

МИНОБРНАУКИ РОССИИ
федеральное государственное бюджетное
образовательное учреждение
высшего образования
«Московский государственный технологический
университет «СТАНКИН»
(ФГБОУ ВО «МГТУ «СТАНКИН»)

Институт
Информационных
и
Систем и
информатики в
Технологий
технических

Кафедра
Управления

системах

Старков Дмитрий Игоревич

РАЗРАБОТКА ИНТЕРАКТИВНОГО ФОТО-
КОМПЛЕКСА
С ВИРТУАЛЬНЫМИ ОБЪЕКТАМИ
ДОПОЛНЕННОЙ
РЕАЛЬНОСТИ

Выпускная квалификационная работа
по направлению 09.03. 04 «Прикладная информатика»

Регистрационный номер № _____

Заведующий кафедрой УИТС
проф.Олейник А.В.

Д.Т.Н.,

Научный руководитель
Елисеева Н.В.

К.Т.Н.,

Студент

Старков Д.И.

Москва 2019 г.

Аннотация
на выпускную квалификационную работу студента
Московского Государственного Технологического
Университета
«СТАНКИН»

Группа ИДБ-15-15

Старков Дмитрий Игоревич

Тема: «Разработка интерактивного фото-комплекса с виртуальными объектами дополненной реальности».

В данной выпускной квалификационной работе был спроектирован интерактивный фото-комплекс с виртуальными трехмерными объектами дополненной реальности, проецирующий объекты виртуальной реальности на видеопоток, снятый камерами корпуса. Комплекс предназначен для привлечения к себе внимания посетителей мероприятий и повышения их лояльности к компании, а также для повышения известности компании.

В первой главе были исследованы и проанализированы возможные аппаратные решения для реализации комплекса и алгоритмы его функционирования, было написано техническое задание на реализацию комплекса с описанием требований к функциональности, безопасности, надежности и т. д.

Во второй главе были спроектированы аппаратная, программная и информационная архитектуры комплекса, были детально описаны выбранные методы реализации функций и архитектура комплекса технических средств комплекса. Были спроектированы алгоритмы работы пользователя с устройством.

В третьей главе была описана реализация отдельных модулей программного обеспечения комплекса, отвечающих за проецирование трехмерного виртуального объекта на

изображение реальности. Были описаны используемые программные инструменты и обоснован их выбор.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	4
1. АНАЛИЗ ТЕХНОЛОГИЙ И ПРОДУКТОВ.....	7
1.1 АНАЛИЗ МЕТОДОВ РЕАЛИЗАЦИИ ИНТЕРАКТИВНОГО ФОТО-КОМПЛЕКСА С ВИРТУАЛЬНЫМИ ОБЪЕКТАМИ ДОПОЛНЕННОЙ РЕАЛЬНОСТИ.....	7
1.2 АНАЛИЗ ИНСТРУМЕНТАЛЬНЫХ СРЕДСТВ РЕАЛИЗАЦИИ ИНТЕРАКТИВНОГО ФОТО-КОМПЛЕКСА С ВИРТУАЛЬНЫМИ ОБЪЕКТАМИ ДОПОЛНЕННОЙ РЕАЛЬНОСТИ.....	9
1.2.1 ВЫБОР ЦЕНТРАЛЬНОГО АППАРАТНОГО УСТРОЙСТВА.....	9
1.2.2 ВЫБОР ФОТО-КАМЕР.....	13
1.2.3 ПРОЕКТИРОВАНИЕ КОРПУСА МУЛЬТИМЕДИЙНОГО УСТРОЙСТВА.....	15
1.2.4 ВЫБОР ОПЕРАЦИОННОЙ СИСТЕМЫ.....	16
1.3 АНАЛИЗ МЕТОДОВ СОВМЕЩЕНИЯ ВИРТУАЛЬНЫХ И РЕАЛЬНЫХ ОБЪЕКТОВ В СИСТЕМАХ ДОПОЛНЕННОЙ РЕАЛЬНОСТИ.....	18
1.4 РАЗРАБОТКА ТЕХНИЧЕСКОГО ЗАДАНИЯ НА РЕАЛИЗАЦИЮ СИСТЕМЫ.....	20
1.4.1 ОБЩИЕ СВЕДЕНИЯ.....	20
1.4.2 НАЗНАЧЕНИЕ И ЦЕЛИ СОЗДАНИЯ(РАЗВИТИЯ) СИСТЕМЫ.....	20
1.4.3 ХАРАКТЕРИСТИКИ ОБЪЕКТА АВТОМАТИЗАЦИИ.....	20
1.4.4 ТРЕБОВАНИЯ К СИСТЕМЕ.....	20
1.4.5 СОСТАВ И СОДЕРЖАНИЕ РАБОТ ПО СОЗДАНИЮ(РАЗВИТИЮ) СИСТЕМЫ.....	23
1.4.6 ПОРЯДОК КОНТРОЛЯ И ПРИЕМКИ СИСТЕМЫ.....	24
1.4.7 ТРЕБОВАНИЯ К ДОКУМЕНТИРОВАНИЮ.....	24
2. ПРОЕКТИРОВАНИЕ ИНТЕРАКТИВНОГО ФОТО-КОМПЛЕКСА С ВИРТУАЛЬНЫМИ ОБЪЕКТАМИ ДОПОЛНЕННОЙ РЕАЛЬНОСТИ.....	25
2.1 РАЗРАБОТКА МЕТОДА ПРОЕКЦИРОВАНИЯ ВИРТУАЛЬНОГО ТРЕХМЕРНОГО ОБЪЕКТА ДОПОЛНЕННОЙ РЕАЛЬНОСТИ.....	25
2.2 ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ МОДЕЛИ ИНТЕРАКТИВНОГО ФОТО-КОМПЛЕКСА.....	31
2.3 ПРОЕКТИРОВАНИЕ АРХИТЕКТУРЫ КОМПЛЕКСА ТЕХНИЧЕСКИХ СРЕДСТВ ДЛЯ РЕАЛИЗАЦИИ ПРОЦЕДУРЫ ФОТОГРАФИРОВАНИЯ.....	33
2.4 ПРОЕКТИРОВАНИЕ АЛГОРИТМА ПОВЕДЕНИЯ СИСТЕМЫ: ФОТОГРАФИРОВАНИЕ С ВИРТУАЛЬНЫМ ТРЕХМЕРНЫМ ОБЪЕКТОМ ДОПОЛНЕННОЙ РЕАЛЬНОСТИ.....	37
3. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ ИНТЕРАКТИВНОГО ФОТО-КОМПЛЕКСА С ВИРТУАЛЬНЫМИ ОБЪЕКТАМИ ДОПОЛНЕННОЙ РЕАЛЬНОСТИ.....	38
3.1 ВЫБОР ИНСТРУМЕНТОВ ДЛЯ РАЗРАБОТКИ.....	38

