, а в действительности размещены под иным углом. Помимо этого, и в идеале и в реальности, края квадрата являются осями координат. Таким образом, мы можем установить положение «камеры» относительно нашего предмета, и точку отсчета начала координат. На чем основан метод установления координат можно продемонстрировать схематически (см. Рисунок 19):

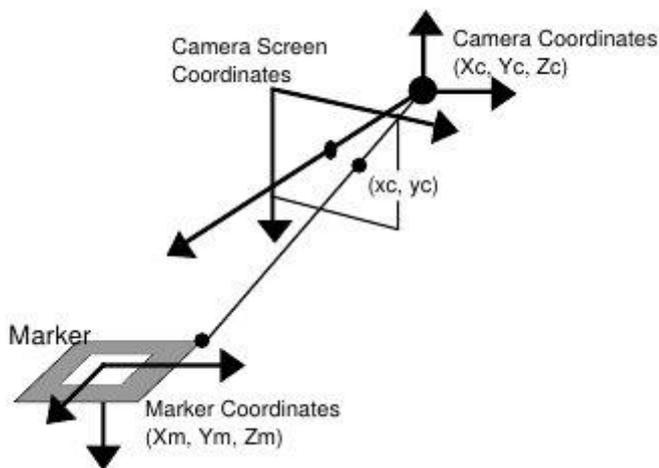


Рисунок 19. Основа метода определения координат

Т.е. Концепция состоит в том, что при изменении угла, с которого смотрит видеокамера, меняется размер проекции (см. Рисунок 20).

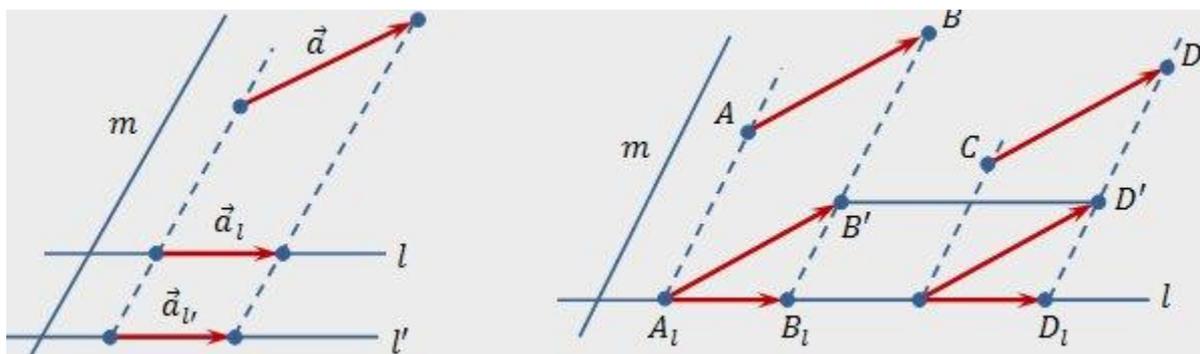


Рисунок 20. Изменения угла просмотра камерой

Зная положение камеры и точки отсчета, можно нарисовать проекцию для 3D модели. В этом случае куба. Как видно, если используется прямоугольник с чистым полем в качестве маркера, то он симметричен, и определить вращение можно только лишь частично. Иногда этого достаточно. Если же нет — то необходимо внести внутрь квадрата дополнительный маркер, он тогда будет выглядеть, например так (см. Рисунок 21) и получить в результате еще и угол поворота, применяя преобразование Хафа. [31] [32]



Рисунок 21. Дополнительный маркер внутри квадрата

1.9 Построение дополненной реальности

Для решения установленной задачи мы применяем описанную выше библиотеку OpenCV. С помощью неё будет делаться поиск маркера на видеопотоке с дальнейшим отображением виртуального предмета.

Для построения 3D пространства нужно знать 2 матрицы: внутреннюю и внешнюю. Затем, воспользовавшись методами OpenGL, можно без каких-либо проблем нарисовать виртуальный объект поверх маркера. [33]

Внутренняя матрица (проекции) состоит из параметров: фокального расстояния по двум осям (f_x, f_y) и координат центра фокуса (c_x, c_y).

$$\begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

Процесс нахождения характеристик камеры называется ее калибрацией. Для его исполнения следует реализовать снимки калибровочного шаблона с помощью используемой камеры, перенести фотографии на компьютер и осуществить подсчет параметров матрицы. С целью получения как можно более точных параметров необходимо соблюдение следующих условий:

- Шаблон шахматной доски распечатывается на чистом листе формата А4.
- Лист шаблона обязан находиться ровно на поверхности, края не должны быть завернуты, предпочтительно отсутствие каких-либо изгибов.
- Размер фотоснимков с камеры должен быть приведен к размеру кадра видеопотока. В отдельных случаях возможна ситуация, когда разрешение фотографий выше разрешения видео. По этой причине перед калибровкой следует уменьшить снимки до необходимого размера.
- Количество снимков шаблона должно быть не меньше 10. Осуществить их необходимо с различных ракурсов ракурсов. Чем больше будет снимков, тем более точные будут получены параметры матрицы проекции.

Внешняя матрица позволяет однозначно задать положение объекта в пространстве.

$$\begin{bmatrix} r_1 1 & r_1 2 & r_1 3 & t_1 \\ r_2 1 & r_2 2 & r_2 3 & t_2 \\ r_3 1 & r_3 2 & r_3 3 & t_3 \end{bmatrix}$$

Диагональные элементы отвечают за растяжение модели. Остальные элементы r – за поворот объекта в пространстве. Элементы t отвечают за перенос.

Расчет её выполняется путём установления четырёх пар точек предмета и соответствующего его положения в кадре. Точки положения в кадре – это вершины четырехугольника, описывающего предмет в кадре.

Сравнительно точек самого предмета нужно выделить несколько моментов:

- Точки предмета задаются в 3D, а точки на кадре в 2D. В соответствии с этим, если задать им ненулевое значение z , то начало координат на z будет сдвинуто относительно плоскости объекта на кадре.
- Точки предмета задаются таким способом: начало координат находится прямо по центру шаблона, а единица длины равняется половине меньшей стороны.

Построение матрицы потом применяются для расчета вектора поворота и переноса с дальнейшим занесением в матрицу модели. Этот процесс не сильно влияет на скорость работы программы. Главная нагрузка проходит в сфере анализа видеопотока и поиска на нём маркера.

Представленный выше подход можно использовать к разным библиотекам компьютерного зрения в плане AR. Ввиду необходимости выполнения калибровки камеры с целью построения правильной матрицы проецирования подход выглядит трудным. Одним из методов решения данной проблемы становится формирование набора характеристик различных камер.

Формируя это направление, можно вполне создать свой собственный фреймворк AR. Разрабатывая разные методы фильтрации точек, возможно

достичь результата обнаружения маркера даже при достаточно сильном угле наклона и при сильном удалении. Это позволит приобрести более реальную картинку и предоставить возможность пользователям проанализировать модель со всех сторон на разном расстоянии.

1.10 Будущее Augmented Reality

Имеются разные взгляды на то, какими станут в завтрашнем дне AR системы.

Дополненная реальность в настоящее время используется в медицине, навигации, военном деле, архитектуре, археологии, биологии, переводах, офисной работе и других типах человеческой деятельности. И развитие не стоит на месте. Возникает новое поколение аппаратных устройств, подобные smart очкам, бесконтактным сенсорным контроллерам. Возникают новые прототипы интерфейсов пользователей для AR систем. Для разработчиков издаются новые версии библиотек.

С формированием технологий дополненная реальность всё основательнее станет просачиваться в нашу ежедневную жизнь. В настоящее время формируется и накапливается все большее и большее число данных, и дополненная реальность станет мостом для доступа к огромному количеству данных. Причем AR будет интегрирована настолько глубоко, что станет восприниматься подсознательно. И непосредственно термин "Дополненная реальность", возможно, прекратит применяться, а цифровые данные станут расцениваться составляющей самой реальности. Главное в таком случае, то что дополненная реальность дает возможность перенести нас на новую степень взаимодействия с цифровым миром без необходимости отказываться от привычного окружения, и полностью погружаться в виртуальность. [34]

1.11 Выводы

С каждым годом приложения с использованием дополненной реальности набирают все большую популярность. Она используется в различных сферах деятельности: производство, ремонт, обучение, продажи,

маркетинг, выставки, руководства пользователя, удаленное обслуживание и навигация.

Для создания функционирующей системы необходима достаточно мощная платформа, которой может быть современное мобильное устройство, в виду их широкой распространенности и постоянно растущих возможностей.

Для отрисовки виртуальных объектов используют маркеры, которые расположены в окружающем пространстве. Они находятся и анализируются специальным программным обеспечением.

ГЛАВА 2 ПОСТАНОВКА ЗАДАЧИ

2.1 Название

Браузер дополненной реальности

2.2 Назначение

Разрабатываемая система предназначена для воспроизведения дополненной реальности в учебных материалах (на примере компании АО «Транснефть - Сибирь»). Система предоставляет возможность показать устройство сложного оборудования без его реального присутствия, его ремонт, так же разные учебные видео материалы в печатной продукции.

2.3 Цель

Повысить качество визуального восприятия графических объектов в учебном материале средствами нейромаркетинга. Увеличить содержательность учебных материалов. Сократить временные затраты на ознакомление с макетами оборудования.

2.4 Задачи

Для достижения поставленных целей необходимо решить следующие задачи:

1. Ознакомиться с форматом интерактивной подачи материала из области нейромаркетинга
2. Выбрать механизм визуализации дополнительного контента в учебном материале
3. Разработать инструмент, позволяющий провести анализ графического объекта и воспроизвести необходимый контент
4. Провести апробацию разработанного приложения

2.5 Список разрабатываемых алгоритмов

1. Алгоритм распознавания (обнаружения) маркера:

- a. Приводим в градации серого
 - b. Бинаризация изображения (порог)
 - c. Определение замкнутых областей
 - d. Выделяем контуры
 - e. Выделяем углы маркера
 - f. Преобразуем координаты
2. Захват маркера
 3. Построение 3D пространства
 4. Отображение виртуального объекта

2.6 Функциональные особенности разрабатываемой системы

1. Создание нового учебника
2. Добавление материала в учебнике
3. Загрузка изображения
4. Создание маркера
5. Добавление к маркеру содержимого: видео, 3D модель оборудования и т. д.
6. Распознавание маркера
7. Воспроизведение содержимого маркера
8. Просмотр содержимого учебника
9. Поиск материала по учебнику
10. Просмотр списка всех учебников

2.7 Выбор платформы реализации

Для разработки были выбраны следующие программные продукты:

- Microsoft Visio 2010
- StarUML
- Microsoft Visual Studio 2013
- язык программирования C++
- СУБД MySQL

- Библиотека OpenCV
- Библиотека OpenGL

Microsoft Visio — векторный графический редактор, редактор диаграмм и блок-схем для Windows.

StarUML — это средство для UML-моделирования, с открытым исходным кодом. Поддерживает стандарт UML 2.0, существует поддержка ER-диаграмм. Распространяется бесплатно для 32-разрядных систем Windows. [35]

Microsoft Visual Studio — продукт компании Microsoft, включает среду для разработки ПО и ряда других инструментальных средств. Он также содержит в себе редактор исходного кода и возможность простого рефакторинга кода. Интегрированный отладчик работает как отладчик исходного кода, так и как отладчик машинного уровня. [36]

C++ — компилируемый, статически типизированный язык программирования общего назначения. Язык программирования C++ широко применяется с целью разработки ПО, представляя собой одним из наиболее известных языков программирования. Поддерживает объектно-ориентированное программирование.

MySQL — свободная реляционная система управления базами данных. Oracle осуществляет поддержку и разработку. Имеется API для большинства языков программирования, в том числе C++.

2.8 Анализ существующих решений

Перед началом разработки необходимо произвести выбор Фреймворка (набора библиотек AR систем).

За минувшее десятилетие различными компаниями и научными сообществами было реализовано ряд библиотек компьютерного зрения. Так как проблема производительности считается довольно острым в данной дисциплине, то большая часть написаны на языке C++.

Кроме хорошей скорости выполнения это также даёт возможность их использования на различных платформах.

Библиотека OpenCV

OpenCV — фреймворк алгоритмов компьютерного зрения с открытым исходным кодом. Сделан на языке C++, кроме того поддерживаются и другие языки. Распространяется в условиях лицензии BSD.

Сегодня OpenCV считается, скорее всего, наиболее известной и сформированной библиотекой. В её структуру входят модули:

- обработки изображений;
- построения несложных пользовательских интерфейсов, загрузки/сохранения видео и аудиоданных;
- анализа движения и отслеживания предметов (оптический поток, шаблоны движения, устранение фона);
- калибровки камеры, поиска стереосоответствий и элементов обработки трехмерных данных;
- поиска, анализа и сопоставления основных точек изображений (Feature Detection and Description);
- обнаружения предметов на изображении (вейвлеты Хаара, HOG и т. д.); [37]
- методы и модели машинного обучения (SVM, деревья принятия решений и т.д.).

Перечень аналогичных библиотек довольно широк, и большое количество их созданы на базе OpenCV.

Достоинствами данной системы является: открытый исходный код, распространение по лицензии BSD, модуль обработки изображения, поддержка платформ android и iOS. [38] [39]

Библиотека ARToolkit

ARToolKit считается свободно распространяемой библиотекой с открытым исходным кодом на языке C с целью формирования AR приложения. Положение и ориентация камеры относительно маркера-карточки рассчитывается на основе способов компьютерного зрения, что дает возможность программисту накладывать виртуальные предметы поверх маркеров. [40]

Библиотека Swarp SDK

Отечественная компания Sectar изготовила собственный продукт Swarp SDK для создания AR приложений на платформе .NET. Данный инструмент применяется в приложениях с AR технологией. У пользователя, использующего инструмент, пропадает необходимость разрабатывать ту часть программы, которая отвечает за обнаружение и передвижение маркера, вывод конкретной сцены и т. д.

В Swarp SDK применяется особая система маркеров. Маркер представлен в виде XML-файла с расширением *.trackable*. Стандартный маркер, содержимое которого приведено ниже (см. Рисунок 22).

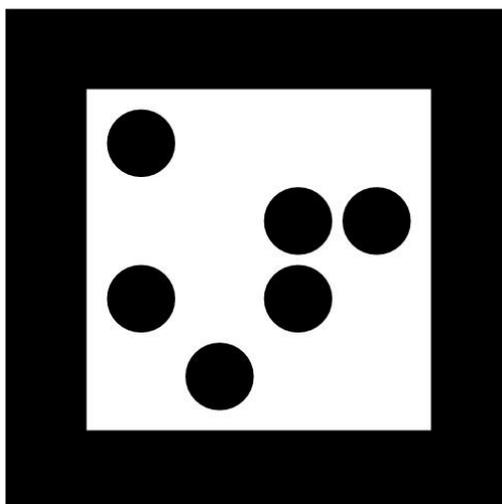


Рисунок 22. Маркер Swarp

Данный SDK – набор COM библиотек, которые исполняют все алгоритмы и процедуры требуемые для технологии AR систем. Так же в

набор входят библиотеки, которые отвечают за работу камеры и получения видео - потока с нее.

Библиотека Metaio

Metaio — немецкая компания, разработчик и первооткрыватель в сфере AR систем и соответствующего ПО. [41]

Продукты компании: metaio SDK, metaio Creator, metaio CVS, metaio Cloud, junaio, metaio Engineer. [42]

Библиотека Augasma

Augasma — это фреймворк для реализации AR систем, разработанный крупнейшим Британским разработчиком Autonomy.

Библиотека использует технологии распознавания изображений, может использовать камеры у планшета или смартфона. Она распознает объекты окружающего нас мира, а затем накладывает поверх них видео, 3d модели, и т.п.

Библиотека Vuforia

Vuforia — это платформа AR систем и инструментарий разработчика ПО для AR систем (Software Development Kit — SDK) для мобильных устройств, разработанные компании Qualcomm. Vuforia применяет технологии компьютерного зрения. С версии 2.5 Vuforia обнаруживает текст, а с 2.6 — обладает возможность различать цилиндрические метки. [43]

Регистрации изображений дает возможность разработчикам располагать и направлять виртуальные предметы, такие как 3D-модели, в связке с действительными образами при просмотре посредством камеры различных устройств. Виртуальный объект отображается на реальном образе так, что достигается эффект — чувство того, что этот объект является составляющей настоящего мира.

Vuforia поддерживает разные 2D- и 3D-типы мишеней, в том числе безмаркерные Image Target, трёхмерные мишени Multi-Target, кроме того

реперные метки, выделяющие в сцене объекты с целью их определения. Вспомогательные функции содержат выявление препятствий с применением так называемых «Виртуальных кнопок» («Virtual Buttons»), детектирование целей и возможность программно формировать и реконфигурировать цели в рамках самомодифицирующегося кода.

Vuforia дает интерфейсы программирования приложений на языках C++, Java, Objective-C, и .Net посредством интеграции с игровым движком Unity. Подобным способом SDK поддерживает разработку нативных приложений дополненной реальности для iOS и Android, в то же время предполагая разработку в Unity, итоги которой могут быть легко перенесены на обе платформы. Приложения AR систем, созданные на платформе Vuforia, совместимы с широким диапазоном устройств, в том числе iPhone, iPad, смартфоны и планшеты на Android с версии 2.2 и процессором, начиная с архитектур ARMv6 или 7 с перспективой выполнения вычислений с плавающей запятой. [44]

Библиотека AForge.NET

AForge.NET считается C# фреймворком с открытым исходным кодом, сформированным для разработчиков и исследователей в сфере искусственного интеллекта и компьютерного зрения. Главным идеологом и разработчиком данного проекта считается один человек (Андрей Кириллов), однако на протяжении всего периода свой вклад вносили разработчики из различных государств мира.

На базе AForge.NET была изобретена библиотека Grاتف, применяемая с целью построения AR систем. Написана на языке C# и обладает хорошей переносимостью на различные платформы.

2.9 Выводы

Таблица 2. Сводная таблица характеристик библиотек

| Критерии Альтернативы | Кроссплатформенность | Наличие документации и литературы | Поддержка различных языков программирования | Широкий выбор дополнительных сервисов (оценка) | Тип лицензии |
|--------------------------|--------------------------|-----------------------------------|---|--|------------------------------|
| OpenCV | iOS, PC, Android, Linux | Много | C/C++, Python, Java, Ruby, Matlab, Lua | Отлично | Boost Software License |
| Vuforia | iOS, Android, Unity | Достаточно | C++, Java, Objective-C, .Net | Отлично | Free + Commercial SDK option |
| ARToolKit | iOS, Android, Unity | Много | Java, Objective-C | Мало | Free + Commercial SDK option |
| Metaio SDK | iOS, Android, Unity, WEB | Мало | Java, Objective-C, .Net | Отлично | Free + Commercial SDK option |
| AForge.NET | WEB, iOS, Android | Мало | .Net | Отлично | Free + Commercial SDK option |
| Aurasma | iOs, Android | Мало | | Мало | Free + Commercial SDK option |
| Swarp SDK | iOS, Android | Достаточно | .Net | Мало | |

Обзор альтернатив с учетом наиболее важных критериев показал (см. Таблица 2), что фреймворки Vuforia, OpenCV, Swarp SDK подходят для разработки системы, однако требования безопасности не позволяют использовать сторонние библиотеки.

Для разработки были выбраны следующие программные продукты: язык программирования C++, база данных SQLite.