3.2 МОДУЛЬ РАСПОЗНАВАНИЯ РУК.....	40
3.3 МОДУЛЬ ПОСТРОЕНИЯ КАРТ ГЛУБИН.....	43
3.4 РАЗРАБОТКА ПРОЕКТИРОВАНИЯ ВИРТУАЛЬНОГО ТРЕХМЕРНОГО ОБЪЕКТА ДОПОЛНЕННОЙ РЕАЛЬНОСТИ.....	45
3.5 РЕАЛИЗАЦИЯ АЛГОРИТМА ПОВЕДЕНИЯ СИСТЕМЫ: ФОТОГРАФИРОВАНИЕ С ВИРТУАЛЬНЫМ ТРЕХМЕРНЫМ ОБЪЕКТОМ ВИРТУАЛЬНОЙ РЕАЛЬНОСТИ.....	48
СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ.....	52

ВВЕДЕНИЕ

В наши дни проводится большое число разнообразных выставок технических, - и не только, - достижений. На них разработчики ищут себе инвесторов или рекламируют свой продукт публике. На выставках обычно присутствует не одна компания. Посетители перемещаются от стенда к стенду, рассматривая представленные экспонаты. Одной из важных задач для компании на подобном мероприятии — привлечение потока посетителей к своему стенду для максимально большого охвата аудитории. Чем больше людей пройдет через стенд компании за время мероприятия — тем эффективней данная маркетинговая стратегия, тем больше потенциальных продаж/заказов, тем больше прибыль компании. Для этого используются различные привлекающие методы: яркие цвета, аудио-информирование, промоутеры, раздача сувенирной продукции и т.д.. Все они так или иначе повышают информирование посетителей о стендах компании. Однако, в условиях конкуренции, большинство присутствующих на мероприятии уже используют подобные средства. Компании становится труднее завлекать к себе новых людей т.к. остальные занимаются тем же. В то же время, многие способы привлечения внимания теряют свою эффективность со временем. Возникают эффекты вроде «баннерной слепоты»^[1] и т. д., конверсия снижается. Таким образом, разработка новых способов управления людскими потоками является актуальной темой сегодняшнего времени.

Помимо перечисленных способов, интерактивные стенды с элементом развлечения также привлекают к себе внимание. Именно это и предлагается использовать. Разрабатываемый комплекс должен привлекать к себе посетителей и повышать их интерес к компании в целом и мотивацию ознакомиться с представленной продукцией. Для этого фото-комплекс должен обладать развлекающей, интерактивной и инновационной составляющей. Он должен вовлекать пользователей во взаимодействие с компанией «вознаграждая» их за это. Иначе, велика вероятность, что он будет проигнорирован ими, как очередной рекламный щит. Для этого предполагается следующая схема взаимодействия:

- пользователь подходит к комплексу, на экране появляется видео-поток с фронтальной камеры комплекса, на котором видно пользователя;
- пользователь выбирает трехмерный объект дополненной реальности, символизирующий компанию или её продукцию;
- выбранный объект вставляется в видео-поток, пользователь задает расположение объекта(предположительно, манипулируя заранее заданным реальным предметом или жестами);
- происходит съемка фотографии, которая отображается на экране вместе с вставленным в нужной позиции объектом;
- пользователю предлагается ввести контакты для отправки фотографии.

Возможно использование средств компьютерного зрения для классификации пользователей для рекомендаций объектов дополненной реальности. Также можно предлагать пользователю использовать введенный контакт для подписки на информационные материалы компании. Необходимо также разработать универсальный интерфейс для компании для экспорта своих объектов в комплекс.

Итоговой целью является разработка универсального интерактивного фото-комплекса с виртуальными объектами дополненной реальности для использования в маркетинговых целях в качестве объекта привлечения внимания людей на массовых мероприятиях.

Для выполнения данной цели необходимо будет решить следующие задачи:

- Анализ
 - Анализ методов и инструментальных средств анализа видео-потока
 - Анализ методов совмещения виртуальных и реальных объектов в системах дополненной реальности
 - Описание предметной области и выбор технологии обучения системы распознаванию посетителей по типам: чиновник, специалист, ученый, студент, школьник
 - Разработка технического задания на реализацию системы
- Разработка технического задания на реализацию системы
- Проектирование

- Разработка метода проецирования виртуального трехмерного объекта дополненной реальности
- моделирование паттернов трехмерных объектов для использования в комплексе
- проектирование архитектуры комплекса технических средств для реализации процедуры фотографирования
- проектирование алгоритма поведения системы: фотографирование с виртуальным трехмерным объектом дополненной реальности
- Реализация
 - Разработка интерфейсов
 - Разработка модулей и архитектуры
 - Разработка методики инсталляции и эксплуатации
 - Документирование системы
 - Апробация системы на примере
- Тестирование

Создание подобного комплекса позволит компаниям использовать его на различных выставках, презентациях и прочих мероприятиях для привлечения внимания к себе и своим продуктам со стороны потенциальных клиентов.

Для этого необходимо будет изучить и проанализировать существующие и доступные аппаратные и программные решения различных частей комплекса, выбрать наиболее подходящие и объединить их в единую систему. Помимо этого необходимо провести анализ требований к интерфейсу со стороны пользователей.

1. АНАЛИЗ ТЕХНОЛОГИЙ И ПРОДУКТОВ

1.1 АНАЛИЗ МЕТОДОВ РЕАЛИЗАЦИИ ИНТЕРАКТИВНОГО ФОТО-КОМПЛЕКСА С ВИРТУАЛЬНЫМИ ОБЪЕКТАМИ ДОПОЛНЕННОЙ РЕАЛЬНОСТИ

Различные компании предлагают аренду или продажу готовых решений интерактивных комплексов. К преимуществам использования готового решения можно отнести отсутствие необходимости в проектировании и изготовлении корпуса, выборе аппаратных комплектующих и низкоуровневой настройке ПО. Большинство готовых решений поставляется с предустановленной операционной системой и набором программного обеспечения, упрощающего настройку комплекса. Многие из них поставляются со встроенным сенсорным экраном. Некоторые снабжены видео-камерой.

Недостатком готовых решений является отсутствие таких с наличием двух камер для проведения стерео-съемки. Для решения данной проблемы можно использовать специальный объектив, позволяющий получать стерео-изображение одной камерой. Благодаря системе зеркал, две половины камеры снимают изображение из двух, разведенных в стороны, точек.

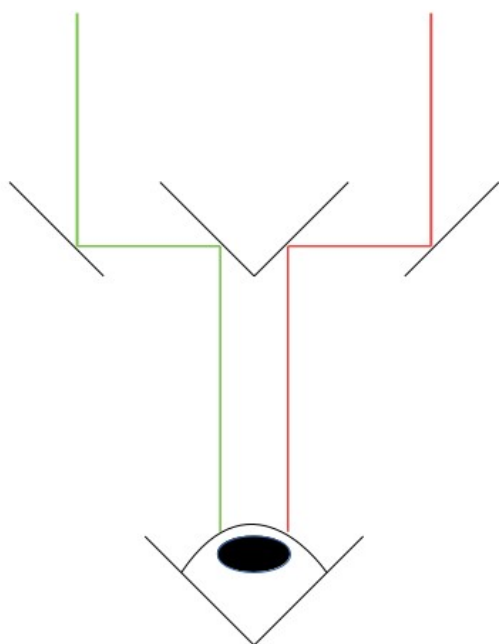


Рис.1.1.1: схема работы объектива для стерео-съемки.

Однако использование подобной технологии связано с проблемой отсутствия готовых решений для специфичных камер. Объективы подобного типа выпускаются для различных фото- и видео-камер, но не для веб-камер. Для последних можно изготовить объектив на заказ. Небольшие отклонения в углах расположения зеркал способны сильно исказить получаемый результат, а потому требования к точности выполнения таких работ будут большими. В связи с этим, стоимость такого решения может быть слишком большой.

Еще одним недостатком использования подобного решения является необходимость встраивания его в готовый комплекс таким образом, чтобы исключить возможность смещения объектива при активном использовании комплекса в местах массового скопления людей.

Другим решением является изготовление собственного комплекса с учетом всех необходимых деталей. Это позволит использовать две камеры, синхронизированные между собой для осуществления стерео-съемки.

Недостатком такого выбора является необходимость проектирования комплекса с нуля: его аппаратных и программных компонентов и их тщательная настройка. Для работы комплекса понадобятся: центральное аппаратное устройство — устройство, производящее необходимые вычисления, обрабатывающее сигналы устройств ввода и выводящее результаты, используя устройства вывода; камеры; устройство ввода для пользователя, которое, при использовании сенсорного экрана, также будет являться устройством вывода; корпус комплекса.