ГЛАВА 3 ПРОЕКТИРОВАНИЕ БРАУЗЕРА ДОПОЛНЕННОЙ РЕАЛЬНОСТИ

3.1 Построение функциональной модели

Взаимодействие системы и пользователя показано на диаграмме вариантов использования (см. Рисунок 23). [45]

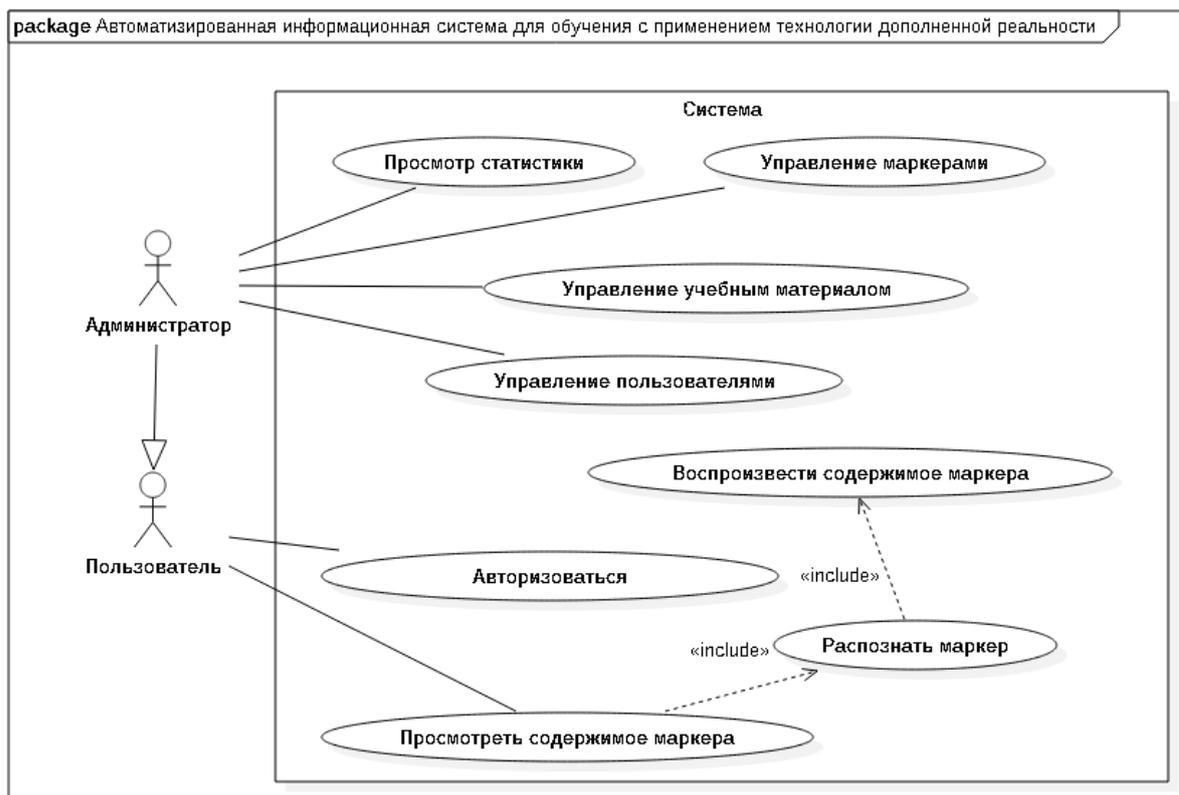


Рисунок 23. Диаграмма вариантов использования

Разработка диаграммы преследует следующие цели:

- сформулировать единые требования к работе проектируемой системы;
- спроектировать начальную концептуальную модель системы с целью ее дальнейшей детализации в форме логических и физических моделей.

Диаграмма вариантов использования дополнена спецификацией, которая представлена в приложении (ПРИЛОЖЕНИ).

3.2 Построение моделей данных

Построение концептуальной модели данных

Для описания концептуальной схемы системы дополненной реальности используется ER диаграмма. Она показывает данные и отношения между ними, обеспечивает детализацию хранилищ данных проектируемой системы, включая идентификацию объектов (сущностей), свойств этих объектов (атрибутов) и их отношения с другими объектами (связи).

ER-диаграммы выполнены в нотации Чена. Для построения диаграммы необходимо выделить сущности, которые встречаются в предметной области.

Пользователем системы может быть только зарегистрированный пользователь. Все пользователи системы имеют разные права. Администратор может создавать пользователей, назначать им права. Он также заполняет учебный материал, создает маркеры дополненной реальности. Студент – обучающийся имеет возможность просматривать учебный материал, просматривает дополненную реальность. В зависимости от назначенных прав пользователь может делать те или иные действия в системе. На пользователя могут назначить несколько прав (см. Рисунок 24).

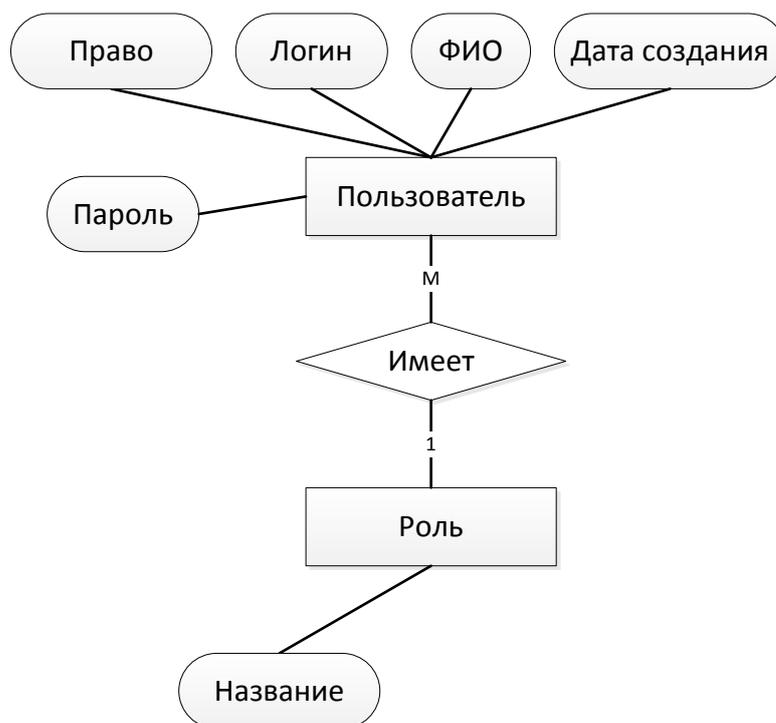
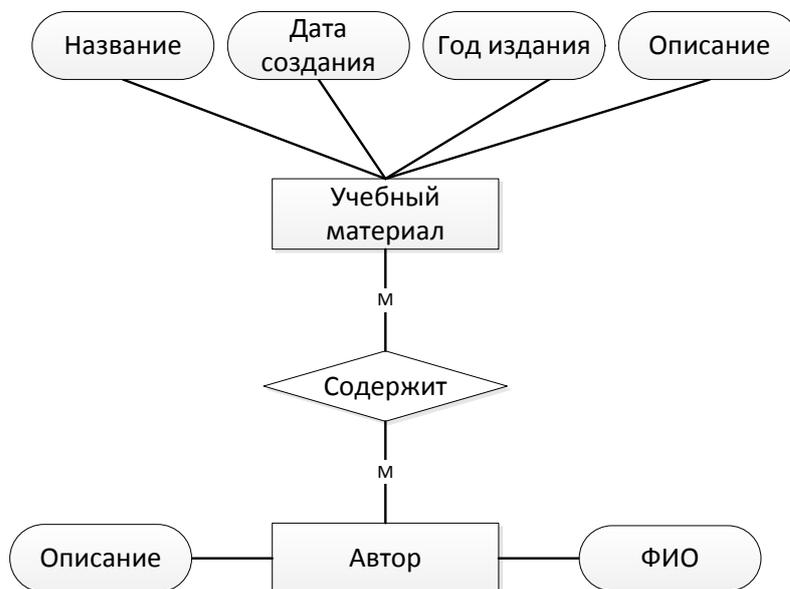


Рисунок 24. Хранилище данных «Пользователи»

Учебный материал может связан с несколькими авторами. (см. 0).



Хранилище данных «Учебники»

Для реализации дополненной реальности необходимо создавать и хранить маркеры дополненной реальности (см. Рисунок 25).

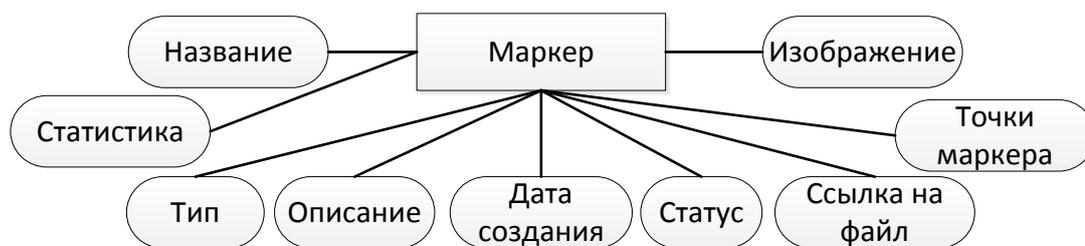


Рисунок 25. Хранилище данных «Маркеры»

На рисунке (см. Рисунок 26) изображена глобальная ER-диаграмма, на которой отображены все хранилища данных, и показаны связи между ними. В таблице ниже (Таблица 3) показаны типы сущности и их краткое описание.

Таблица 3. Описание сущностей

| Имя типа сущности | Краткое описание | Разновидность |
|--------------------------|-------------------------------|----------------------|
| Пользователь | Пользователи системы | сильный |
| Право | Права пользователей | сильный |
| Автор | Авторы учебных материалов | сильный |
| Учебный материал | Учебный материал в учебниках | сильный |
| Маркер | Маркер дополненной реальности | сильный |

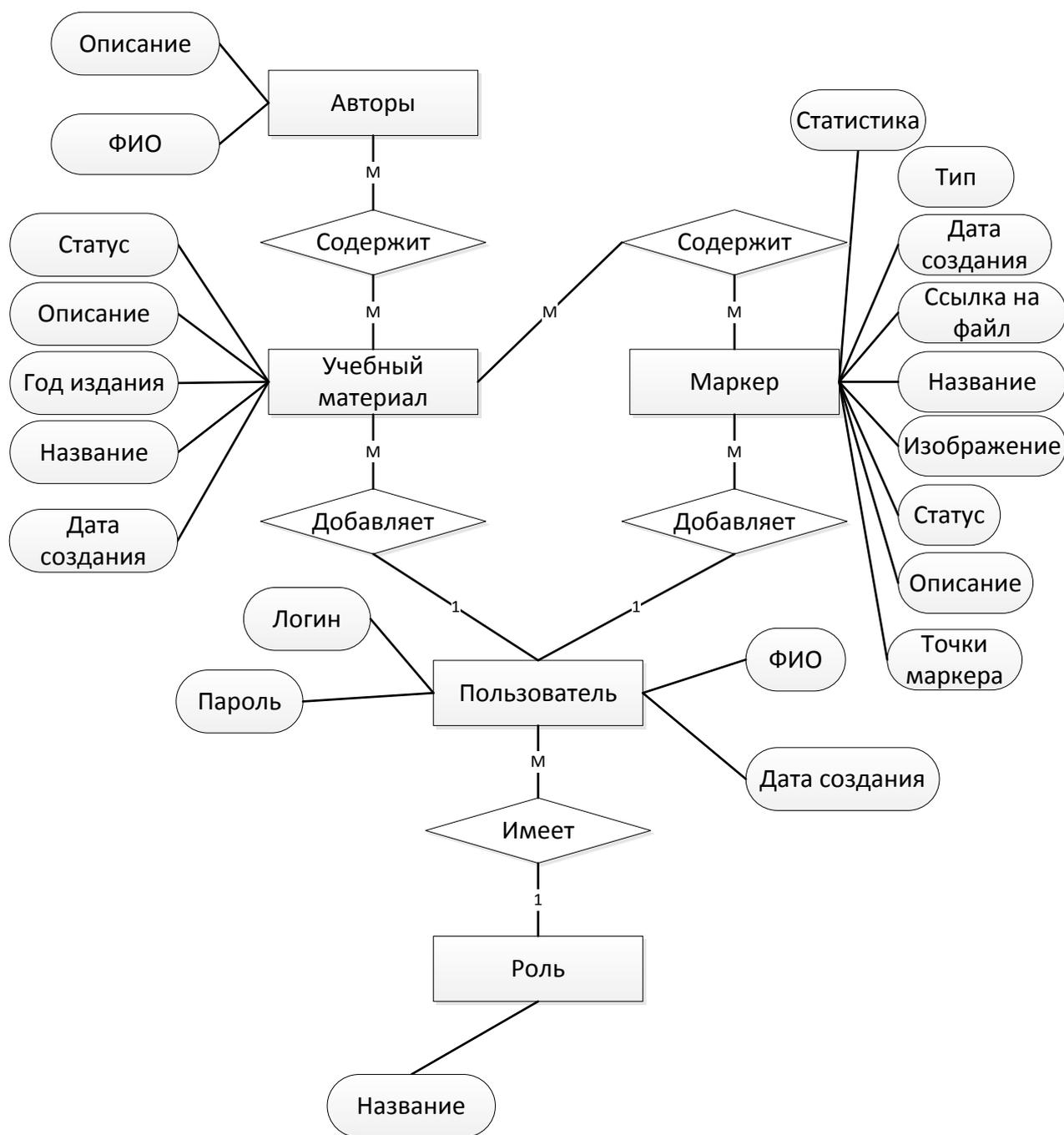


Рисунок 26. Глобальная ER диаграмма

В таблице (см. Таблица 4) показана подробная информация, характеризующая связи между сущностями.

Таблица 4. Описание типов связей

| Имя типа связи | Краткое описание | Степень связи | Список типов сущностей, участвующих в связи | |
|----------------|---|---------------|---|---|
| | | | | |
| Содержит | Связывает Учебный материал и Авторов | бинарная | Авторы | М |
| | | | Учебный материал | М |
| Содержит | Связывает Учебный материал с Маркером | бинарная | Учебный материал | М |
| | | | Маркер | М |
| Добавляет | Связывает Пользователя с Учебным материалом | бинарная | Пользователь | 1 |
| | | | Учебный материал | М |
| Добавляет | Связывает Пользователя с Маркером | бинарная | Пользователь | 1 |
| | | | Маркер | М |
| Имеет | Связывает Пользователя с Правами | бинарная | Пользователь | М |
| | | | Право | М |

В таблицах 5-9 описаны атрибуты сущностей.

Таблица 5. Сущность «Пользователь»

| Имя атрибута | Домен | Тип по составу | Тип по значению |
|---------------|-----------|----------------|-----------------|
| Дата создания | Текстовый | простой | однозначный |
| ФИО | Текстовый | простой | однозначный |
| Логин | Текстовый | простой | однозначный |
| Пароль | Текстовый | простой | однозначный |

Таблица 6. Сущность «Роль»

| Имя атрибута | Домен | Тип по составу | Тип по значению |
|--------------|-----------|----------------|-----------------|
| Название | Текстовый | простой | однозначный |

Таблица 7. Сущность «Маркер»

| Имя атрибута | Домен | Тип по составу | Тип по значению |
|---------------|-----------|----------------|-----------------|
| Название | Текстовый | простой | однозначный |
| Тип | Булевой | простой | однозначный |
| Описание | Текстовый | простой | однозначный |
| Дата создания | Дата | простой | однозначный |

| Имя атрибута | Домен | Тип по составу | Тип по значению |
|---------------|------------------------|----------------|-----------------|
| Статус | Перечисляемый (статус) | простой | однозначный |
| Файл | Файл | простой | однозначный |
| Изображение | Файл | простой | однозначный |
| Точки маркера | Файл | простой | однозначный |
| Статистика | Числовой | простой | однозначный |

Таблица 8. Сущность «Учебный материал»

| Имя атрибута | Домен | Тип по составу | Тип по значению |
|---------------|------------------------|----------------|-----------------|
| Название | Текстовый | простой | однозначный |
| Год издания | Числовой | простой | однозначный |
| Дата создания | Дата | простой | однозначный |
| Статус | Перечисляемый (статус) | простой | однозначный |
| Описание | Текстовый | простой | однозначный |

Таблица 9. Сущность «Авторы»

| Имя атрибута | Домен | Тип по составу | Тип по значению |
|--------------|-----------|----------------|-----------------|
| ФИО | Текстовый | простой | однозначный |
| Описание | Текстовый | простой | однозначный |

В таблице ниже (см. Таблица 10) описаны домены атрибутов

Таблица 10. Домены

| Наименование | Описание | Диапазон значений |
|------------------------|---------------------------------------|--|
| Текстовый | Строка, состоящая из символов | [А-Я]+{[а-я], [А-Z], [а-z], [0-9], [Символы]} |
| Дата | Формат даты ДД.ММ.ГГГГ ЧЧ:ММ:СС | ДД – [01..31] ММ - [01..12] ГГГГ - [1900..+∞] ЧЧ - [00..23] ММ - [00..59] СС - [00..59] |
| Числовой | Целое число | {[0-9]} |
| Перечисляемый (статус) | Перечисляемы тип | («Активный», «Не активный») |
| Булевой (Тип) | True - «Видео», False- | (True, False) |

| | | |
|------|------------------------|--|
| | «3D модель» | |
| Файл | Массив двоичные данные | |

Построение логической модели данных

На рисунке (см. Рисунок 27) отображена логическая модель данных, на которой отображены сущности, их атрибуты и связи между ними. Диаграмма выполнена в методологии IDEF1X. Методология основана на подходе Чена и позволяет построить модель данных, эквивалентную реляционной модели в третьей нормальной форме.

Таблица «Пользователи» содержит информацию о пользователях системы, данная таблица связана с таблицей «Права» связью один ко многим. Одному пользователю можно назначить только одно право, это сделано, чтобы разграничить права пользователей в систему: есть пользователи, которые могут добавлять других пользователей, управлять маркерами, учебным материалом, а есть пользователи, которые могут только пользоваться браузером дополненной реальности. Таблица «Маркеры» содержит информацию о маркерах: дескрипторы, название маркера, изображение маркера, содержание (что будет показано при обнаружении маркера). «Учебный материал», содержит в себе много маркеров и связан с таблицей «Учебники» связью много ко многим.

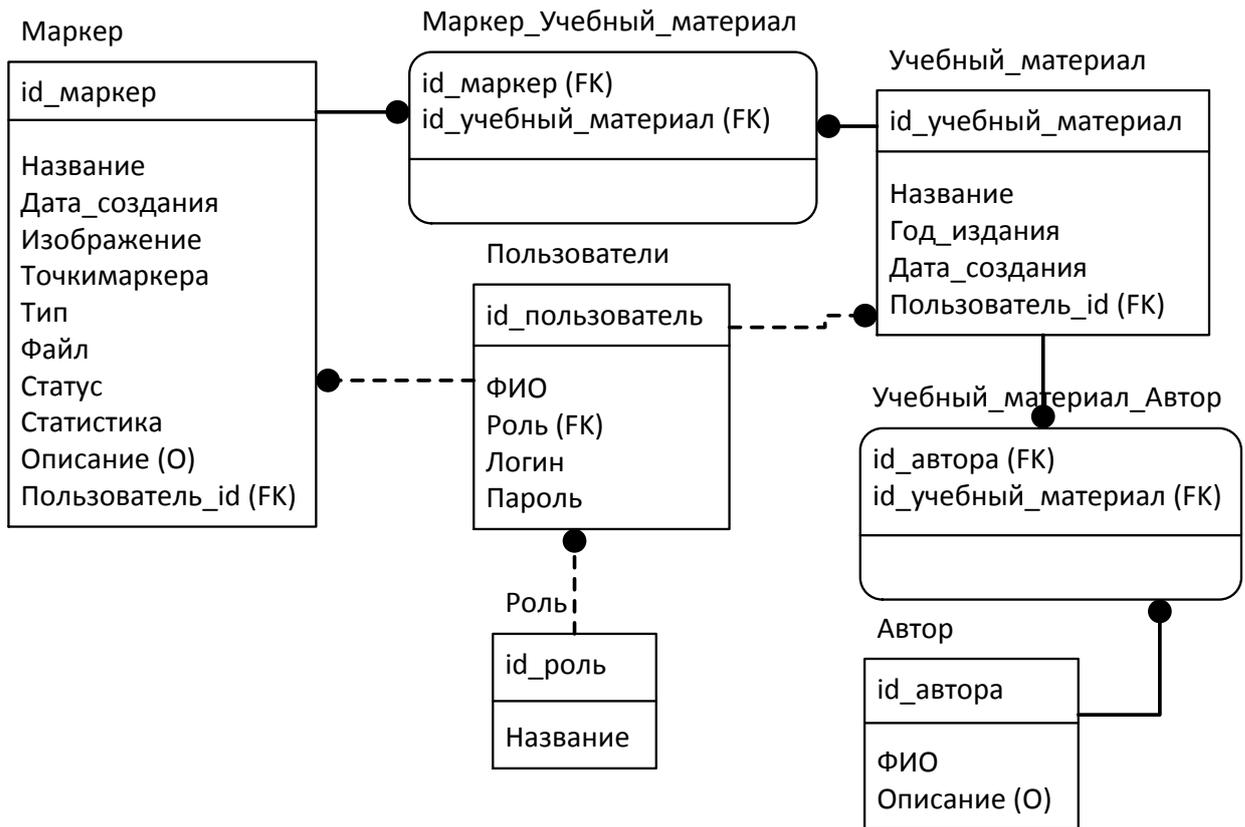


Рисунок 27. Логическая модель данных (IDEF1x)

Построение физической модели данных

Физическая модель данных определяет способ размещения данных в среде хранения и способы доступа к этим данным, которые поддерживаются на физическом уровне. Она включает всю информацию, требуемую с целью реализации конкретной базы данных. На рисунке (см. Рисунок 28) изображена физическая модель данных. В качестве конкретной БД была выбрана SQLite.