В данной работе будет рассмотрен последний вариант — изготовление собственного комплекса.

1.2 АНАЛИЗ ИНСТРУМЕНТАЛЬНЫХ СРЕДСТВ РЕАЛИЗАЦИИ ИНТЕРАКТИВНОГО ФОТО-КОМПЛЕКСА С ВИРТУАЛЬНЫМИ ОБЪЕКТАМИ ДОПОЛНЕННОЙ РЕАЛЬНОСТИ

1.2.1 ВЫБОР ЦЕНТРАЛЬНОГО АППАРАТНОГО УСТРОЙСТВА

В настоящее время на рынке доступно множество решений для автономных комплексов. В данном случае будут рассматриваться варианты сборки центрального устройства из модулей или использования готовых плат.

С точки зрения бюджета — покупка необходимых модулей для самостоятельной сборки аппаратной части дороже, чем приобретение готового микрокомпьютера. Также ручная сборка может потребовать дополнительных навыков. Однако микрокомпьютеры не отличаются высокими вычислительными мощностями, которые могут потребоваться для анализа видеопотока в режиме реального времени.

Разбор наиболее известных вариантов микрокомпьютеров:

1. Raspberry Pi 3 Model A+^[3]:

Одноплатный микрокомпьютер с заявленной спецификацией:

Процессор: Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz;

Оперативная память: 512MB LPDDR2 SDRAM;

Поддержка подключения к беспроводной сети;

HDMI-выход;

Один порт USB 2.0;

Один CSI вход для подключения Raspberry Pi Camera Module;

Порт MicroSD для загрузки операционной системы;

Питание: 5 В/2.5 А.

Данная модель не подойдет под цели проекта в виду того, что обладает малым объемом оперативной памяти и всего одним USB портом.

2. Raspberry Pi 3 Model B+^[4]:

Усовершенствованная версия предыдущей модели.

Спецификация:

Процессор: Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz;

Оперативная память: 1GB LPDDR2 SDRAM;

Поддержка подключения к беспроводной сети или к проводной через вход USB 2.0(максимальная пропускная способность — 300 Мбит/с);

HDMI-выход;

4 порта USB 2.0;

Один CSI вход для подключения Raspberry Pi Camera Module;

Порт MicroSD для загрузки операционной системы;

Питание: 5 В/2.5 А.

Число портов USB удовлетворяет условиям. Объем оперативной памяти вдвое выше, но её тип - LPDDR2 SDRAM может замедлять работу системы. Отсутствие GPU сильно повышает нагрузку на процессор, что также может быть причиной медленной работы.

3. Orange Pi Pc Plus^[5]:

Одноплатный микрокомпьютер с открытыми схемами.

Спецификация:

Процессор: H3 Quad-core Cortex-A7 H.265/HEVC 4K;

GPU: Mali400MP2 GPU @600MHz;

Оперативная память: 1GB DDR3 (общая с GPU);

Слот под TF/MMC карту памяти;

Возможность проводного подключения через порт Ethernet;

Один CSI вход;

HDMI выход;

Три порта USB 2.0 HOST и один USB 2.0 OTG;

Питание: 5 В/2 А;

Несмотря на чуть менее слабый процессор^[13], чем у Raspberry Pi 3 Model B+, наличие GPU может позволить всей системе работать быстрее. Тип оперативной памяти DDR3 работает быстрее. Число USB портов удовлетворяет требованиям.

4. Banana Pi BPI-F2^[6]:

Одноплатный компьютер с открытыми схемами.

Спецификация:

Процессор: ARM Cortex-A9 MPCore;

GPU: OpenGL ES 2.0 3D графический ускоритель;

Оперативная память: 2GB DDR3

Поддержка карт MicroSD, TF и eMMC;

Порт Mini PCI;

Интерфейс SATA;

Поддержка подключения проводной сети через порт Ethernet;

Порт HDMI;

Видео-входы: 1 DVP и 1 MIP;

2 USB 2.0 HOST порта и 1 USB OTG;

Питание: 12 В/2 А.

Значительно более мощная модель. Однако, при подключении камер через порты USB — их может не хватить для подключения сенсорного экрана. Подключение же камер к портам DVP и MIP может вызвать конфликты из-за использования разных интерфейсов.

5. StereoPi^[7]:

Плата-расширение для одноплатных компьютеров Raspberry Pi Compute Module 3, Raspberry Pi Compute Module 3 Lite, Raspberry Pi Compute Module 1, Raspberry Pi Compute Module 3+ и Raspberry Pi Compute Module 3+ Lite разработанная специально для подключения двух камер. Спецификация:

2 входа CSI для подключения камер;

2 USB порта;

Ethernet-порт;

HDMI выход;

Питание: 5В;

Поддерживает подключение камер сторонних производителей.

К данному решению необходима одна из перечисленных ранее моделей Raspberry Pi Compute Module. Спецификации:

CM3^[8]:

Процессор: четырехядерный BCM2837^[12](основан на кластере ARM Cortex A-53);

Оперативная память: 1GB;

4GB eMMC Flash — устройство;

CM3+^[9]:

Процессор: четырехядерный BCM2837(основан на кластере ARM Cortex A-53);

Оперативная память: 1GB;

8/16/32GB eMMC Flash — устройство в зависимости от комплектации;

CM3+ Lite^[10]:

Процессор: четырехядерный BCM2837(основан на кластере ARM Cortex A-53);

Оперативная память: 1GB;

Из данного списка наиболее предпочтительным вариантом является модель StereoPi в связке с Raspberry Pi Compute Module 3+. Подключение обеих камер через вход CSI напрямую к CPU заметно ускорит работу системы. Четырехядерный процессор CM3 позволит плате работать в 10 раз быстрее, чем аналогичная система на базе CM1. Использование CSI-входов также снимает требования на число USB-портов т. к. они не будут задействованы для подключения камер. Возможность использования камер сторонних производителей дает возможность подобрать камеры, имеющие необходимые характеристики. Отсутствие GPU снизит скорость работы некоторых аспектов системы, но общая скорость работы возрастет за счет использования CSI-камер, вместо USB-камер.

Таблица 1.2.1

Микро компьютер/ характеристика	Raspberry Pi 3 Model A+	Raspberry Pi 3 Model B+	Orange Pi Pc Plus	Banana Pi BPI-F2	Stereo Pi + Raspberry Pi Compute Module 3	Stereo Pi + Raspberry Pi Compute Module 3+	Stereo Pi + Raspberry Pi Compute Module 3+ Lite
Процессор	Cortex-A53, 1.4 ГГц	Cortex-A53, 1.4 ГГц	Cortex-A7	Cortex-A9	BCM 2837	BCM 2837	BCM 2837
Оперативная память	512MB LPDDR2	1GB LPDDR2	1GB DDR3	2GB DDR3	1GB LPDDR2	1GB LPDDR2	1GB LPDDR2
Беспроводное подключение	Возможно	Возможно	Возможно	Отсутствует	Отсутствует	Отсутствует	Отсутствует

Порт HDMI	Имеется	Имеется	Имеется	Имеется	Имеется	Имеется	Имеется
Порт USB	USB 2.0 — 1 шт.	USB 2.0 — 4 шт.	USB 2.0 HOST — 3 шт., OTG — 1 шт.	USB 2.0 HOST — 2 шт., OTG — 1 шт.	USB 2.0 — 2 шт.	USB 2.0 — 2 шт.	USB 2.0 — 2 шт.
Вход CSI	1 шт.	1 шт.	1 шт.	Отсутствует	2 шт.	2 шт.	2 шт.
GPU	Отсутствует	Отсутствует	Mali400 MP2 GPU @600M Hz	OpenGL ES 2.0 3D графический ускоритель	Отсутствует	Отсутствует	Отсутствует

Зеленым цветом отмечены показатели характеристик устройства, являющиеся его преимуществами. Красным — недостатками. Желтым — нейтральные показатели.

1.2.2 ВЫБОР ФОТО-КАМЕР

Выбор камер для комплекса — важный этап. Т.к. планируется использование трехмерных моделей — может потребоваться использование двух камер для определения глубины на кадрах видео-потока:

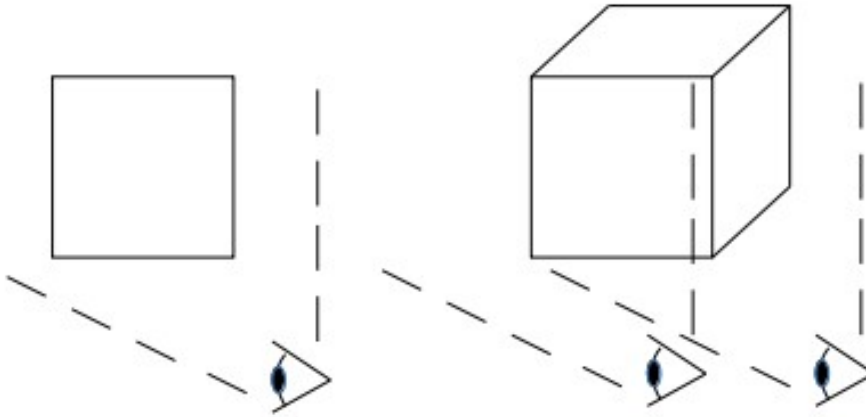


Рис. 1.2.1

Выбранное центральное устройство имеет два варианта использования камеры: через вход CSI и через USB порт. Первый вариант работает заметно быстрее т. к. видео-поток транслируется напрямую в CPU, избегая посредников. Также, большинство USB-камер предназначены для съемки полноценного видео и оснащены микрофоном, который не будет использован в аппарате, однако его стоимость прибавляется к общей стоимости камеры. Таким образом, предпочтительным способом подключения камер с учетом наличия двух CSI входов на выбранном центральном устройстве будет именно CSI.

Выбор камеры также зависит от необходимого разрешения. Чем больше разрешение камеры — тем качественней будет выходящее изображение. Однако слишком большое разрешение вызовет большую нагрузку на систему обработки, что замедлит её вплоть до визуальных зависаний.