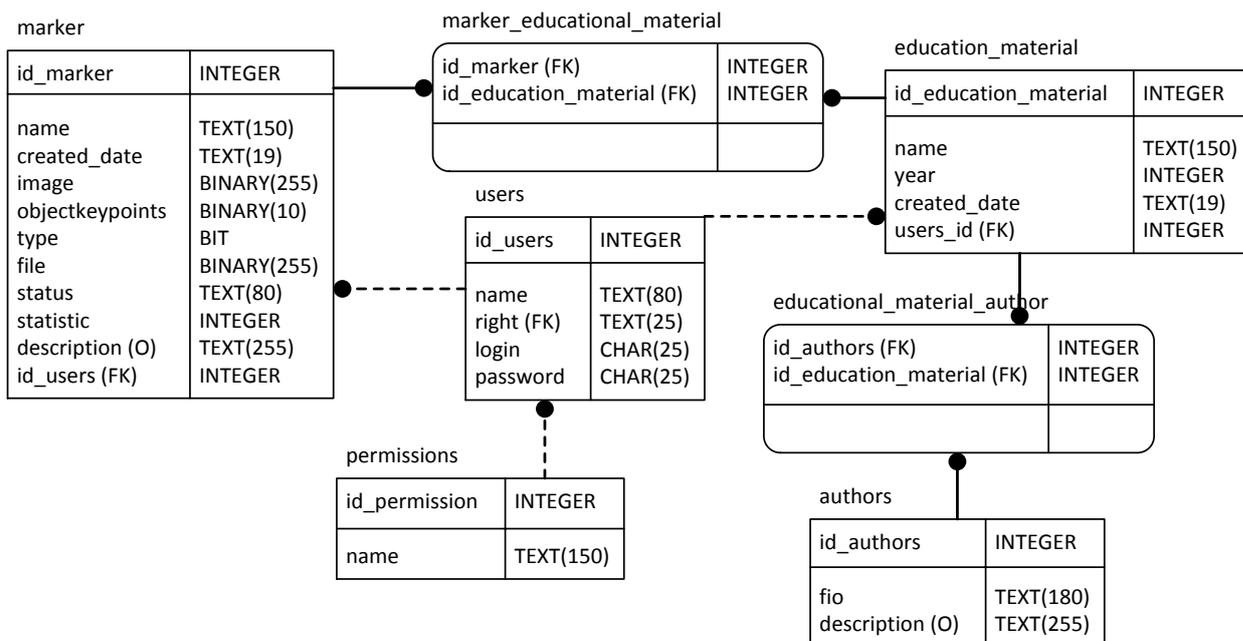


Рисунок 28. Физическая модель данных

3.3 Построение модели архитектуры

Для работы рассматриваемой системы нужны следующие модули:

Камера

Компонент камеры обеспечивает, что каждый показ кадров фиксируется и передается в модуль распознавания маркера. После включения камеры, кадр камеры автоматически преобразуется в аппаратно-зависимый формат и задает нужный размер изображения.

Конвертер изображения

Конвертер форматов преобразует кадры с формата камеры в формат, пригодный для рендеринга и для отслеживания. Данное изменение также содержит уменьшение изображение с камеры в различных разрешениях доступных в преобразованном стеке кадров.

Распознавание маркера

Компонент распознавания маркера содержит алгоритмы компьютерного зрения, для того чтобы выявлять и отслеживать объекты реального мира в рамках видеокамеры. На основе изображения с камеры, различные алгоритмы заботятся о выявлении маркеров. Итоги хранятся в состоянии предмета, который использует фоновое видео визуализации и могут быть доступны из кода приложения.

Отрисовка объекта

Этот модуль визуализации создает 3D изображение или видео, хранящееся в объекте.

Код приложения

В коде должны быть реализованы алгоритмы для всех вышеперечисленных компонентов. Для каждого обработанного кадра объект обновляется и вызывается метод отрисовки. Необходимо:

- Определять объекты для вновь обнаруженных целей, маркеров или обновленных состояний этих элементов
- Отрисовывать слой дополненной реальности

База маркеров

База маркеров создается с помощью алгоритма для создания маркеров. На основании загруженного изображения:

- a. Приведение в градации серого
- b. Бинаризация изображения (порог)
- c. Определение замкнутых областей
- d. Выделение контуров
- e. Выделение углов маркера
- f. Преобразование координаты

В результате получается двоичный файл, содержащий координаты маркера. Эти двоичные файлы добавляются в базу данных вместе с другими

параметрами маркера: название, тип, описание, дата создание, изображение и статус

На рисунке ниже (см. Рисунок 29) представлена архитектура прототипа

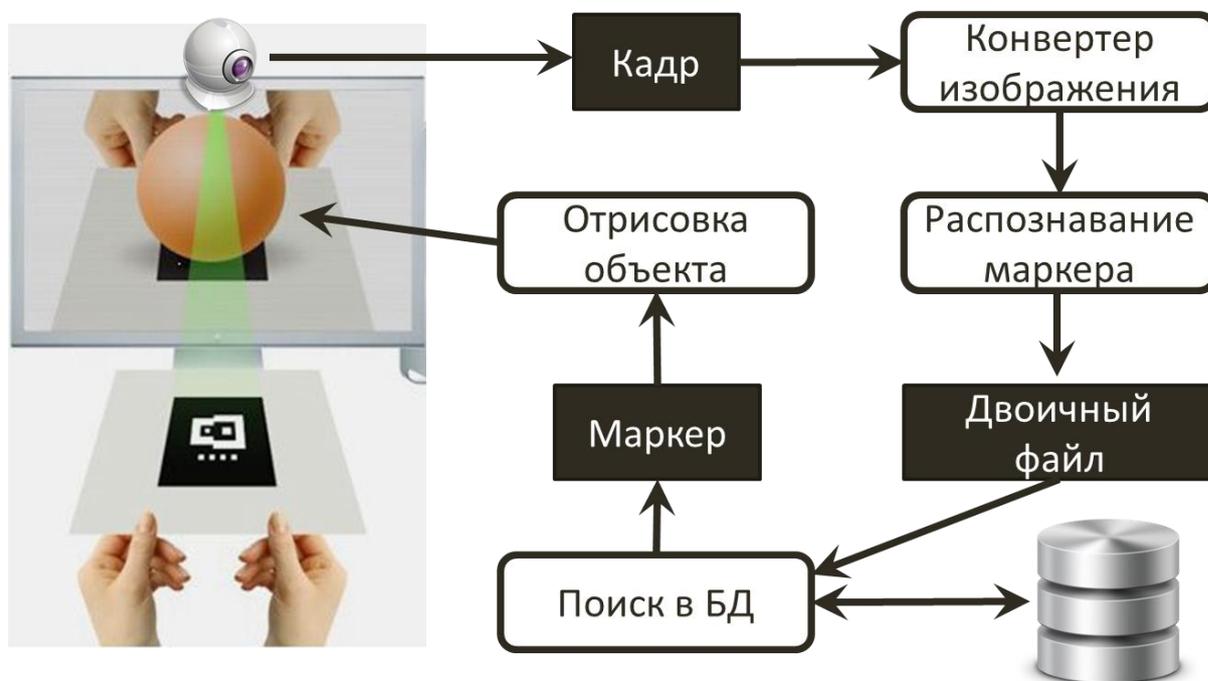


Рисунок 29. Архитектура прототипа системы

3.4 Интерфейс

На рисунке (см. Рисунок 30) представлен главный вход в систему. Пользователи разграничены в правах, так как обычный пользователь не может загружать маркеры в системы, это может делать только администратор.

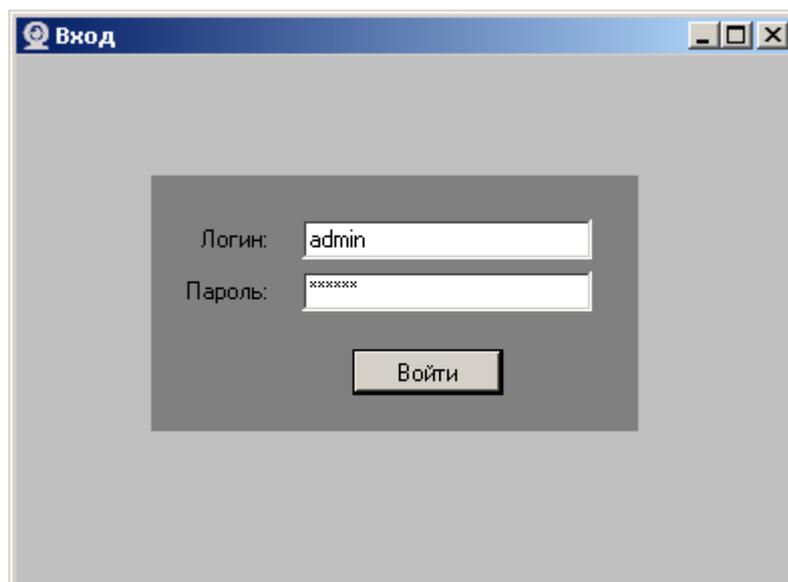


Рисунок 30. Окно входа в систему

Еще одно из возможностей администратора просматривать и редактировать пользователей (см. Рисунок 31). С данной страницы можно перейти на страницу редактирования пользователя или добавить нового пользователя в систему (см. Рисунок 32).

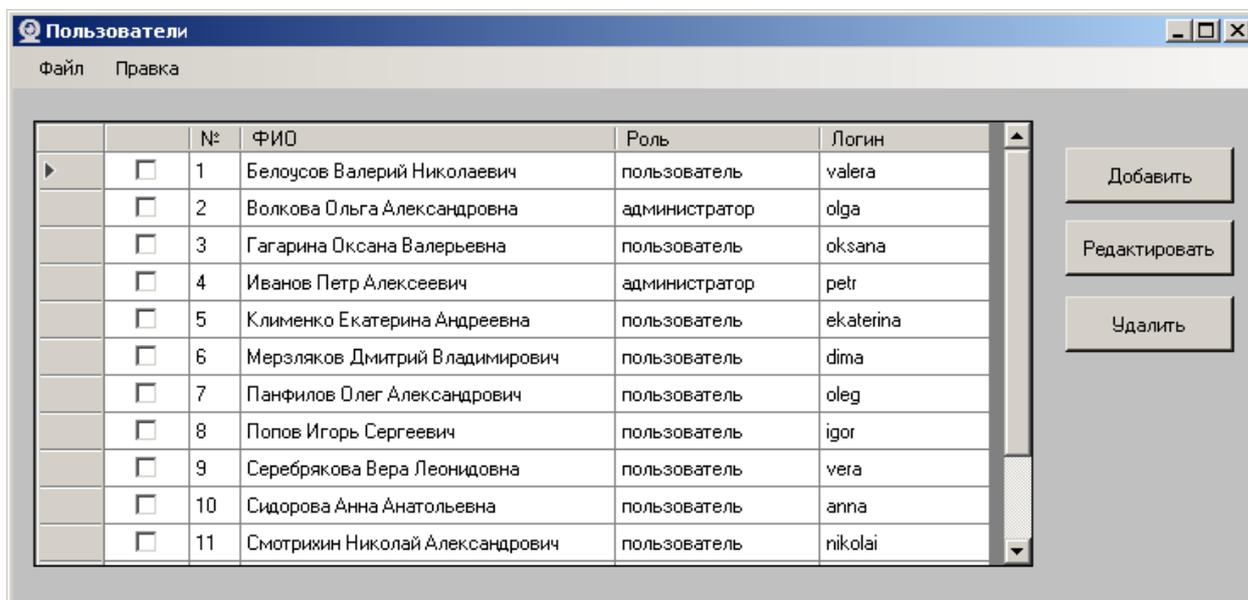


Рисунок 31. Окно просмотра пользователей

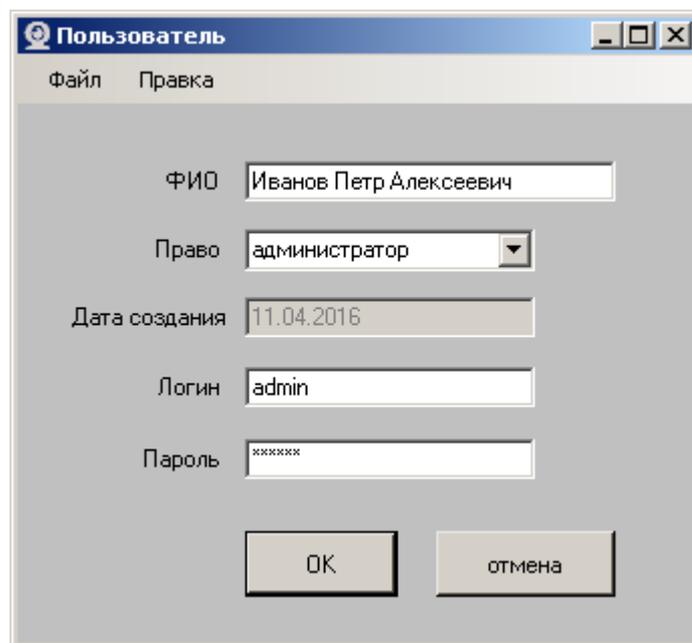


Рисунок 32. Окно редактирования пользователя

Основной задачей данной программы является распознавание объектов на изображении, для этого необходимо, чтобы была база маркеров для воспроизведения контента. Для его пополнения у администратора имеется следующий интерфейс (см. Рисунок 33, Рисунок 34).

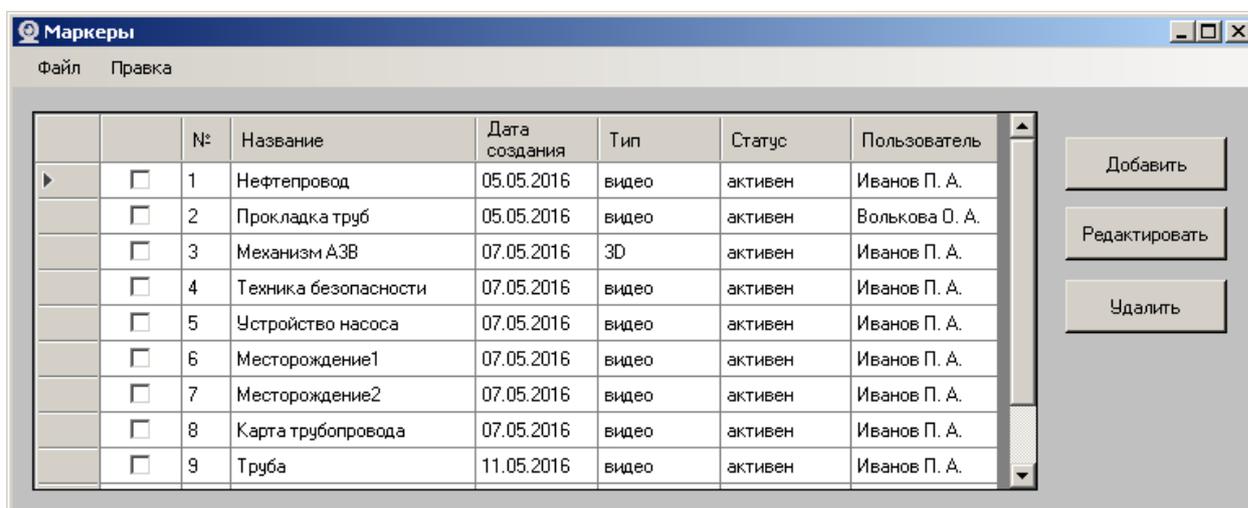


Рисунок 33. Окно просмотра списка всех маркеров

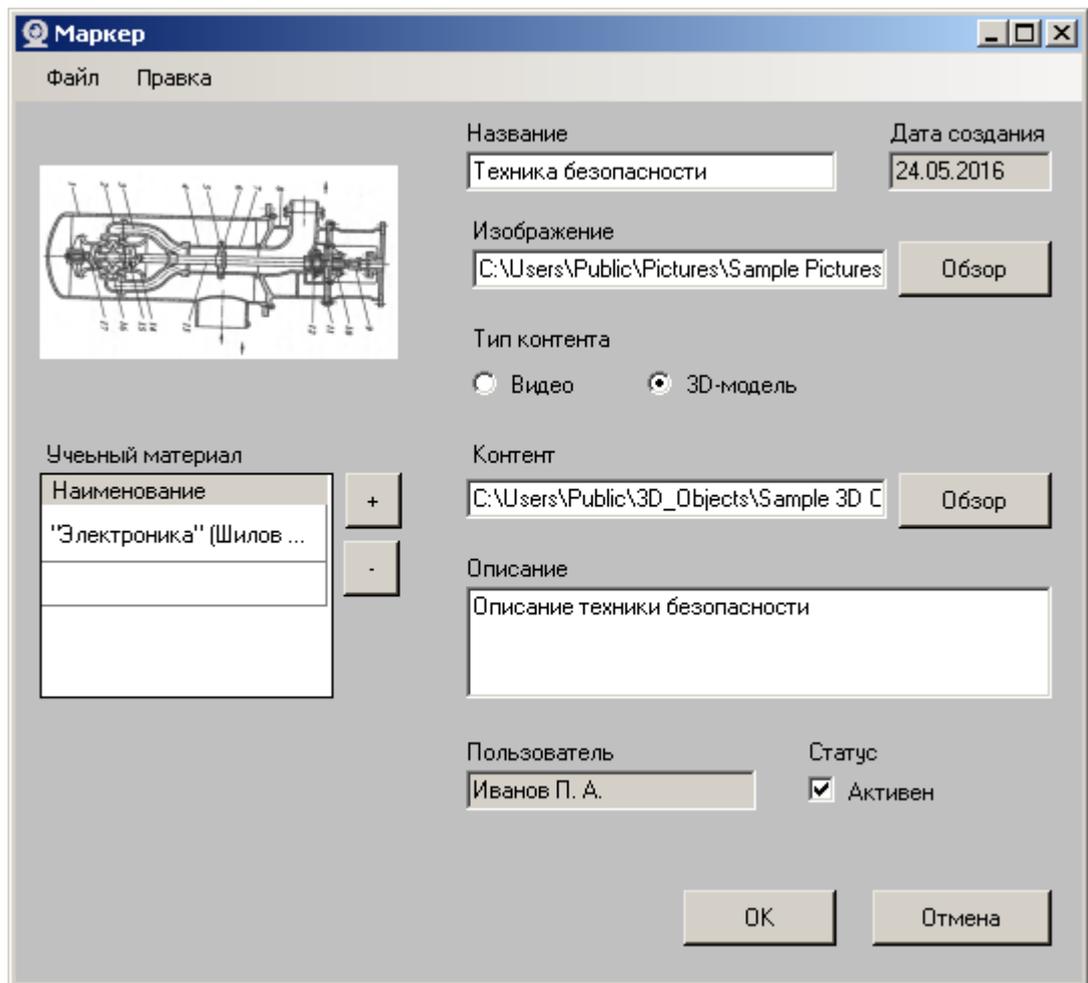


Рисунок 34. Окно для добавления и редактирования маркера

На рисунке представлен интерфейс системы, когда включена камера и направлена на учебный материал. Система пытается распознать маркер на потоке кадров, если система удалось распознать маркер, то она отображает его содержимое (см. 58Рисунок 35).