Зависимости качества изображения и производительности системы от разрешения используемых камер выглядит приблизительно следующим образом:



Рис. 1.2.2

Возможность использования камер от сторонних производителей позволяет выбирать камеры с различными характеристиками и особенностями такими, как наличие широкоугольного объектива или линзы «рыбий глаз», возможность регулировки фокусного расстояния и ночной съемки. Однако для целей комплекса перечисленные особенности не нужны.

Возможные варианты камер:

1. Raspberry Pi Camera Module. CSI-камера с сенсором 5 Мп. Поддержка разрешений: 1080p(full HD) 30 кадров в секунду, 720p 60 кадров в секунду.

2. Raspberry Pi Camera Module v2^[14]. CSI-камера, оснащенная 8-Мп сенсором. Поддерживает разрешения: 1080p(full HD) 30 кадров в секунду, 720p 60 кадров в секунду.

3. RPi Camera (D) WaveShare^[15]. CSI-камера с сенсором 5 Мп. Поддержка разрешений: 1080p(full HD) 30 кадров в секунду, 720p 60 кадров в секунду.

Предпочтение стоит отдать модели Raspberry Pi Camera Module v2. При прочих равных эта модель обладает сенсором 8 Мп, что позволит делать более качественные снимки для отображения на экране. Стоимость её при этом выше.

1.2.3 ПРОЕКТИРОВАНИЕ КОРПУСА МУЛЬТИМЕДИЙНОГО УСТРОЙСТВА

К корпусу комплекса существует несколько требований: его габариты не должны быть слишком большими для нахождения его на стенде компании, монитор должен находиться на уровне глаз пользователя, камеры должны смотреть по направлению к пользователю, корпус должен быть устойчивым.

Корпус должен соответствовать эргономическим нормам. У пользователя должна быть возможность наблюдения за происходящим на экране с расстояния 2м от него.

Внутри корпуса должны быть предусмотрены крепления для центрального устройства, экрана и камер. Все элементы должны быть прочно закреплены. Экран должен иметь опору сзади во избежание его выдавливания внутрь при эксплуатации.

Центр тяжести корпуса должен находиться внизу за счет использования утяжелителей. Это повысит устойчивость.

При выборе типа используемых материалов стоит рассматривать следующие варианты:

1. Дерево. Основными преимуществами являются дешевизна, простота обработки и экологичность. К недостаткам можно отнести недолговечность, малый запас прочности, ограниченность в типах обработки, пожароопасность, возможность скалывания кусков в процессе эксплуатации.

В виду того, что комплекс при эксплуатации будет часто транспортироваться в собранном виде и должен быть рассчитан на частое использование, данный тип материала не подходит из-за своих прочностных характеристик.

2. Пластик. Существует огромное количество видов пластика, которые существенно отличаются друг от друга стоимостью, сложностью и стоимостью обработки и прочностью. В большинстве случаев не экологичны.

Отдельные распространенные виды данного типа материалов могут быть использованы в задаче т. к. обладают достаточными прочностными характеристиками, имеют удовлетворительную цену и имеется возможность обработать их, создав корпус нужной формы. При выборе в таком случае стоит обратить внимание на пожароопасность и токсичность отдельных его видов.

3. Металл. А именно различные сорта стали. Дороги, сложны в обработке, имеют высокую массу, подвержены коррозии. Не горят, обладают отличными прочностными характеристиками, долговечны.

Отлично подходят для решения задачи благодаря своим характеристикам.

1.2.4 ВЫБОР ОПЕРАЦИОННОЙ СИСТЕМЫ

На выбранное центральное устройство можно установить операционную систему с поддержкой архитектуры процессора ARM. Семейство операционных систем windows рассматривать нецелесообразно из-за высокой стоимости, высокого потребления ресурсов, малой гибкости и отсутствия преимуществ перед другими ОС.

Ниже будут рассмотрены открытые операционные системы, поддерживающие архитектуру ARM^[16].

Android — операционная система с открытым исходным кодом, созданная на базе ядра Linux. Распространяющаяся по лицензии Apache License 2.0 (GNU GPL v2 для модификаций ядра Linux). Лицензия позволяет использовать данную систему в коммерческих целях бесплатно, не предупреждая авторов. Сильно ограничивает в инструментах разработки.

Lubuntu — дистрибутив Linux, производный от Ubuntu. Использует легковесное окружение рабочего стола LXDE. Нетребователен к ресурсам. Распространяется по лицензии GNU GPL, что позволяет использовать его в коммерческих целях свободно. По-умолчанию распространяется с системой управления пакетами apt, позволяющей легко получать обновления необходимых программ.

Debian — дистрибутив GNU/Linux. Распространяется по лицензии GNU GPL. Имеет наибольшее из всех дистрибутивов хранилище пакетов — готовых к использованию программ и библиотек. Использует систему управления пакетами apt. Однако обновление пакетов в стабильной ветке системы происходит лишь для устранения уязвимостей и при выпуске новой версии дистрибутива.

Raspbian — свободный дистрибутив на базе Debian, оптимизированный для работы на устройствах Raspberry Pi. Также использует систему управления пакетами apt.