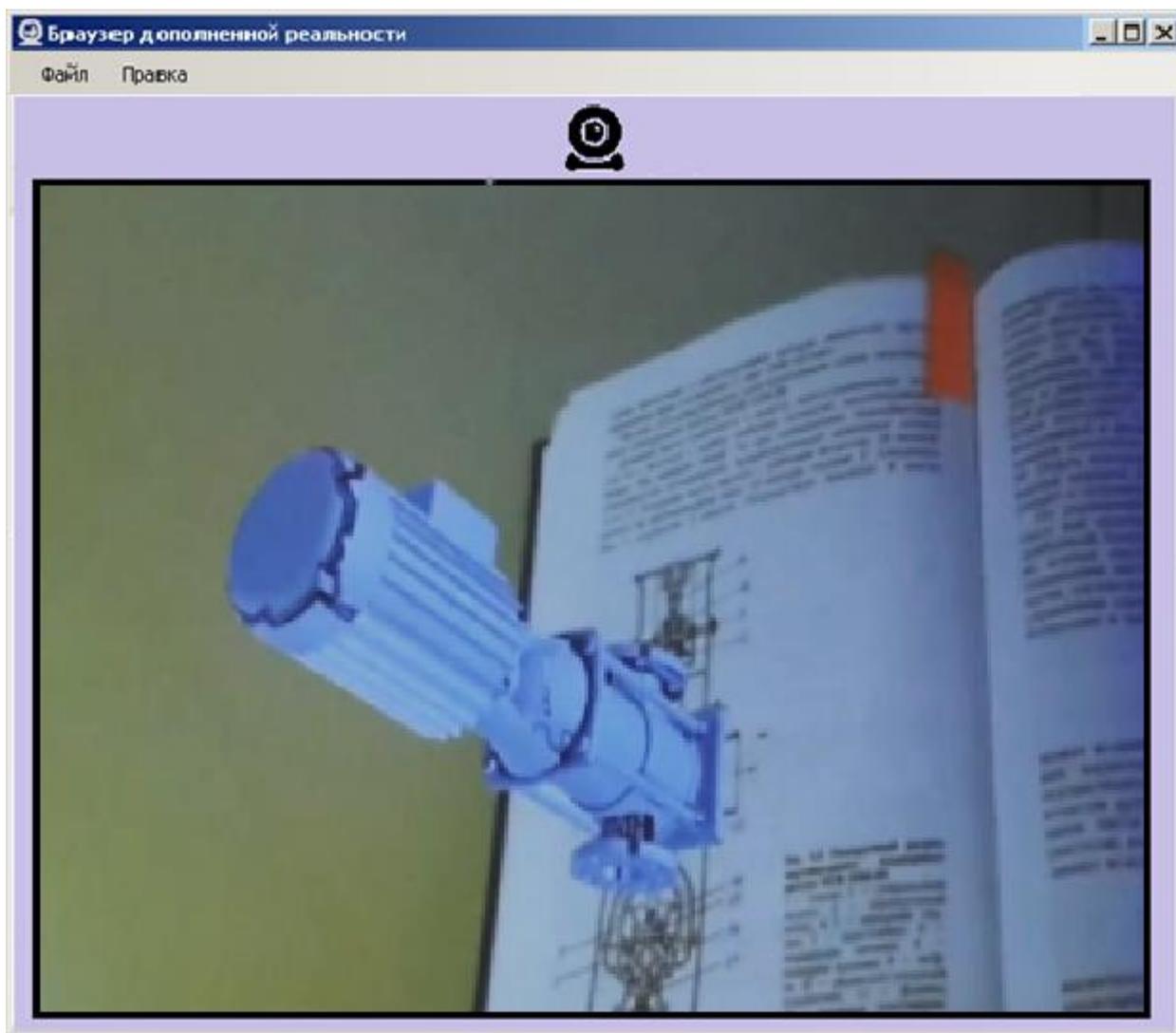


Рисунок 35. Главное окно. Отображение 3D модели

Для просмотра статистики по просмотру маркеров администратор выбирает дату начала сбора статистики на странице просмотра статистики (см. Рисунок 36), выбирает дату окончания сбора статистики, выбирает опцию группировать по учебному материалу и нажимает кнопку «Сформировать». Система формирует отчет по статистике распознавания маркеров за указанный период. Где выводится название маркера и количество раз, сколько этот маркер был распознан, за указанный период времени. (Рисунок 37). Если опция группировать по учебному материалу была выбрана, то выводится столбец учебный материал, и данные сгруппированы, по учебным материалам. (см. Рисунок 38). Внизу таблицы

отчет по статистике распознавания маркеров можно вывести на печать или в Excel.

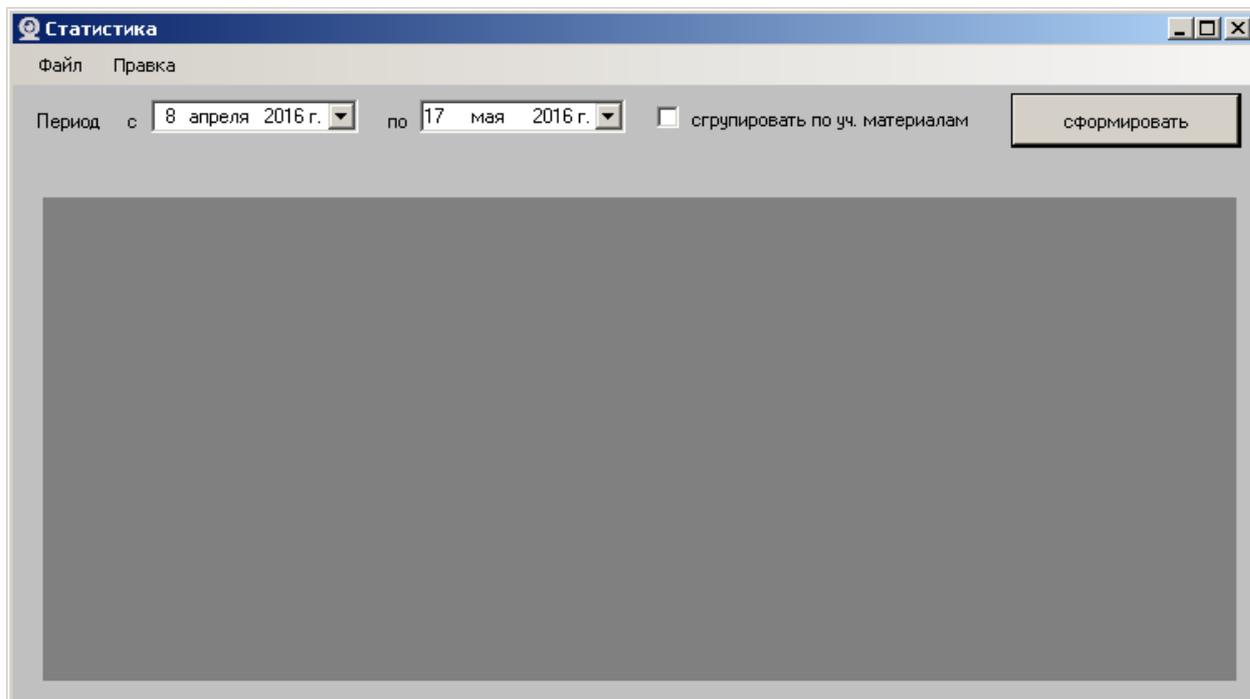


Рисунок 36. Окно просмотра статистики

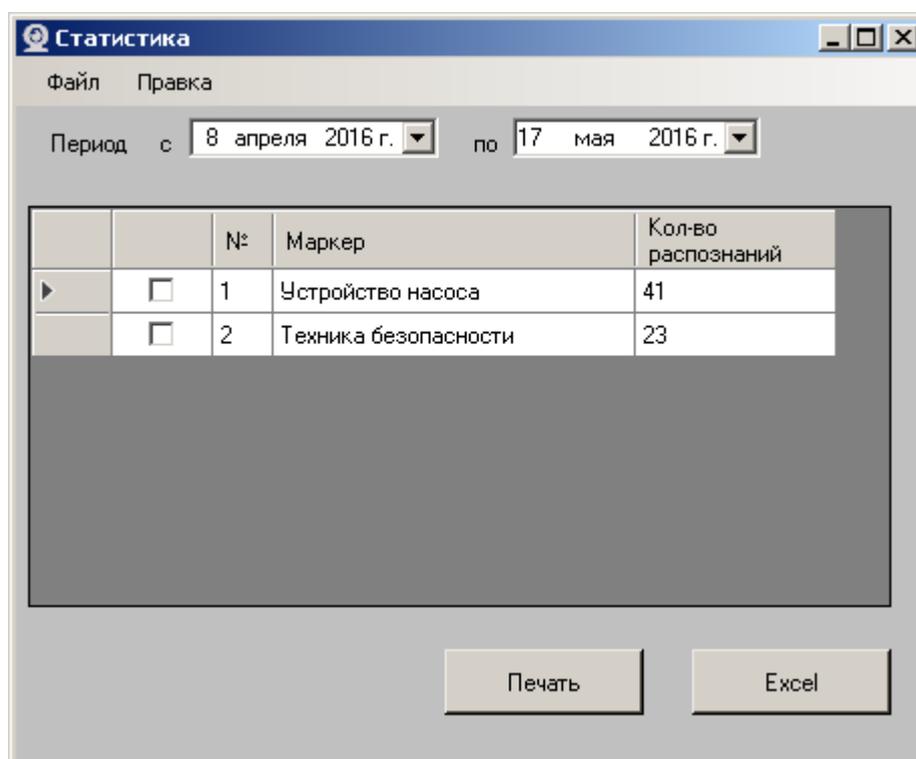


Рисунок 37. Окно просмотра статистики по маркерам

| | | № | Маркер | Учебный материал | Кол-во распознаний |
|---|--------------------------|---|----------------------|----------------------|--------------------|
| ▶ | <input type="checkbox"/> | 1 | Устройство насоса | Обучающий материал | 41 |
| | <input type="checkbox"/> | 2 | Техника безопасности | Техника безопасности | 11 |
| | <input type="checkbox"/> | 3 | Техника безопасности | Обучающий материал | 12 |

Рисунок 38. Окно просмотра статистики по маркерам, сгруппированным по учебным материалам

3.5 Выводы

В данной главе было произведено концептуального проектирования, были выделены основные сущности предметной области, построены модели данных: концептуальная, логическая в нотации IDEF1x и физическая модели данных.

Была построена функциональная модель – диаграмма вариантов использования, показывающая взаимодействие пользователя с системой и написаны к ней спецификации. Построена модель архитектуры, в которой были выделены модули: камера, конвертер изображения, распознавание маркера, отрисовка объекта, код приложения, база маркеров. Был разработан интерфейс. Был произведен анализ существующих решений – рассмотрены библиотеки дополненной реальности: OpenCV, Vuforia, ARToolKit, Metaio SDK, AForge.NET, Aurasma, Swarp SDK.

Так же в этой главе был представлен интерфейс.

ГЛАВА 4 РЕАЛИЗАЦИЯ БРАУЗЕРА ДОПОЛНЕННОЙ РЕАЛЬНОСТИ

4.1 Компоненты браузера дополненной реальности

При реализации браузера дополненной реальности была использована библиотека OpenCV.

OpenCV (англ. *Open Source Computer Vision Library*, библиотека компьютерного зрения с открытым исходным кодом) — библиотека алгоритмов, реализованная на C/C++ для компьютерного зрения, численных алгоритмов общего назначения и обработки изображений с открытым кодом, которая может свободно использоваться в академических и коммерческих целях — распространяется в условиях лицензии BSD.

В настоящее время OpenCV (англ. *Open Source Computer Vision Library*) является, наверное, самой известной и развитой библиотекой компьютерного зрения.

Подключаемые модули библиотеки указаны на рисунке ниже.

```
#include <opencv2/core/core.hpp>
#include "opencv2/highgui/highgui.hpp"
#include "opencv2/imgproc/imgproc.hpp"
#include <iostream>
#include <stdio.h>
#include <stdlib.h>
#include <opencv/cv.h>
#include <opencv/highgui.h>
```

Рисунок 39. Подключение модулей библиотеки OpenCV

Для реализации браузера дополненной реальности используется Web-камера. Программа запускается на настольном компьютере.

По данным статистики StatCounter (инструмент для анализа веб-трафика) в мире пользователями планшетов являются 5.59%, пользователями мобильных устройств 35.99%, пользователями настольных компьютеров являются 58.42% из всех пользователей Интернета (см. Рисунок 40).

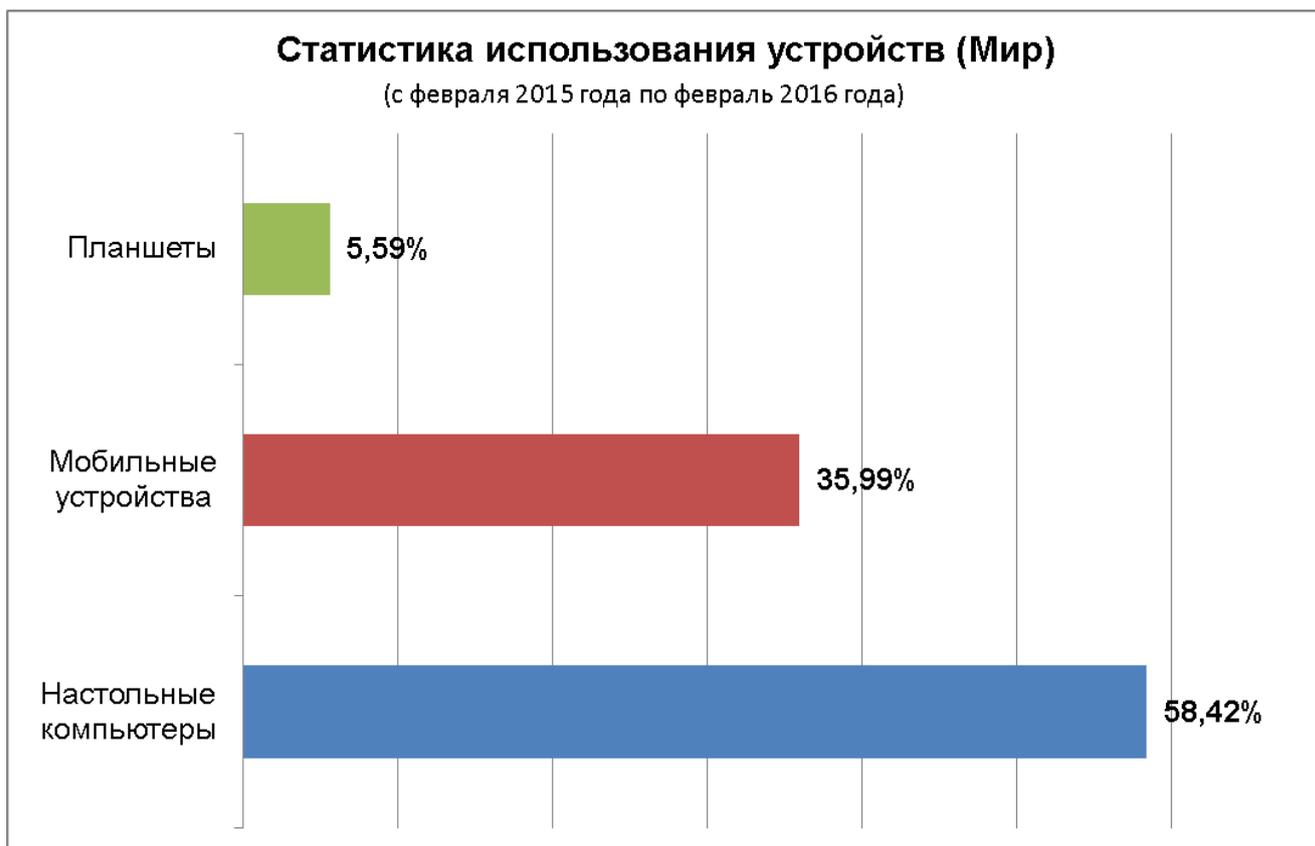


Рисунок 40. Статистика использования устройств (Мир)

По данным статистики StatCounter в России пользователями планшетов являются 5.12%, пользователями мобильных устройств 11.78%, пользователями настольных компьютеров являются 83.10% из всех пользователей Интернета (см. Рисунок 41).

Из этого можно сделать вывод, что основная часть пользователей является пользователями настольных компьютеров более половины в мире и на 30% больше в России, поэтому для реализации используется настольный компьютер и Web-камера, не смотря на общее снижение пользователей настольных компьютеров с 2014 года.



Рисунок 41. Статистика использования устройств (Российская федерация)

4.2 Алгоритм воспроизведения дополненной реальности

При разработке браузера дополненной реальности был реализован следующий алгоритм (см. Рисунок 43):

ШАГ 1. Захват изображения

ШАГ 1.1 Подключение камеры

ШАГ 1.2 Захват видеопотока с камеры

ШАГ 1.3 Считывание в цикле кадров видеопотока

ШАГ 2. Распознавание особых точек и дескрипторов

ШАГ 2.1 Нахождение особых точек на изображении

ШАГ 2.2 Вычисление дескрипторов особых точек

ШАГ 3. Сопоставление вычисленных на **ШАГЕ 2.2** дескрипторов с базой данных дескрипторов маркеров. Получение id маркеров, вектора точек

ШАГ 4. Воспроизведение соответствующего контента

ШАГ 4.1 Если возвращенное количество точек в векторе больше заданного значения, то переходим на **ШАГ 4.2** иначе на **ШАГ 4.4**

ШАГ 4.2 Получение из БД содержимого маркера

ШАГ 4.3 Вывод содержимого маркера на изображение

ШАГ 4.4 Вывод изображения на экран

Диаграмма деятельности описывающая данный алгоритм представлена на рисунке ниже (см. Рисунок 42).