Armbian — свободный дистрибутив на базе Debian, разработанный специально для процессоров на архитектуре ARM. Использует систему управления пакетами apt.

ArchLinux ARM — дистрибутив Linux, распространяющийся по лицензии GPLv.2 (любые модификации должны распространяться с такой же лицензией). Использует модель rolling release, позволяющую получать новые версии пакетов практически сразу, после их появления. Данная особенность иногда приводит к нестабильной работе в виду пришедших обновлений. Позволяет самостоятельно настроить операционную систему, минимизируя потребление ресурсов. Оптимизирован для архитектуры ARM. Несмотря на повышенную сложность использования (используется принцип KISS — Keep It Simple, Stupid — запрет использования более сложных средств, чем необходимо, выливающийся в необходимость знания низкоуровневых аспектов операционной системы для работы с ней), данный дистрибутив характеризуется отличной гибкостью.

Из данного списка наиболее подходящим вариантом является Raspbian. Данный дистрибутив оптимизирован под использование на конкретных процессорах и архитектуре плат Raspberry Pi. Он легковесен и потребляет малое число ресурсов. С 2014 года в нем появилась поддержка использования стерео-режима для нескольких камер. Открытые исходные коды и полная свобода настройки позволят отключить не используемые модули ОС, при желании, и, что куда более важно, удалить не используемое пользователем, но нагружающее систему, ПО, предустановленное по умолчанию.

1.3 АНАЛИЗ МЕТОДОВ СОВМЕЩЕНИЯ ВИРТУАЛЬНЫХ И РЕАЛЬНЫХ ОБЪЕКТОВ В СИСТЕМАХ ДОПОЛНЕННОЙ РЕАЛЬНОСТИ

Существуют разные методы обеспечения дополненной реальности. Большинство из них делятся на следующие этапы:

1. Поиск точек интереса, координатных меток и оптических потоков на изображении с камеры. На данном этапе используются такие способы, как: обнаружение углов, контрастных регионов и контуров, отсеивание и другие способы обработки изображений. Т.к. в качестве ключевых объектов для комплекса будут выступать человеческие руки, не обладающие строгой геометрией и постоянством формы и цвета, то предполагается использование методов машинного обучения для тренировки модели для дальнейшего обнаружения объектов на кадрах видео-потока.
2. Восстановление мировых координат по полученным на первом шаге данным. Данная задача подразделяется на 3 типа:
 - Трехмерная структура сцены известна, на изображении присутствуют координатные метки.
 - Часть сцены неизвестна. В таком случае используются алгоритмы, отталкивающиеся от относительных позиций.
 - Нет информации о геометрии сцены. Используются техники «структуры из движения» (Structure from motion, Sfm).

Для комплекса получение информации о геометрии сцены предполагается с использованием стерео-съёмки — получения стерео-пары путем съёмки сцены двумя камерами с небольшим отличием в ракурсе^[17].

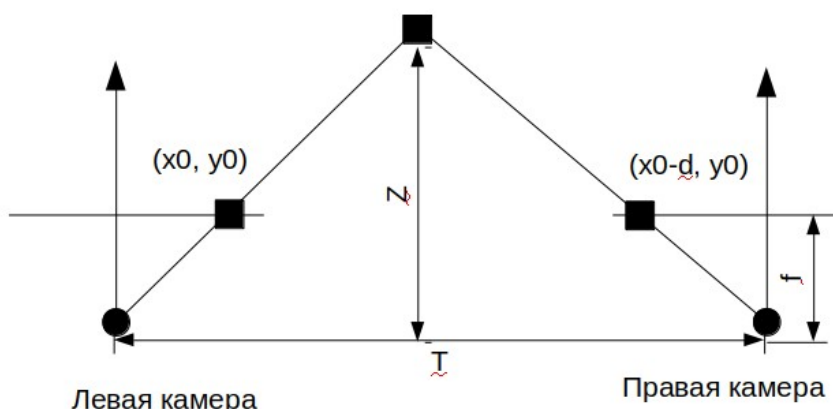


Рис. 1.3.1: определить дистанцию до объекта — Z можно по отношению: $(T-d)/(Z-f)=T/Z$, из которого следует: $Z=fT/d$, где T — расстояние между левой и правой камерами, d — смещение координаты объекта на проекциях камер, f — фокусное расстояние камер.

3. Подстановка виртуального объекта на изображение с учетом мировой системы координат и геометрии сцены для получения дополненной реальности. На данном этапе используются различные системы рендеринга трехмерных объектов.

1.4 РАЗРАБОТКА ТЕХНИЧЕСКОГО ЗАДАНИЯ НА РЕАЛИЗАЦИЮ СИСТЕМЫ

1.4.1 ОБЩИЕ СВЕДЕНИЯ

Полное наименование системы:

Плановые сроки начала и окончания работы по созданию системы:

10.01.19 — 10.05.19

1.4.2 НАЗНАЧЕНИЕ И ЦЕЛИ СОЗДАНИЯ(РАЗВИТИЯ) СИСТЕМЫ

Назначение системы: комплекс должен реализовывать развлекательную и маркетинговую функции для посетителей.