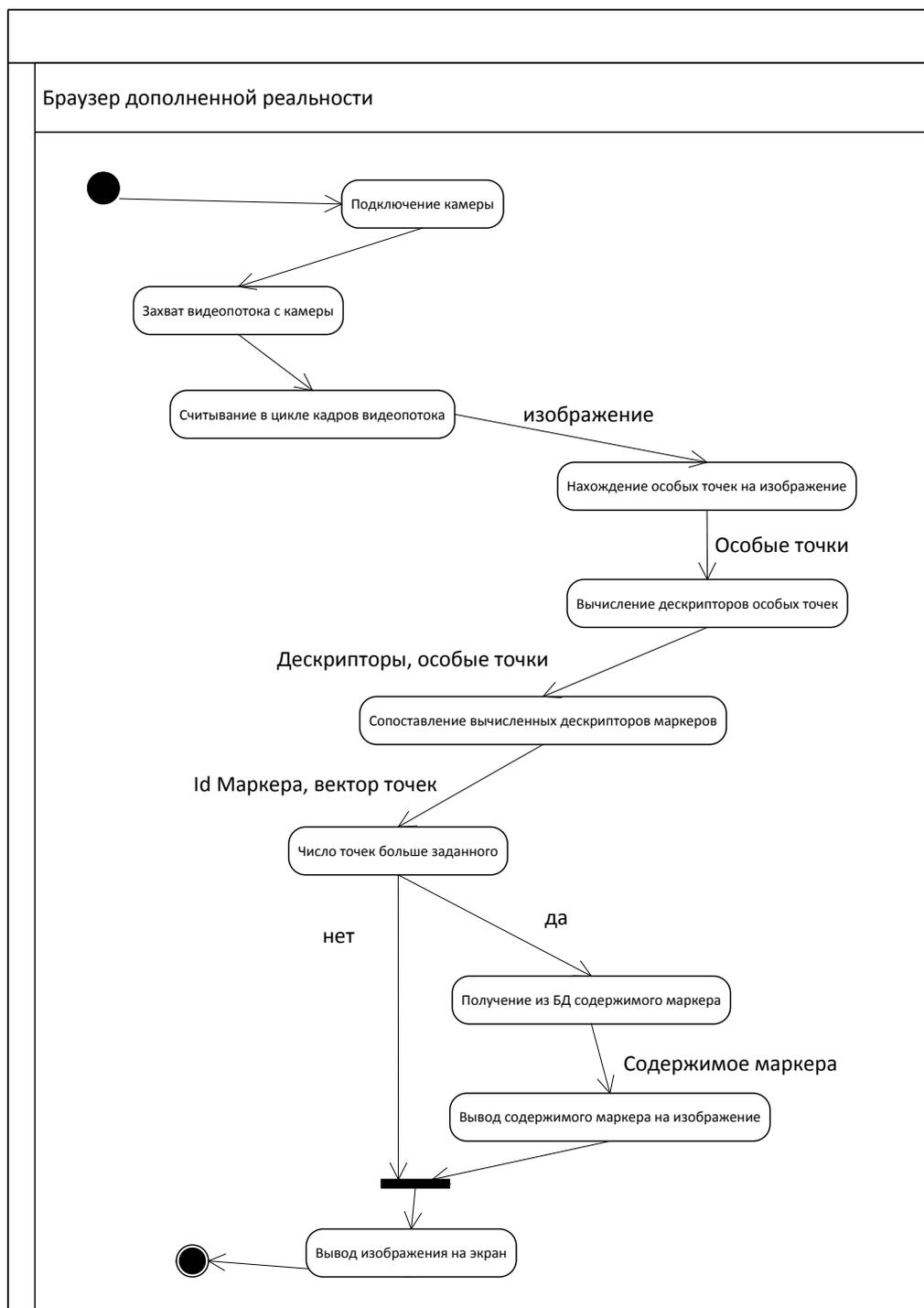


Рисунок 42. Диаграмма деятельности алгоритма воспроизведения дополненной реальности

4.3 Распознавание (обнаружение) маркера

При разработке браузера дополненной реальности был реализован следующий алгоритм:

Алгоритм распознавания маркера:

- 1) приведение в градации серого,
- 2) бинаризация изображения (порог),
- 3) определение замкнутых областей,
- 4) выделяем контуры,
- 5) выделяем углы маркера,
- 6) преобразуем координаты.

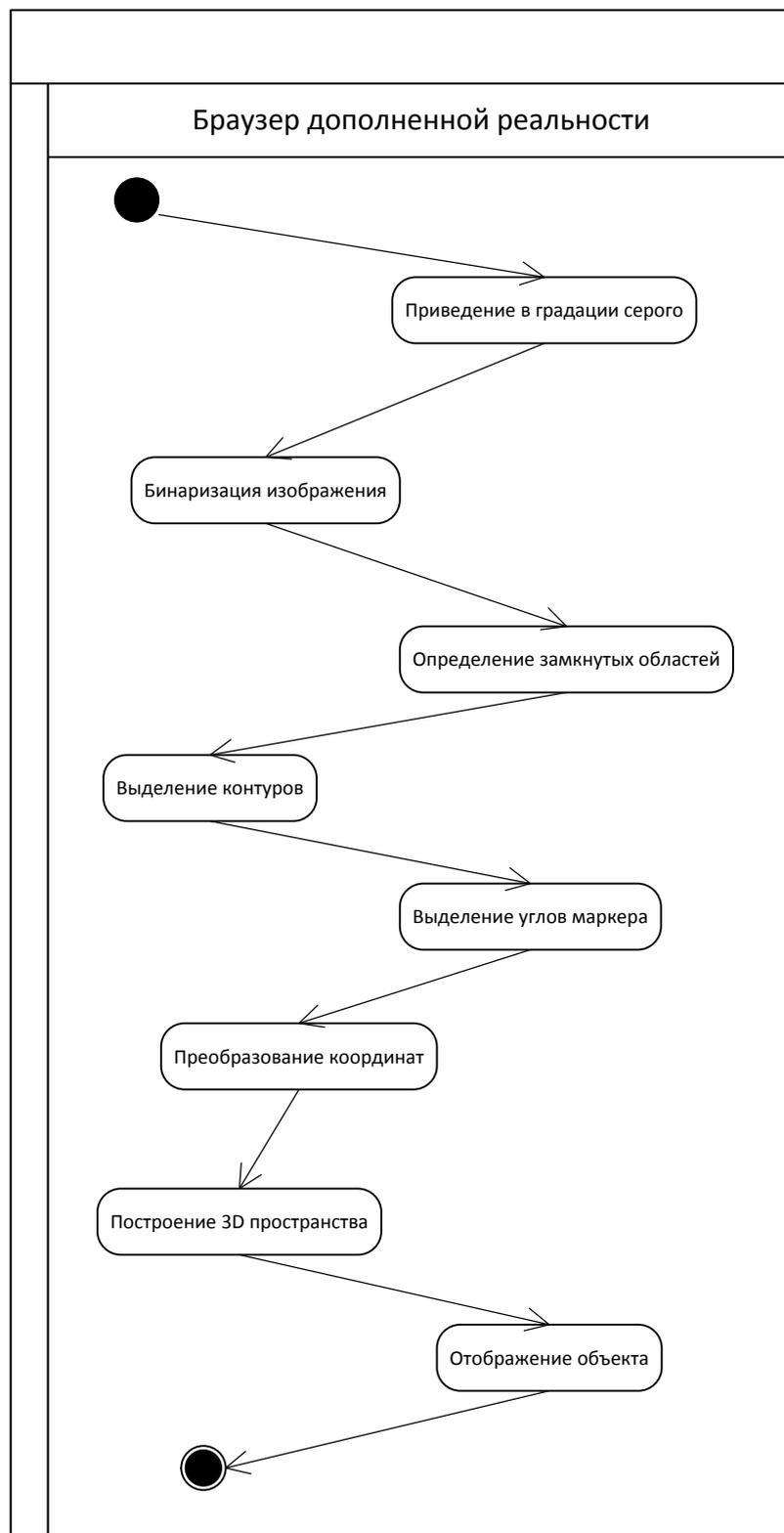


Рисунок 43. Диаграмма деятельности распознавания маркера браузера дополненной реальности

4.3.1 Приведение в градацию серого

Существует три основных способов перевода цветного изображения в градацию серого.

- Светлота(Lightness)

$$GS = (\max(R,G,B) + \min(R,G,B))/2$$

- Светимость(Luminosity)

$$GS = 0.21 \times R + 0.72 \times G + 0.07 \times B$$

- Среднее(Average)

$$GS = (R + G + B) / 3$$

Все три способа дают не сильно отличающийся друг от друга результат (см Рисунок 44). Из перечисленных способов используется вариант светимость. Изображение, полученное с Web-камеры, фиксируется в объекте `gray` типа `IplImage`, и приводится в градацию серого с помощью выбранного способа: $RGB[A] \rightarrow GRAY: Y \leftarrow 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B$ в функции `cvCvtColor` (см. Листинг 1, Рисунок 45).



Исходное изображение



"Светлота"



"Светимость"



"Среднее"

Рисунок 44. Сравнение вариантов приведения градации серого

```
IplImage* gray = cvCreateImage(cvGetSize(image), image->depth, 1);  
cvCvtColor(image, gray, CV_BGR2GRAY);
```

Листинг 1. Приведение в градацию серого

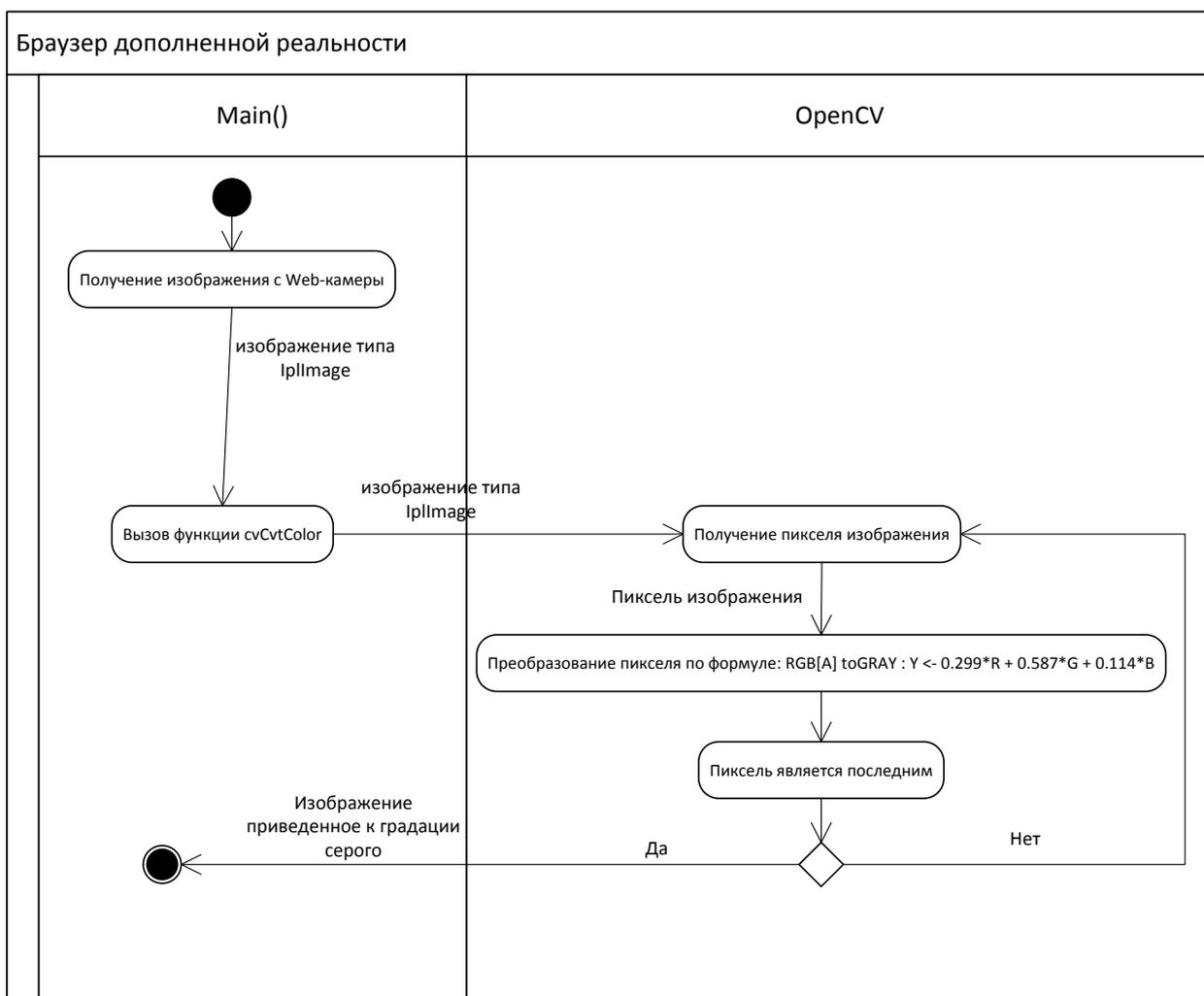


Рисунок 45. Диаграмма деятельности. Приведение в градации серого

4.3.2 Бинаризация изображения

Следующий шаг алгоритма – бинаризация изображения. Бинаризация – операция нахождения порогового значения, в результате которой получается бинарное изображение. Целью этой операции является уменьшение количества информации. В процессе данной операции исходное изображение (в градациях серого) преобразуется в черно-белое изображение, пиксели которого имеют только два значения – 0 или 1

При бинаризации изображения находится порог яркости. Данный порог не высчитывается с помощью различных алгоритмов.

Простое пороговое преобразование никак не предусматривает, что части предметов могут обладать различной яркостью из-за разности в

освещённости. Это исправляется путем использования адаптивного порогового преобразования:

- берется среднее значение пикселя окрестности
- пиксели окрестности умножаются на весовой коэффициент (взвешиваются) в соответствии с функцией, гауссовой функцией (см Листинг 2).

```
cvAdaptiveThreshold(image, gray, 250, CV_ADAPTIVE_THRESH_GAUSSIAN_C, CV_THRESH_BINARY, 7, 1);
```

Листинг 2. Адаптивная бинаризация изображения

Существует разные критерии расчета порога: Отса, Бернсена, Эйквеля, Ниблэка и т.п. Наиболее результативным, равно как по быстродействию, так и по качеству, считается критерий Отса.

Данный способ применяет гистограмму распределения значений яркости пикселей изображения. Создается гистограмма по значениям $p_i = n_i/N$, где N – это общее кол-во пикселей на изображении, n_i – это кол-во пикселей с уровнем яркости i . Спектр яркостей разделяется на два класса с помощью порогового значения уровня яркости k , k — целое значение от 0 до L . Каждому классу соответствуют относительные частоты $\omega_0 \omega_1$:

$$\omega_0(k) = \sum_{i=1}^k p_i \quad \omega_1(k) = \sum_{i=k+1}^L p_i = 1 - \omega_0(k)$$

$$\mu_0(k) = \sum_{i=1}^k \frac{i p_i}{\omega_0} \quad \mu_1(k) = \sum_{i=k+1}^L \frac{i p_i}{\omega_1}$$

Средние уровни для каждого из двух классов изображения:

Далее вычисляется максимальное значение оценки качества разделения изображения на две части:

где $(\sigma_{\text{кл}})^2 = \omega_0 \omega_1 (\mu_1 - \mu_0)^2$, – межклассовая дисперсия, а $(\sigma_{\text{общ}})^2$ – это общая дисперсия для всего изображения целиком.

$$\eta(k) = \max_{k-1}^{L-1} \left(\frac{\sigma_{кл}^2(k)}{\sigma_{общ}^2} \right)$$

Ниже представлен код функции, который работает с 24-битным изображением, преобразуя его в 8-битное бинаризованное (см. Листинг 3).
 Диаграмма, описывающая работу алгоритма Рисунок 46.

```

void Threshold(IplImage* Image, IplImage* gray, CvRect Rect)
{
    uchar* char1;
    char1 = (uchar*)(Image->imageData);
    uchar* char2;
    char2 = (uchar*)(gray->imageData);
    int i, j;
    for (i = 0; i<Rect.height; i++)
        for (j = 0; j<Rect.width; j++)
        {
            if (char1[(Rect.x + j) * 3 + 0 + (Rect.y + i)*Image->widthStep]>100 &&
                char1[(Rect.x + j) * 3 + 1 + (Rect.y + i)*Image->widthStep]>140)
                char2[j + i*gray->widthStep] = 255;
            else
                char2[j + i*gray->widthStep] = 0;
        }
    }

```

Листинг 3. Критерий Отса

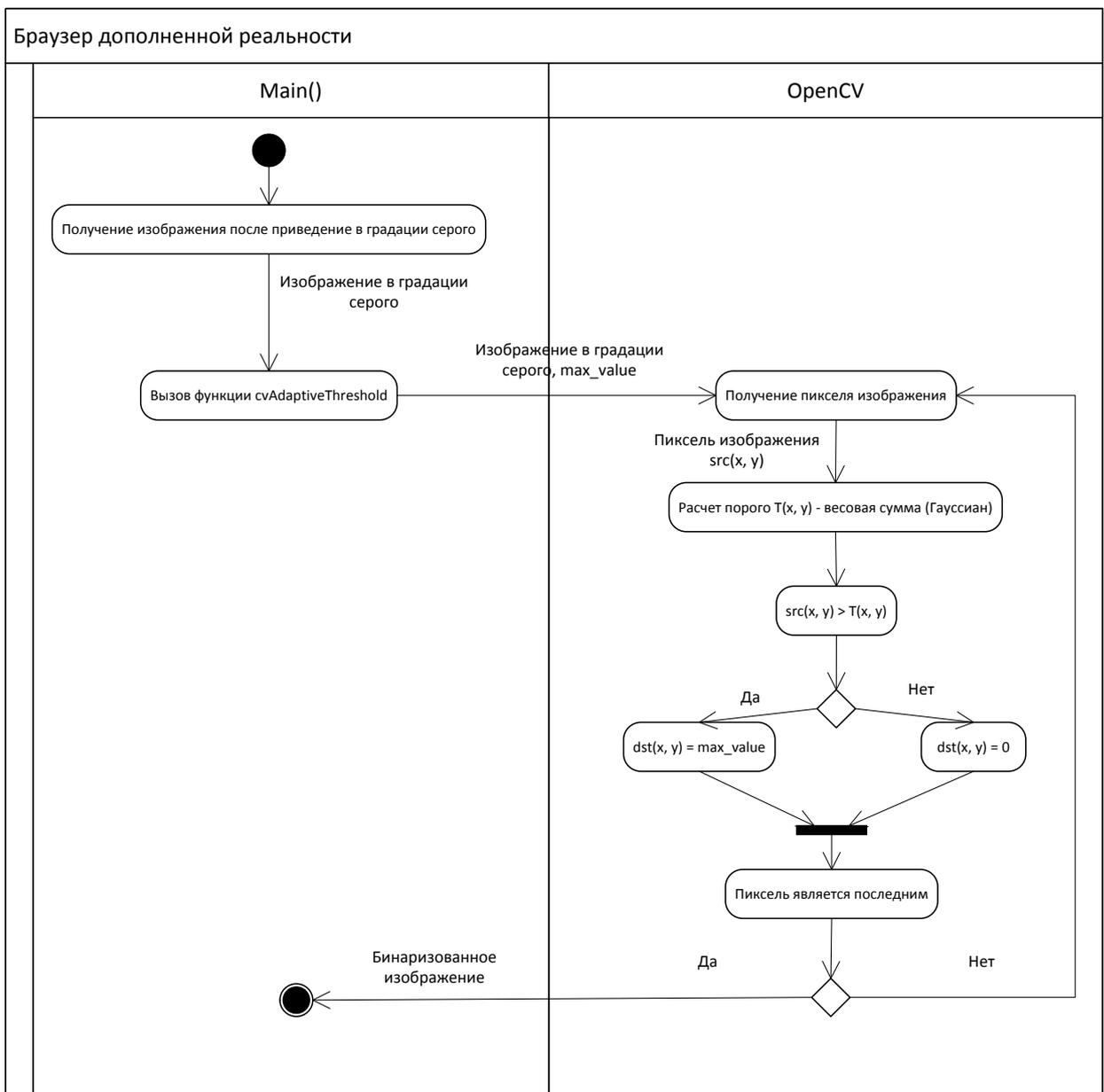


Рисунок 46. Диаграмма деятельности. Бинаризация изображения.

4.3.3 Определение замкнутых областей и выделение контуров

Далее по ходу алгоритма распознавания маркера идет определение замкнутых областей и выделение контуров. Цель состоит в получении обработанного изображения с идеально выделенными контурами. Ниже представлены несколько методов для обнаружения границ:

- Перекрёстный оператор Робертса

$$\begin{bmatrix} +1 & 0 \\ 0 & -1 \end{bmatrix} \text{ and } \begin{bmatrix} 0 & +1 \\ -1 & 0 \end{bmatrix}$$

- Оператор Собеля

$$G = \sqrt{G_x^2 + G_y^2}$$

$$\Theta = \arctan\left(\frac{G_y}{G_x}\right)$$

$$G = |G_x| + |G_y|$$

- Оператор Превитта

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -1 & 0 & +1 \\ -1 & 0 & +1 \end{bmatrix} * A \quad \text{and} \quad G_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ +1 & +1 & +1 \end{bmatrix} * A$$

- Оператор Кирша

$$h_{n,m} = \max_{z=1,\dots,8} \sum_{i=-1}^1 \sum_{j=-1}^1 g_{ij}^{(z)} \cdot f_{n+i,m+j}$$

- Оператор Робинсона

$$E = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad NE = \begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix} \quad N = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad NW = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -2 \end{bmatrix}$$

$$W = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \quad SW = \begin{bmatrix} 0 & -1 & -2 \\ 1 & 0 & -1 \\ 2 & 1 & 0 \end{bmatrix} \quad S = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad SE = \begin{bmatrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{bmatrix}$$

- Marr-Hildreth Edge Detector;

$$\nabla^2 g(x, y) = \frac{1}{\sigma^2} \left(\frac{(x^2 + y^2)}{\sigma^2} - 2 \right) e^{-\frac{(x^2 + y^2)}{2\sigma^2}}$$

- Выделение контура методом LoG

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

- Оператор Кэнни

$$H_{ij} = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{(i+(k+1))^2 + (j-(k+1))^2}{2\sigma^2}\right), i, j = 1 \dots (2k+1)$$

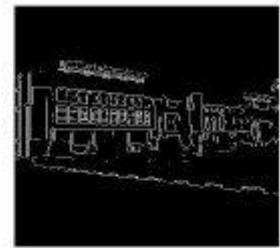
На рисунке ниже представлены деятельность представленных выше алгоритмов выявления границ предметов на изображении (см Рисунок 47). При сопоставлении изображений можно выделить закономерность: операторы Робертса, Собеля и Превитта предоставляют весьма различные итоги. Marr-Hildreth, LoG и Канни почти идентично находят контуры предмета, Кирш и Робинсон дали такой же результат. Однако наблюдая за полученными результатами можно сделать заключение о том, что метод Канни справляется на порядок лучше других.



Original



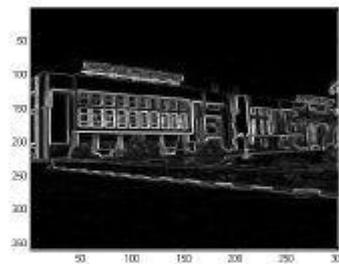
Roberts



Sobel



Prewitt



Kirsch



Robinson



Marr-Hildreth



LoG



Canny

Рисунок 47. Оригинальное изображение и итог функционирования различных методов выделения контуров

Функция для определения контуров областей определяет замкнутые области, используется алгоритм Канны из библиотеки OpenCV (см Листинг 4)

```
CVAPI(int) cvFindContours(CvArr* image, CvMemStorage* storage, CvSeq** first_contour,
int header_size CV_DEFAULT(sizeof(CvContour)),
int mode CV_DEFAULT(CV_RETR_LIST),
int method CV_DEFAULT(CV_CHAIN_APPROX_SIMPLE),
CvPoint offset CV_DEFAULT(cvPoint(0, 0)));
```

Листинг 4. Определение замкнутых областей

Метод Канни основывается на трех аспектах:

- Хорошее выявление – сигнал/шум
- Хорошая местоположение (верно установленные положения границы)
- Единственный отклик на одну границу

Из данных критериев создается целевая функция стоимости ошибок, минимизацией которой находится "оптимальный" линейный оператор с целью свертки с изображением.

Метод детектора границ Канни не ограничивается вычислением градиента сглаженного изображения. В контуре границы оставляются только точки максимума градиента изображения, а не максимальные точки, вблизи с границей, удаляются. Здесь также используются сведения о направлении границы с целью того, чтобы устранить точки непосредственно вблизи с границей и не разъединять саму границу возле локальных максимумов градиента. Далее с помощью двух порогов удаляются слабые границы. Часть границы при этом обрабатывается как целое. В случае если значение градиента где-нибудь на прослеживаемой части превзойдет верхний порог, в таком случае данная часть остается также "допустимой" границей и в тех местах, где значение градиента опускается ниже данного порога, вплоть до тех пор пока она не станет ниже нижнего порога. В случае если же отсутствует хоть одна точка со значением большим верхнего порога на всем фрагменте, то он удаляется. Такой гистерезис позволяет уменьшить в выходных границах число разрывов. Включение в метод Канни шумоподавления с одной стороны увеличивает стабильность итогов, а с другой – повышает вычислительные расходы и приводит к искажению и в том числе к потере подробностей границ. Так, например, подобным методом скругляются углы объектов и разрушаются границы в точках соединений.

В общем виде детектор границ Канни работает соответствующим способом:

Начальное изображение, представленное в виде функции $f(x, y)$, подвергается операции сглаживания с помощью свертки:

$$g(x, y) = G_0(x, y, \sigma) * f(x, y), \quad G_0(x, y, \sigma) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right)$$

где σ – параметр степени сглаживания (чем выше σ , тем больше сглаживается изображение).

Вычисление градиента изображения в вертикальном g_v и горизонтальном g_H направлениях с помощью одного из операторов первой производной (Робертса, Собела, Превитта и др.). Полученные значения объединяются с помощью известной формулы:

$$E(i, j) = \sqrt{(g_v(i, j))^2 + (g_H(i, j))^2}$$

Направление градиента:

$$\Theta(i, j) = \tan^{-1} [g_v(i, j) / g_H(i, j)]$$

$$E_T(i, j) = \begin{cases} E(i, j) & \text{если } E(i, j) > T \\ 0 & \text{иначе} \end{cases}$$

где T – выбирается таким образом, чтобы компоненты границы были сохранены, в то время как большинство шумов заглушено.

Далее производится процедура *non-maximal suppression*. Метод уточняет обнаруженные на прошлом шаге контуры посредством обнуления значений градиента, соответствующих тем элементам изображения, которые на самом деле не находятся на вершине «гребня» градиента. Процедура осуществляется с применением двух порогов T_1 и T_2 , при этом $T_1 > T_2$. Анализ наступает в точке на «гребне», соответствующее значение градиента которой больше, чем T_1 , и длится вплоть до тех пор, пока высота «гребня» станет меньше порога T_2 . Эта процедура нужна для определения значимых контуров изображения. (см Рисунок 48)

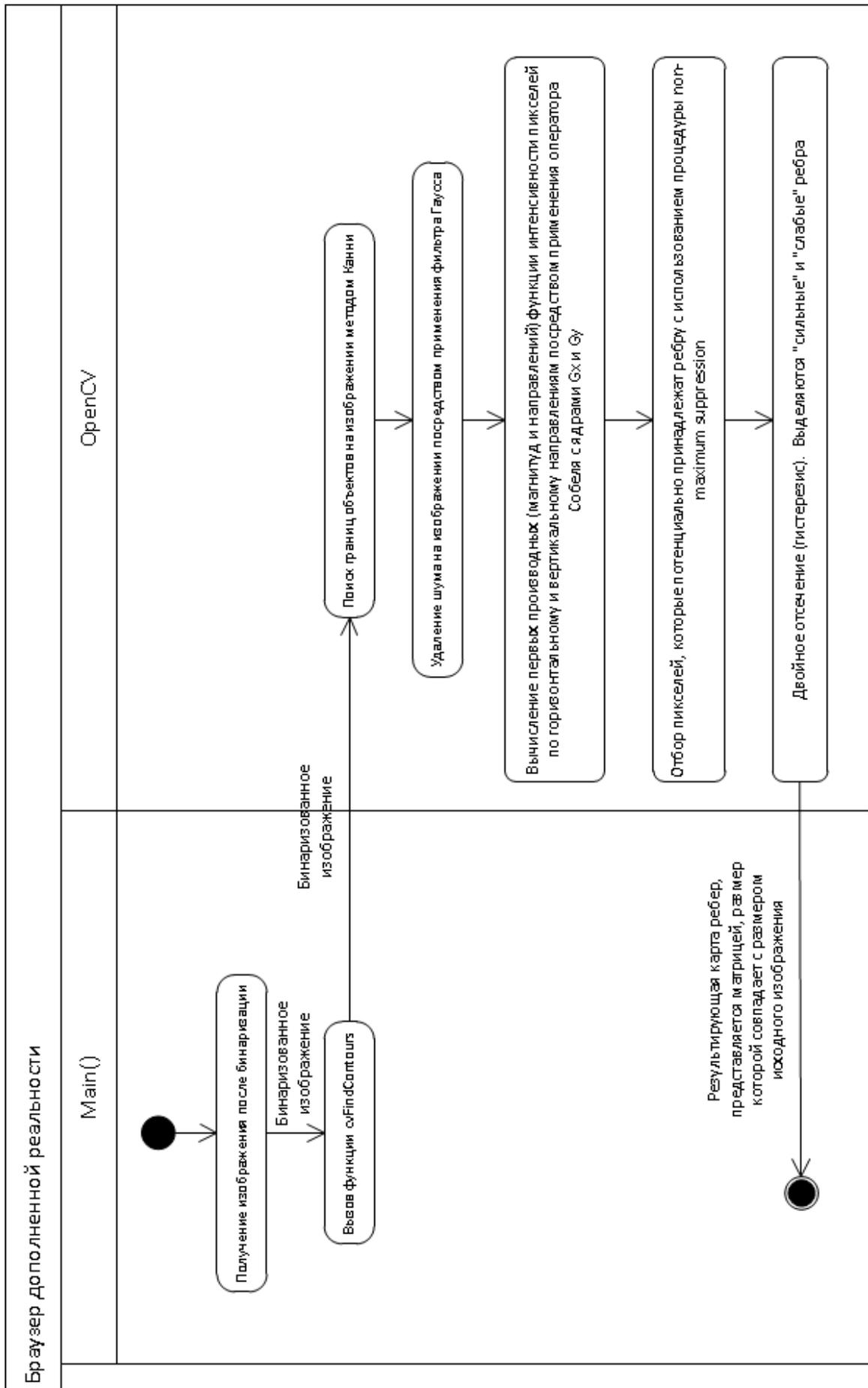


Рисунок 48. Диаграмма деятельности. Определение замкнутых областей.

4.3.4 Выделение углов маркера

В библиотеке OpenCV за выделение углов отвечает функция `cvFindCornerSubPix()` (см. Листинг 5).

```
cvFindCornerSubPix(gray, corners, corner_count, cvSize(11, 11), cvSize(-1, -1),
cvTermCriteria(CV_TERMCRIT_EPS + CV_TERMCRIT_ITER, 30, 0.1));
```

Листинг 5. Выделение углов маркера

Выделение углов маркера базируется на том, что каждый градиент из центра \mathbf{q} к точке \mathbf{p} , находящийся в окрестности \mathbf{q} , ортогонален к градиенту изображения в \mathbf{p} в зависимости от изображения и замеренных шумов (см **Ошибка! Источник ссылки не найден.**).

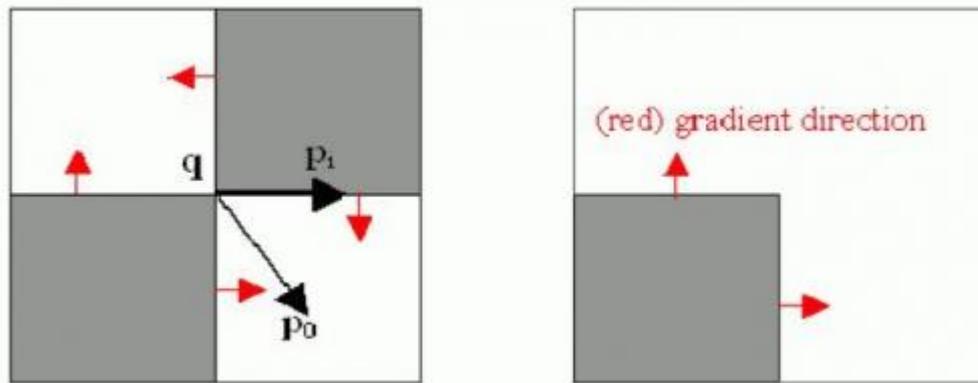


Рисунок 49. Выделение углов маркера

$$\text{Рассмотрим выражение } \varepsilon_i = \mathbf{DI}_{p_i}^T \cdot (\mathbf{q} - \mathbf{p}_i) \quad (1)$$

где \mathbf{DI} – градиент изображения в одной из точек \mathbf{p}_i в окрестности \mathbf{q} . Значение \mathbf{q} обязано быть обнаружено такое, что выражение (1) минимизируется. Система уравнений может быть определена при обнулении (1):

$$\sum_i (\mathbf{DI}_{p_i} \cdot \mathbf{DI}_{p_i}^T) \mathbf{q} = \sum_i (\mathbf{DI}_{p_i} \cdot \mathbf{DI}_{p_i}^T \mathbf{p}_i)$$

где градиенты суммируются в окрестности («окне поиска») \mathbf{q} . Вызов первого градиента \mathbf{G} и второго градиента \mathbf{b} дает: $\mathbf{q} = \mathbf{G}^{-1} \cdot \mathbf{b}$

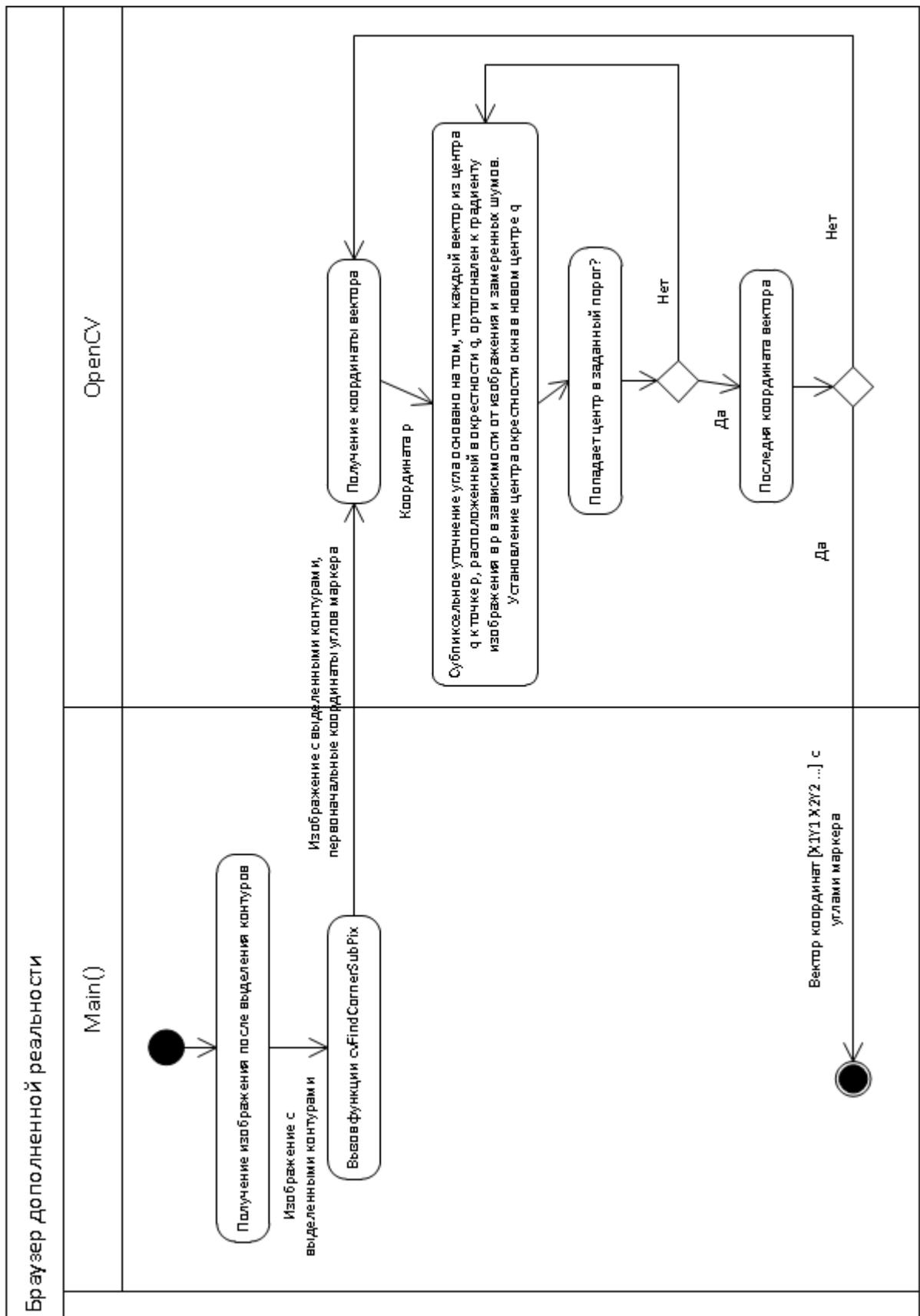


Рисунок 50. Диаграмма деятельности. Выделение углов маркера.

Алгоритм определяет центр окрестности окна в данном новом центре q и потом осуществляет итерации пока центр не станет попадать в установленный порог. (см Рисунок 50)

4.3.5 Преобразование координат

Преобразования координат – аффинные преобразования. Имеет место быть соотношение для любых двух точек A и B :

$|A'B'| = k|AB|$, где k — не равное нулю число, называемое **коэффициентом подобия**.

Необходимо изменить размер проекции, в зависимости с какого угла смотрит камера. На основе полученных результатов в предыдущих пунктах (полученный прямоугольник) (см Листинг 6, Рисунок 51).

```
CvPoint2D32f p[4];
CvPoint2D32f q[4];

frame = cvQueryFrame(vid);
if (!frame)
    MessageBox::Show("error frame",
        "error frame", MessageBoxButtons::OK);

IplImage* blank = cvCreateImage(cvGetSize(frame), 8, 3);
cvZero(blank);
cvNot(blank, blank);

q[0].x = (float)frame->width * 0;
q[0].y = (float)frame->height * 0;
q[1].x = (float)frame->width;
q[1].y = (float)frame->height * 0;

q[2].x = (float)frame->width;
q[2].y = (float)frame->height;
q[3].x = (float)frame->width * 0;
q[3].y = (float)frame->height;

p[0].x = corners[x_x].x;
p[0].y = corners[y_y].y;
p[1].x = corners[x_x1].x;
p[1].y = corners[y_y1].y;

p[2].x = corners[x_x2].x;
p[2].y = corners[y_y2].y;
p[3].x = corners[x_x3].x;
p[3].y = corners[y_y3].y;
```

Листинг 6. Преобразование координат

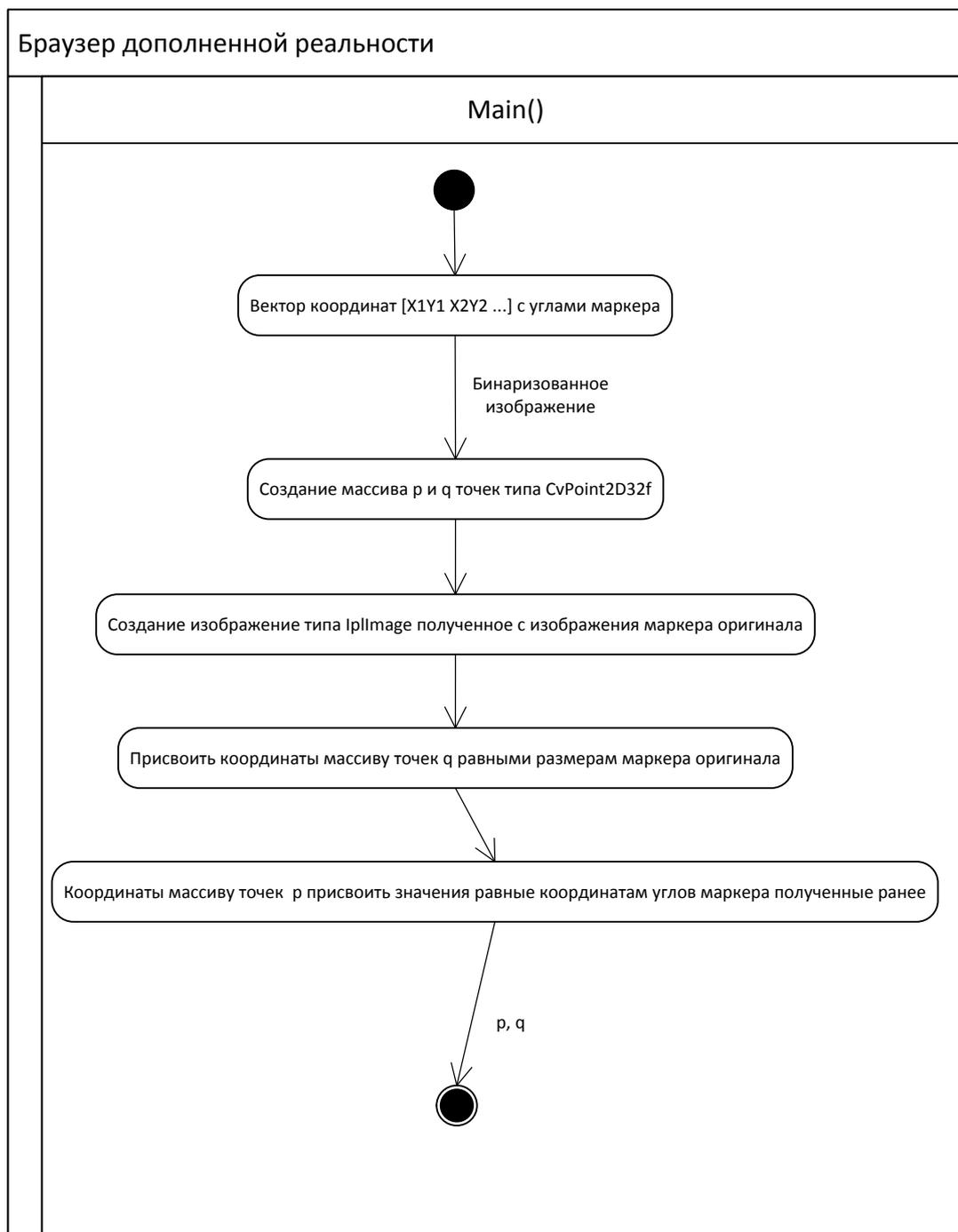


Рисунок 51. Диаграмма деятельности. Преобразование координат

4.4 Построение 3D пространства

Исходя из координат, полученных в предыдущем пункте, преобразуется виртуальный объект (см. Листинг 7). Известны новые координаты объекта. Объект перестраивается, с помощью функции `cvWarpPerspective` и заново перезаписывается в памяти программы для последующего отображения. (см. Рисунок 52)

```

cvGetPerspectiveTransform(q, p, warp_matrix);

cvZero(neg_img);
cvZero(cpy_img);

cvWarpPerspective(frame, neg_img, warp_matrix);
cvWarpPerspective(blank, cpy_img, warp_matrix);
cvNot(cpy_img, cpy_img);

cvAnd(cpy_img, image, cpy_img);
cvOr(cpy_img, neg_img, image);

```

Листинг 7. Построение 3D пространства

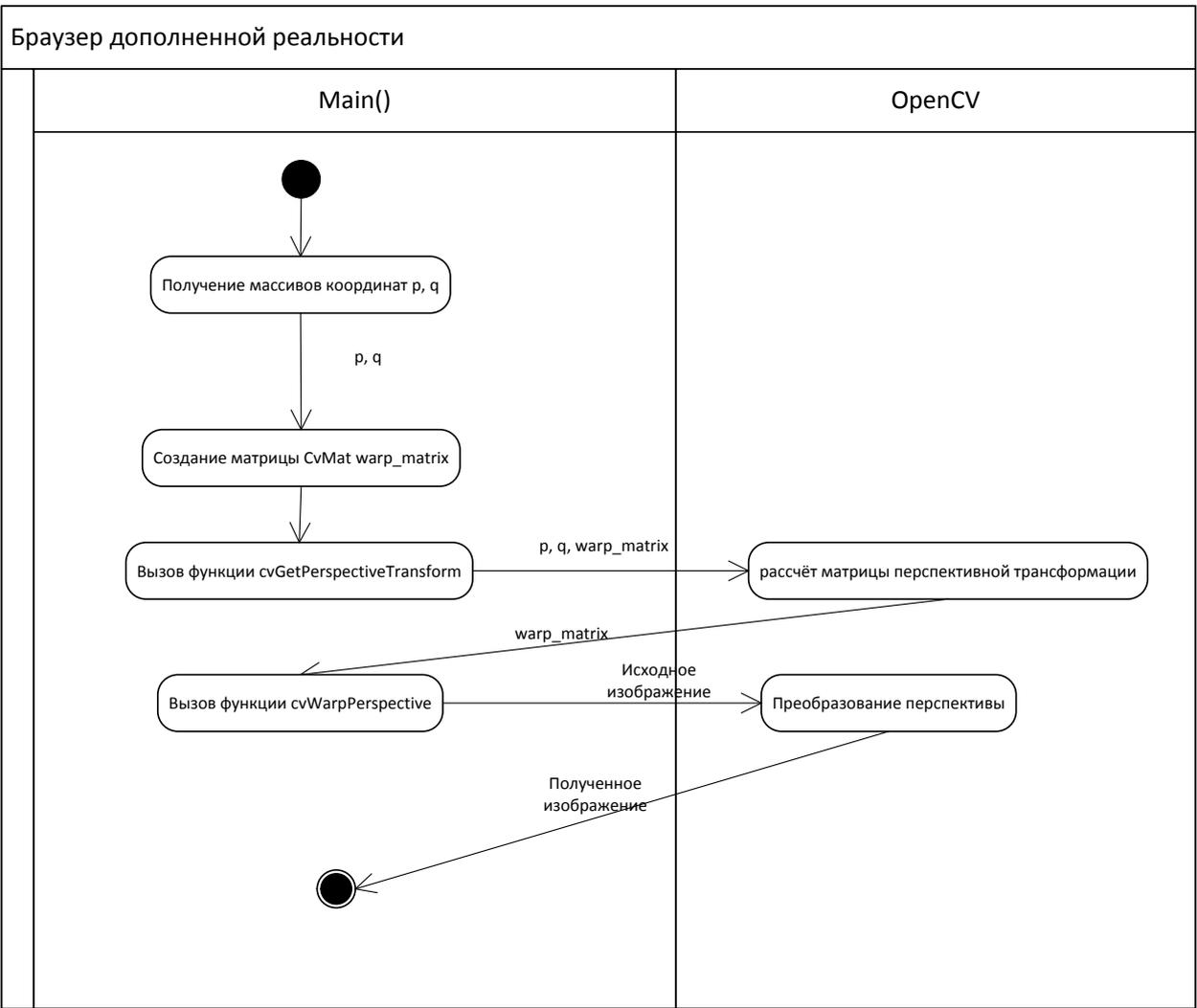


Рисунок 52. Диаграмма деятельности Построение 3D пространства

4.5 Отображение виртуального объекта

Для вывода виртуального объекта используется функция cvShowImage (см. Листинг 8б, Рисунок 53).

```
cvShowImage("W", image);
```

Листинг 8. Отображение виртуального объекта

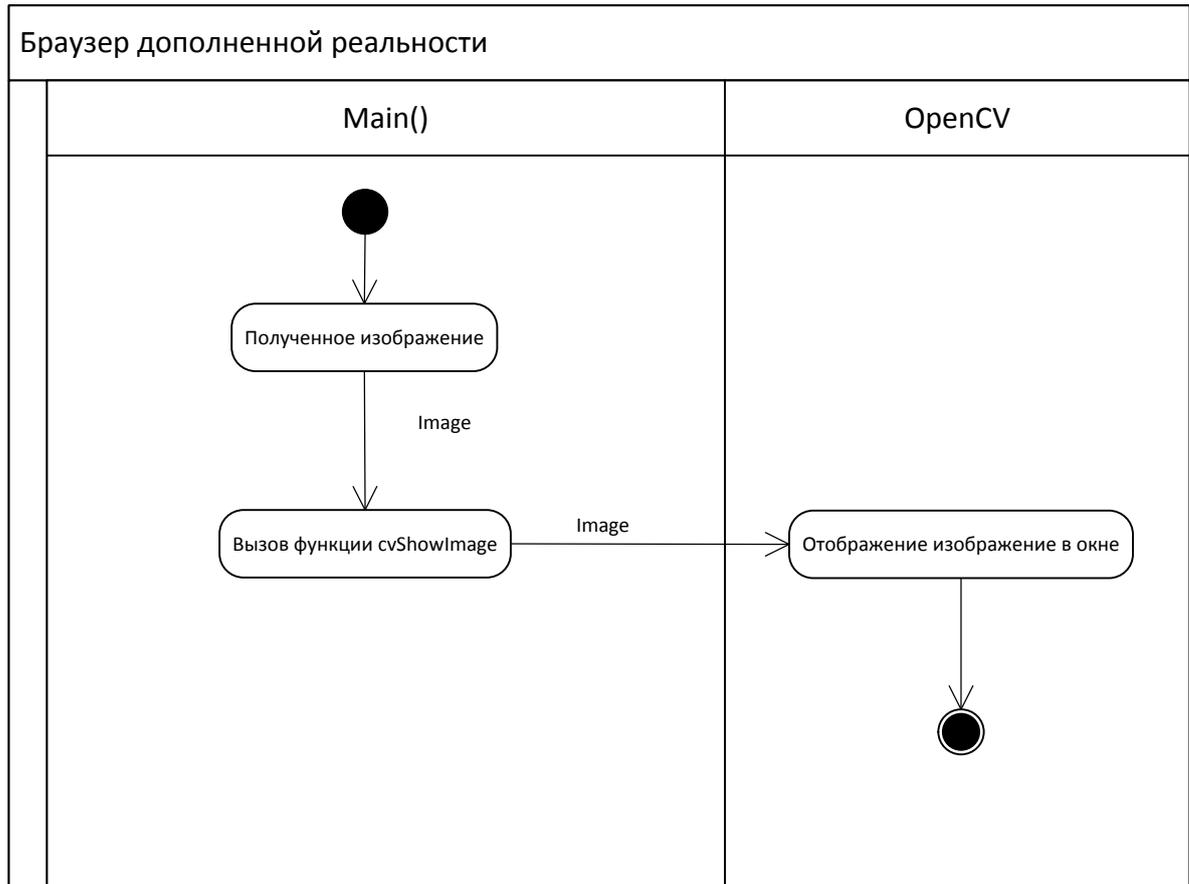


Рисунок 53. Диаграмма деятельности Отображение виртуального объекта

4.6 Выводы

В ходе данной главы была решена задача разработки библиотеки виртуальной реальности, был построен алгоритм воспроизведения дополненной реальности. Для решения задачи был рассмотрен алгоритмы действий для распознавания маркера:

1. Распознавание (обнаружения) маркера:
 - a. Приведение в градации серого
 - b. Бинаризация изображения (порог)
 - c. Определение замкнутых областей
 - d. Выделение контуров
 - e. Выделение углов маркера
 - f. Преобразование координат

2. Захват маркера
3. Построение 3D пространства
4. Отображение виртуального объекта

На основе этих действий программа выдает результат – отображает виртуальный объект на экране по обнаруженному маркеру.

ЗАКЛЮЧЕНИЕ

Разрабатываемая информационная система позволяет увеличить эффективность обучения, информативность печатной продукции и качество обучения.

При использовании печатной продукции не всегда можно легко показать изучаемый материал. Картинок может быть не достаточно, необходимо показать более подробно излагаемый материал: макеты, видео материалы. Рассмотренная в работе технология AR систем позволяют расширить возможности печатной продукции: выводить в ней видео и 3D модели.

В результате первой главы была изучена предметная область «Дополненная реальность», было рассмотрено ее применения, алгоритмы распознавание маркера дополненной реальности и построения объектов дополненной реальности.

На этапе концептуального проектирования были выделены основные сущности предметной области, построены модели данных: концептуальная, логическая в нотации IDEF1x и физическая модели данных.

Система должна позволять хранить и работать со следующей информацией:

- Пользователи (ФИО, права, логин, пароль)
- Права пользователей (название, описание)
- Учебный материал (заголовок, текст, пользователь, статус)
- Маркеры (название, тип, описание, дата создания, статус, файл контента, изображение, особые точки)
- Авторы (ФИО, описание)

Были выделены функциональные особенности системы:

1. Создание нового учебного материала
2. Загрузка изображения
3. Создание маркера

4. Создание авторов
5. Назначение авторов на учебный материал
6. Просмотр статистики
7. Добавление к маркеру содержимого: видео, 3D модель оборудования и т. д.
8. Добавление к маркеру учебного материала
9. Распознавание маркера
10. Воспроизведение содержимого маркера
11. Авторизоваться
12. Просмотр списка всех учебных материалов

В третьей главе была построена функциональная модель – диаграмма вариантов использования, показывающая взаимодействие пользователя с системой и написаны к ней спецификации. Построена модель архитектуры, в которой были выделены модули: камера, конвертер изображения, распознавание маркера, отрисовка объекта, код приложения, база маркеров. Был разработан интерфейс. Был произведен анализ существующих решений – рассмотрены AR библиотеки: OpenCV, Vuforia, ARToolKit, Metaio SDK, AForge.NET, Aurasma, Swarp SDK.

На основании требований к системе были подобраны средства разработки ПО: Microsoft Visual Studio 2013, язык программирования C++, СУБД MySQL, библиотека OpenCV, библиотека OpenGL. Были выделены роли пользователей («Администратор», «Студент») и соответствующие им права. Был описан механизм взаимодействия элементов системы во время записи на прием и добавления отзывов. Была построена модель архитектуры и интерфейс.

В четвертой главе был реализован браузер дополненной реальности, решающий задачи распознавания маркера, его захвата, построения 3D пространства и отображения виртуального объекта. На основе этих действий программа выдает результат, отображает его на экране в зависимости от распознанного маркера.

Применение дополненной реальности в обучение облегчает понимание сложных моделей, монтажной документации, инструкции по эксплуатации, чертежей и т. д. Проще показать механизм внутренних элементов компонентов. Дополненная реальность кроме того обеспечивает эффективный способ обучения персонала с целью сервиса и ремонта оборудования, различных установок или машин.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1) M. B. Hirokazu Kato, «Marker Tracking and HMD Calibration for a Video-Based Augmented Reality Conferencing System,» *Computer Society Washington*, 1999.
- 2) R. T. Azuma, «A Survey of Augmented Reality,» *Teleoperators and Virtual Enviroments*, 1997.
- 3) M. Tikander, Development and evaluation of augmented reality audio systems, Helsinki, 2009.
- 4) K. K. J. D. Joseph Rozier, «An Augmented Reality System of Linked Audio,» [В Интернетe]. Available: <http://www.icad.org/websiteV2.0/Conferences/ICAD2000/ICAD2000.html>. [Дата обращения: 2015].
- 5) К. М. России, Глобальная спутниковая навигационная система ГЛОНАСС. Интерфейсный кон- трольный документ, 2008.
- 6) «SLAM (метод),» [В Интернетe]. Available: https://ru.wikipedia.org/wiki/SLAM_%28%D0%BC%D0%B5%D1%82%D0%BE%D0%B4%29. [Дата обращения: 2015].
- 7) «Использование технологии дополненной реальности для бизнеса,» [В Интернетe]. Available: <http://appclub.im/news/details/10035>. [Дата обращения: 2015].
- 8) «Захват движения,» [В Интернетe]. Available: https://ru.wikipedia.org/wiki/%D0%97%D0%B0%D1%85%D0%B2%D0%B0%D1%82_%D0%B4%D0%B2%D0%B8%D0%B6%D0%B5%D0%BD%D0%B8%D1%8F. [Дата обращения: 2015].
- 9) «Optical Motion Capture Guide A Guide to Optical Motion Capture,» [В Интернетe]. Available: http://physbam.stanford.edu/cs448x/old/Optical_Motion_Capture_Guide.html. [Дата обращения: 2015].
- 10) «Модуль отслеживания 2D/3D объектов Intel Perceptual Computing SDK,» [В Интернетe]. Available: <http://www.intuit.ru/studies/courses/10619/1103/lecture/18232?page=3>. [Дата обращения: 2015].
- 11) «History of QR Code,» [В Интернетe]. Available: <http://www.qrcode.com/en/history/>. [Дата обращения: 2015].
- 12) В. В. Анна Дягтярева, «Преобразование Хафа,» [В Интернетe]. Available: <http://www.ict.edu.ru/ft/002407/num1degt.pdf>. [Дата обращения: 2015].
- 13) «Преобразование Хафа,» [В Интернетe]. Available: https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B5%D0%BE%D0%B1%D1%80%D0%B0%D0%B7%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5_%D0%A5%D0%B0%D1%84%D0%B0. [Дата

- обращения: 2015].
- 14) М. Антон, «Автоматическое выделение,» 27 май 2014. [В Интернете]. Available: <http://habrahabr.ru/post/224339/>. [Дата обращения: 2015].
 - 15) L. F. F. B. a. N. S. T. Hirata, «Fast QR Code Detection in Arbitrarily Acquired Images,» 2015. [В Интернете]. Available: http://sibgrapi.sid.inpe.br/col/sid.inpe.br/sibgrapi/2011/07.11.00.23/doc/8662_2_final.pdf.
 - 16) S. F. N. N. Xiang Zhang, «Visual Marker Detection and Decoding in AR Systems: A Comparative Study,» *Computer Society Washington*, 2002.
 - 17) «Scale-invariant feature transform,» [В Интернете]. Available: https://en.wikipedia.org/wiki/Scale-invariant_feature_transform. [Дата обращения: 2015].
 - 18) А. Бабий, «Применение SURF для создания маркера дополненной реальности,» 6 июнь 2012. [В Интернете]. Available: <http://habrahabr.ru/post/144845/>. [Дата обращения: 2015].
 - 19) «Построение SIFT дескрипторов и задача сопоставления изображений,» [В Интернете]. Available: <http://habrahabr.ru/post/106302/>. [Дата обращения: 2015].
 - 20) А. Бабий, «Распознавание маркера дополненной реальности,» 3 январь 2012. [В Интернете]. Available: <http://geektimes.ru/post/135659/>. [Дата обращения: 2015].
 - 21) К. В.А., «XI Международная научно-практическая конференция студентов, аспирантов и молодых ученых,» в *Алгоритм распознавания маркера дополненной реальности*, Томск.
 - 22) М. Sezgin и В. Sankur, «Survey over image thresholding techniques and quantitative performance evaluation,» *Journal of Electronic Imaging*, 2002.
 - 23) S. S. H. N. Ehsan Nadernejad, «Edge detection techniques: Evaluations and comparisons,» *Applied Mathematical Sciences*, 2008.
 - 24) «Computer Vision Algorithm,» [В Интернете]. Available: <http://www.hitl.washington.edu/artoolkit/documentation/vision.htm>. [Дата обращения: 2015].
 - 25) D. B. J. M. M. a. F. P. K. P. J. Crockett, «Automated detection and tracking of solar magnetic bright points,» май 2009. [В Интернете]. Available: <http://mnras.oxfordjournals.org/content/397/4/1852.full.pdf>. [Дата обращения: 2015].
 - 26) «Adaptive Thresholding,» [В Интернете]. Available: <http://homepages.inf.ed.ac.uk/rbf/HIPR2/adpthrsh.htm>.
 - 27) «Otsu's method,» [В Интернете]. Available: https://en.wikipedia.org/wiki/Otsu's_method. [Дата обращения: 2015].
 - 28) А. Иванов, «Алгоритмы заливки изображений, популярно и с видео,» [В Интернете]. Available: <http://habrahabr.ru/post/116374/>.
 - 29) Г. Алексей, «Алгоритмы заливки многоугольника,» [В Интернете].

- Available: <http://habrahabr.ru/post/116398/>. [Дата обращения: 2015].
- 30) X. S. a. J. G. Jianqiang Wang, «A Region Tracking-Based Vehicle Detection Algorithm in Nighttime Traffic Scenes,» 2013. [В Интернете]. Available: <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3892849/pdf/sensors-13-16474.pdf>. [Дата обращения: 2015].
 - 31) М. Л. М., Математические методы распознавания образов, Москва: МГУ, 2004.
 - 32) «Проекции векторов на прямую и на плоскость,» [В Интернете]. Available: <http://mathhelpplanet.com/static.php?p=proektsii-vektorov-na-pryamuyu-i-na-ploskost>. [Дата обращения: 2015].
 - 33) «Speeded up robust features,» [В Интернете]. Available: https://en.wikipedia.org/wiki/Speeded_up_robust_features. [Дата обращения: 2015].
 - 34) «Дополненная реальность – текущее состояние и тенденции развития,» [В Интернете]. Available: http://vmm.pstu.ru/index.php?option=com_content&view=article&id=524:2014-09-10-13-29-56&catid=41:scinews. [Дата обращения: 2015].
 - 35) «About StarUML,» [В Интернете]. Available: <http://staruml.sourceforge.net/en/>. [Дата обращения: 2015].
 - 36) [В Интернете]. Available: <http://www.visualstudio.com/downloads/download-visual-studio-vs>. [Дата обращения: 2014].
 - 37) «Метод Виолы-Джонса (Viola-Jones) как основа для распознавания лиц,» 2 декабрь 2011. [В Интернете]. Available: Метод Виолы-Джонса (Viola-Jones) как основа для распознавания лиц. [Дата обращения: 2015].
 - 38) л. к. р. в. I. Кирилл Корняков, «Краткая история проекта OpenCV,» [В Интернете]. Available: <http://habrahabr.ru/company/itseez/blog/146434/>. [Дата обращения: 2015].
 - 39) «OpenCV,» [В Интернете]. Available: <https://ru.wikipedia.org/wiki/OpenCV>. [Дата обращения: 2015].
 - 40) «ARToolKit,» [В Интернете]. Available: <https://en.wikipedia.org/wiki/ARToolKit>. [Дата обращения: 2015].
 - 41) «Metaio,» [В Интернете]. Available: <https://ru.wikipedia.org/wiki/Metaio>. [Дата обращения: 2015].
 - 42) «Metaio SDK: дополненная реальность стала полнее,» [В Интернете]. Available: <http://habrahabr.ru/company/intel/blog/198716/>. [Дата обращения: 2015].
 - 43) «Vuforia,» [В Интернете]. Available: <https://ru.wikipedia.org/wiki/Vuforia>. [Дата обращения: 2015].
 - 44) A. Bas, «Vuforia: немного магии в нашей реальности,» [В Интернете]. Available: <http://habrahabr.ru/post/198862/>. [Дата обращения: 2015].
 - 45) «List of Unified Modeling Language tools,» [В Интернете]. Available:

- http://en.wikipedia.org/wiki/List_of_UML_tools.
- 46) OpenCV шаг за шагом. Обработка изображения - пороговое преобразование <http://robocraft.ru/blog/computervision/357.html>. 2016 [Электронный ресурс].
 - 47) Фильтрация ключевых точек изображений. <http://www.arealidea.ru/articles/issledovanie-metodov-filtratsii-klyuchevykh-tochek-izobrazheniy-s-tselyu-povysheniya-kachestva-i-sko/>. 2016. [Электронный ресурс].
 - 48) Structural Analysis and Shape Descriptors OpenCV 2.4.12.0 documentation. http://docs.opencv.org/2.4/modules/imgproc/doc/structural_analysis_and_shape_descriptors.html?highlight=findcontours#findcontours. 2016. [Электронный ресурс].
 - 49) Miscellaneous Image Transformations. Docs opencv 2.4. http://docs.opencv.org/2.4/modules/imgproc/doc/miscellaneous_transformations.html. 2016. [Электронный ресурс].
 - 50) feature2d module. 2D Features framework. Docs opencv 2.4 . http://docs.opencv.org/2.4/doc/tutorials/features2d/table_of_content_features2d/table_of_content_features2d.html#table-of-content-feature2d/ 2016/ [Электронный ресурс].
 - 51) *imgproc* module. Image Processing. Docs opencv 2.4. 2016. http://docs.opencv.org/2.4/doc/tutorials/imgproc/table_of_content_imgproc/table_of_content_imgproc.html#table-of-content-imgproc. [Электронный ресурс].
 - 52) Андрей Бабий. Распознавание маркера дополненной реальности. <https://geektimes.ru/post/135659/>. 2012. [Электронный ресурс].
 - 53) Marker-based Augmented Reality on iPhone or iPad. <https://www.packtpub.com/books/content/marker-based-augmented-reality-iphone-or-ipad> [Электронный ресурс].
 - 54) Sobel Derivatives. http://docs.opencv.org/2.4/doc/tutorials/imgproc/imgtrans/sobel_derivatives/sobel_derivatives.html. 2016. [Электронный ресурс].
 - 55) Sobel Edge Detector. <http://homepages.inf.ed.ac.uk/rbf/HIPR2/sobel.htm>. 2016. [Электронный ресурс].
 - 56) Daniel Vaquero, Rogerio Feris, Matthew Turk, Ramesh Raskar. Multiflash Imaging and Applications. Daniel <http://ilab.cs.ucsb.edu/index.php/component/content/article/12/58>. 2016. [Электронный ресурс].
 - 57) Ramesh Raskar, Kar-Han Tan, Rogerio Feris, Jingyi Yu, Matthew Turk. Non-photorealistic Camera: Depth Edge Detection and Stylized Rendering using Multi-Flash Imaging. https://www.eecis.udel.edu/~jye/lab_research/SIG04/SIG_YU_RASKAR.pdf . 2016. [Электронный ресурс].

- 58) Shubham Saini, Bhavesh Kasliwal, Shraey Bhatia. COMPARATIVE STUDY OF IMAGE EDGE DETECTION ALGORITHMS. <http://arxiv.org/ftp/arxiv/papers/1311/1311.4963.pdf>. 2016. [Электронный ресурс].
- 59) Slawo Wesolkowski, Ed Jernigan. COLOR EDGE DETECTION IN RGB USING JOINTLY EUCLIDEAN DISTANCE AND VECTOR ANGL. <http://w3.gel.ulaval.ca/~parizeau/vi99/PDF-files/S1/paper57.pdf>. 2016. [Электронный ресурс].
- 60) Canny Edge Detection . http://docs.opencv.org/3.1.0/da/d22/tutorial_py_canny.html#gsc.tab=0. 2016. [Электронный ресурс].
- 61) Harvey Rhody. Lecture 15: Image Feature Detection. 2005.
- 62) Г. Буч, Д. Рамбо, А. Джекобсон. “Язык UML Руководство пользователя”

Спецификация диаграммы вариантов использования

Таблица 11. Управление пользователями

| | |
|-----------------------------------|--|
| Название прецедента использования | Управление пользователями |
| Цель прецедента использования | Управление данными о пользователях в системе |
| Главные актеры | Администратор, система |
| Предусловие | Администратор открыл страницу списка пользователей |
| Постусловие | Если прецедент был успешным, то система вносит данные в БД о пользователе (добавляет, удаляет, изменяет) |
| Основной поток | <p>Добавить пользователя:</p> <ol style="list-style-type: none"> Администратор нажимает кнопку «Добавить нового пользователя» Система открывает окно для добавления нового пользователя Администратор вводит ФИО (fio фамилия, имя, отчество), статус (status), должность, выбирает право, которые было подставлено из таблицы «Права» и подтверждает добавление (А, Б). Система добавляет данные о пользователе в таблицу «Пользователи» поля: ФИО, Статус, Должность, Права и открывает окно списка всех пользователей. <p>Изменить пользователя:</p> <ol style="list-style-type: none"> Администратор выбирает пользователя на странице всех пользователей Система открывает окно для изменения данных о пользователе Администратор изменяет поля и нажимает на сохранить (А, Б). Система сохраняет данные о пользователе в таблицу «Пользователи» поля: ФИО (фамилия, имя, отчество), Статус, Должность, Права и открывает окно списка всех пользователей. <p>Удалить пользователя:</p> <ol style="list-style-type: none"> Администратор выбирает пользователя на странице всех пользователей Нажимает на удалить (А) Система удаляет пользователя и все связанные с ним данные и открывает окно списка всех пользователей. |

| | |
|----------------------|--|
| | <p>Добавить право:</p> <ol style="list-style-type: none"> 1. Администратор нажимает клавишу «Добавить новое право» 2. Система открывает окно для добавления нового права 3. Администратор вводит название права и описание (не обязательно) и подтверждает добавление (В, Г). 4. Система добавляет данные о праве в таблицу «Права» поля: название (name), описание (description) и открывает окно списка всех прав. <p>Изменить право:</p> <ol style="list-style-type: none"> 1. Администратор выбирает право на странице всех прав 2. Система открывает окно для изменения данных о праве 3. Администратор изменяет поля название и описание (не обязательно) и нажимает на сохранить (В, Г). 4. Система сохраняет данные о праве в таблицу «Права» поля: название (name), описание (description) и открывает окно списка всех прав. <p>Удалить право:</p> <ol style="list-style-type: none"> 1. Администратор выбирает право на странице всех прав 2. Нажимает на удалить (В) 3. Система удаляет право и все связанные с ним данные и открывает окно списка всех прав. |
| Альтернативный поток | <p>А1. Администратор не подтвердил добавление (сохранение, удаление).</p> <p>Б1. Администратор не ввел обязательные поля (ФИО, Статус, Должность). Система вывела сообщение о том, что не все обязательные поля заполнены. (З).</p> <p>В1. Администратор не подтвердил добавление (сохранение, удаление).</p> <p>Г1. Администратор не ввел название права. Система вывела сообщение о том, что не все обязательные поля заполнены. (З).</p> |
| Точка расширения | Нет |
| Точка включения | Нет |

Таблица 12. Управление маркерами

| | |
|-----------------------------------|--|
| Название прецедента использования | Управление маркерами |
| Цель прецедента использования | Управление данными о маркерах, Связывает маркеры и учебные материалы |

| | |
|----------------|--|
| Главные актеры | Администратор, система |
| Предусловие | Администратор открыл страницу списка маркеров |
| Постусловие | Если прецедент был успешным, то система вносит данные в БД о маркере (добавляет, удаляет, изменяет) |
| Основной поток | <p>Добавить маркер:</p> <ol style="list-style-type: none"> 1. Администратор нажимает кнопку «Добавить маркер» 2. Система открывает окно для добавления маркера 3. Администратор вводит название, тип, описание (не обязательное), дата создания, статус, файл, изображения, файл контента и подтверждает добавление (А, Б). 4. Система приводит в градации серого, производит алгоритм нахождения особых точек изображения и вычисление дескрипторов особых точек. 5. Проверяет наличие такого маркера. (В) 6. Добавляет данные о маркере в таблицу «Маркеры». <p>Изменить маркер:</p> <ol style="list-style-type: none"> 1. Администратор выбирает маркер на странице всех маркеров 2. Система открывает окно для изменения данных о маркере 3. Администратор изменяет поля и нажимает на сохранить (А, Б). 4. Система приводит в градации серого, производит алгоритм нахождения особых точек изображения и вычисление дескрипторов особых точек. 5. Проверяет наличие такого маркера. (В) 6. Система сохраняет данные о маркере в таблицу «Маркеры». <p>Удалить маркер:</p> <ol style="list-style-type: none"> 1. Администратор выбирает маркер на странице всех маркеров 2. Нажимает на кнопку удалить (А) 3. Система удаляет пользователя и все связанные с ним данные и открывает окно списка всех маркеров. <p>Связывание маркера и учебного материала</p> <ol style="list-style-type: none"> 1. Администратор выбирает учебный материал 2. Система открывает окно для добавления маркеров к учебному материалу 3. Нажимает кнопку «Добавить маркеры» 4. Администратор выбирает маркеры из списка и нажимает кнопку сохранить (Г). <p>Система добавляет данные о маркерах к учебному материалу и</p> |

| | |
|----------------------|---|
| | открывает окно списка всех учебных материалов. |
| Альтернативный поток | <p>A1. Администратор не подтвердил добавление (сохранение, удаление).</p> <p>B1. Администратор не ввел обязательные поля (название, файл изображения, файл контента, тип). Система вывела сообщение о том, что не все обязательные поля заполнены. (3).</p> <p>B1. Такой маркер есть. (3).</p> <p>Г1 Администратор не подтвердил действие (2)</p> |
| Точка расширения | Нет |
| Точка включения | Нет |

Таблица 13. Управление учебным материалом

| | |
|-----------------------------------|--|
| Название прецедента использования | Управление учебным материалом |
| Цель прецедента использования | Управление данными об учебном материале и авторах |
| Главные актеры | Администратор, система |
| Предусловие | Администратор открыл страницу списка учебников |
| Постусловие | Если прецедент был успешным, то система вносит данные в БД об учебнике (добавляет, удаляет, изменяет) |
| Основной поток | <p>Добавить учебник:</p> <ol style="list-style-type: none"> 1. Администратор нажимает кнопку «Добавить новый учебник» 2. Система открывает окно для добавления нового учебника 3. Администратор вводит год издания, дату создания, статус, описание (не обязательно) и нажимает на сохранить(А, Б). 4. Система добавляет данные об учебном материале в таблицу «Учебный материал» и присваивает полю пользователь_id ключ текущего Администратор а и открывает окно списка всех учебников. <p>Изменить учебник:</p> <ol style="list-style-type: none"> 1. Администратор выбирает учебный материал на странице всех учебных материал 2. Система открывает окно для изменения данных об учебном материале 3. Администратор изменяет поля и нажимает на сохранить (А, Б). 4. Система сохраняет данные об учебнике в таблицу «Учебный материал» и открывает окно списка всех учебных |

| | |
|----------------------|---|
| | <p>материалов.</p> <p>Удалить учебный материал:</p> <ol style="list-style-type: none"> 1. Администратор выбирает учебный материал на странице всех учебных материалов 2. Нажимает на клавишу удалить (А) 3. Система удаляет учебный материал и все связанные с ним данные и открывает окно списка всех учебных материалов. <p>Добавить автора:</p> <ol style="list-style-type: none"> 1. Администратор нажимает кнопку «Добавить автора» 2. Система открывает окно для добавления нового автора 3. Администратор вводит ФИО (фамилия, имя, отчество), описание (не обязательно) и нажимает на клавишу сохранить (В, Г). 4. Система добавляет данные об авторе в таблицу «Авторы» и открывает окно списка всех авторов. <p>Изменить автора:</p> <ol style="list-style-type: none"> 1. Администратор выбирает автора на странице всех авторов 2. Система открывает окно для изменения данных об авторе 3. Администратор изменяет поля: ФИО, описание и нажимает на клавишу сохранить (В, Г). 4. Система сохраняет данные об авторе в таблицу «Авторы» и открывает окно списка всех авторов. <p>Удалить автора:</p> <ol style="list-style-type: none"> 1. Администратор выбирает учебный материал на странице всех авторов 2. Нажимает на клавишу удалить (В) 3. Система удаляет автора и все связанные с ним данные и открывает окно списка всех авторов. <p>Назначить автора учебному материалу</p> <ol style="list-style-type: none"> 1. Администратор выбирает учебный материал 2. Система открывает окно для добавления автора к учебному материалу 3. Нажимает кнопку «Добавить автора» 4. Администратор выбирает автора из списка и нажимает кнопку сохранить (А, Б). <p>Система добавляет данные об авторе к учебному материалу и открывает окно списка всех учебных материалов.</p> |
| Альтернативный поток | <p>А1. Администратор не подтвердил добавление (сохранение, удаление).</p> <p>Б1. Администратор не ввел обязательные поля.</p> |

| | |
|------------------|--|
| | Система вывела сообщение о том, что не все обязательные поля заполнены. (3). В1. Администратор не подтвердил добавление автора (сохранение, удаление). Г1. Администратор не ввел обязательные поля (ФИО). Система вывела сообщение о том, что не все обязательные поля (ФИО) заполнены. (3). |
| Точка расширения | Нет |
| Точка включения | Нет |

Таблица 14. Просмотреть содержимое маркера

| | | |
|-----------------------------------|---|--|
| Название прецедента использования | | Просмотреть содержимое маркера |
| Цель прецедента использования | | Просмотреть содержимое маркера |
| Главные актеры | | Пользователь, система |
| Предусловие | | Пользователь включил камеру и навел ее на учебный материал (маркер) |
| Постусловие | | Если прецедент был успешным, то система выведет на экран содержимое маркера (видео, 3D модель) |
| Основной поток | <ol style="list-style-type: none"> 1. Подключение камеры 2. Пользователь наводит камеру на изображение (маркер) 3. Система захватывает кадр с камеры - изображение 4. Приведение изображения в градацию серого 5. Нахождение особых точек на изображении 6. Вычисление дескрипторов особых точек 7. Преобразование координат 8. Поиск маркера в БД в соответствии с полученными координатами (A) 9. Построение 3D пространства 10. Отображение виртуального объекта | |
| Альтернативный поток | A1. Нет совпадения маркера (1) | |
| Точка расширения | | Нет |
| Точка включения | | Распознать маркер |

Таблица 15. Распознать маркер

| | |
|-----------------------------------|--|
| Название прецедента использования | Распознать маркер |
| Цель прецедента использования | Распознать маркер, для последующей визуализации на видео-потоке вместо маркера |

| | | |
|----------------------|---|---|
| | | медиа-контент |
| Главные актеры | | система |
| Предусловие | | Системе поступил на вход кадр с камеры |
| Постусловие | | Если прецедент был успешным, то система обнаружила изображение маркера на кадре, полученном с видеопотока |
| Основной поток | <ol style="list-style-type: none"> 1. Подключение камеры 2. Система захватывает видеопоток 3. Система получает кадр - изображение 4. Полученное изображение приводит в градацию серого 5. Находит на изображении особые точки 6. Вычисляет дескрипторы особых точек 7. Преобразует координаты 8. Ищет маркер в БД в соответствии с полученными координатами | |
| Альтернативный поток | | Нет |
| Точка расширения | | Нет |
| Точка включения | | Воспроизвести содержимое маркера |

Таблица 16. Воспроизвести содержимое маркера

| | | |
|-----------------------------------|--|---|
| Название прецедента использования | | Воспроизвести содержимое маркера |
| Цель прецедента использования | | Визуализации на видео-потоке содержимого маркера |
| Главные актеры | | система |
| Предусловие | | Координаты маркера |
| Постусловие | | Если прецедент был успешным, то система находит в БД маркер по координатам и отображает его |
| Основной поток | <ol style="list-style-type: none"> 1. Ищет маркер в БД в соответствии с полученными координатами (A) 2. Строит 3D пространство 3. Отображает виртуальный объект в соответствии с типом объекта, хранящимся в таблице (видео, 3D модель) | |
| Альтернативный поток | | A1. Маркер не найден. Действие останавливается |
| Точка расширения | | Нет |
| Точка включения | | Воспроизвести содержимое маркера |

Таблица 17. Просмотр статистики

| | |
|-----------------------------------|---------------------|
| Название прецедента использования | Просмотр статистики |
|-----------------------------------|---------------------|

| | |
|-------------------------------|---|
| Цель прецедента использования | Просмотреть основную статистику по количеству распознанных маркеров |
| Главные актеры | Администратор, система |
| Предусловие | Администратор открыл страницу просмотра статистики |
| Постусловие | Если прецедент был успешным, то выводит данные со статистикой за указанный период времени |
| Основной поток | <ol style="list-style-type: none"> 1. Администратор выбирает дату начала сбора статистики 2. Администратор выбирает дату окончания сбора статистики 3. Администратор выбирает опцию группировать по учебному материалу 4. Нажимает кнопку «Сформировать»(А) 5. Система формирует отчет по статистике распознавания маркеров за указанный период. Где выводится название маркера и количество раз, сколько этот маркер был распознан, за указанный период времени.(Б) 6. Если опция группировать по учебному материалу была выбрана, то выводится столбец учебный материал, и данные сгруппированы, по учебным материалам. 7. Внизу таблицы отчет по статистике распознавания маркеров можно вывести на печать или в Excel. |
| Альтернативный поток | <p>А1. Администратор не нажал клавишу сформировать отчет (2)</p> <p>Б1. Данные по статистике за указанные период пусты.</p> |
| Точка расширения | Нет |
| Точка включения | Нет |

Таблица 18. Авторизоваться

| | |
|-----------------------------------|--|
| Название прецедента использования | Авторизоваться |
| Цель прецедента использования | Авторизоваться в системе |
| Главные актеры | Пользователь, система |
| Предусловие | Пользователь открыл страницу авторизации |
| Постусловие | Если прецедент был успешным, то пользователь авторизовывается в системе |
| Основной поток | <ol style="list-style-type: none"> 1. Пользователь заполняет поле логин 2. Пользователь заполняет поле пароль 3. Нажимает клавишу «Войти» (А) 4. Нажимает кнопку «Сформировать»(А) 5. Система формирует запрос к базе данные, проверяет соответствие логина и пароля, если пара логин-пароль найдена, |

| | |
|----------------------|---|
| | то пользователь авторизуется в системе.(Б) |
| Альтернативный поток | А1. Администратор не нажал клавишу войти (2) Б1. Пара логин-пароль не найдены в базе, система выводит сообщение на страницу об ошибке введенных данных или о том, что такого пользователя нет в системе. (2) |
| Точка расширения | Нет |
| Точка включения | Нет |