Цели создания системы: привлечение посетителей мероприятия к стенду компании. Оценивать работу комплекса предполагается по числу людей, воспользовавшихся им за время мероприятия.

1.4.3 ХАРАКТЕРИСТИКИ ОБЪЕКТА АВТОМАТИЗАЦИИ

Объектом автоматизации является привлечение внимания к себе со стороны компании для большего охвата аудитории мероприятия.

Эксплуатация производится в местах массового скопления людей. Предполагается взаимодействие посетителей с комплексом.

1.4.4 ТРЕБОВАНИЯ К СИСТЕМЕ

Требования к системе в целом:

Комплекс должен предоставлять интерфейс пользователю для выбора объекта дополненной реальности, который будет в дальнейшем вставлен на изображение. После этого пользователь должен расположить ключевые объекты в пределах области видимости камер комплекса. После съемки изображения, на экран пользователю выводится результат работы комплекса с предложением отправить его по введенным контактам. Пользователю должна быть предоставлена возможность оформить подписку на рассылку новостей от

компании по указанному адресу. В случае оформления подписки, указанные контакты должны быть сохранены в памяти устройства. Должна быть предусмотрена возможность выгрузки собранных контактов, загрузка трехмерных объектов для использования и смена появляющегося при перезагрузке системы логотипа ответственным лицом.

Требования к численности и квалификации персонала системы и режиму его работы:

Обслуживающий персонал должен быть ознакомлен со справочной документацией на устройство, иметь навыки обращения с компьютером, обладать хорошими коммуникативными навыками. Предполагается наличие 1-2 сотрудников вблизи комплекса при его работе. Для настройки комплекса и выгрузки собранных данных необходимо наличие человека с навыками системного администрирования и работы с Unix-подобными операционными системами. Режим работы обслуживающего персонала должен быть подобран с учетом необходимости постоянного нахождения рядом с комплексом обслуживающего персонала в течении всей длительности мероприятия.

Показатели назначения:

Основным показателем эффективной работы комплекса является число обслуженных им посетителей.

Требования к надежности:

Комплекс должен иметь возможность автономной работы в течении длительного времени. Корпус комплекса должен быть рассчитан на взаимодействие с посетителями в местах массового скопления людей. В случае зависания необходимо предусмотреть механизм жесткой перезагрузки системы с автоматическим запуском необходимого ПО по окончании загрузки ОС.

Требования к безопасности:

Комплекс должен отвечать требованиям пожарной безопасности, а также быть устойчивым во избежание заваливания набок с риском травмирования посетителей. В случае возникновения аварийной ситуации комплекс должен производить отключение собственных систем. На корпусе комплекса в доступном для обслуживающего персонала месте должна располагаться кнопка выключения питания. Управление комплексом должно ограничивать пользователя, не давая ему возможности оперировать какими-либо программными компонентами и подсистемами комплекса, кроме специально предназначенных для него.

Требования к эргономике и технической эстетике:

Интерфейс пользователя должен быть разработан с учетом эргономики. Внешний вид комплекса должен удовлетворять моральным и этическим нормам, оставлять положительные впечатления у посетителей и привлекать последних к себе и стенду компании.

Требования к транспортабельности:

Комплекс должен обладать достаточными прочностными характеристиками для частой транспортировки.

Требования к защите информации от несанкционированного доступа:

В целях обеспечения защиты информации пользователей, видео-поток не должен сохраняться на устройстве. Сделанное изображение должно удаляться с устройства после отправки его по указанному пользователем адресу. Устройство, в случае подключения к сети, должно фильтровать входящий/исходящий трафик для предотвращения попыток получить несанкционированный контроль извне. Контакты пользователей, оформивших подписку на рассылку,

должны храниться в зашифрованном виде. Должен использоваться алгоритм шифрования «Кузнечик», согласно ГОСТ Р34.12-2015 «Информационная технология. Криптографическая защита информации. Блочные шифры».

Требования по сохранности информации при авариях:

Должна быть предусмотрена возможность резервного копирования файлов комплекса с помощью подключения по протоколу SSH.

Требования к функциям (задачам):

Подсистема работы с пользователем должна обеспечивать:

- съемку видео-потока: видео-поток должен идти с частотой кадров — не менее 20 кадров в секунду с разрешением Full HD;
- поиск ключевых объектов на кадрах видео-потока: алгоритм поиска не должен давать ложных срабатываний, число отсутствия срабатываний должно быть минимальным;
- вставку трехмерных объектов дополненной реальности на кадры видео-потока: объекты на выходном изображении должны находиться между двух ключевых объектов масштабированными до соответствия их размеров расстоянию между ключевыми объектами;
- сбор данных о посетителях: система должна собирать данные о числе посетителей, воспользовавшихся ею, и email-адреса пользователей, оформивших подписку на информационную рассылку;
- шифрование собранных данных: данные должны быть зашифрованы алгоритмом «Кузнечик» согласно ГОСТ Р34.12-2015;

- хранение данных: данные должны храниться в файловой системе съемного накопителя устройства;
- отправку изображений по электронной почте: отправляемое письмо должно быть построено согласно загруженному в систему шаблону.

Подсистема работы с администратором устройства должна обеспечивать:

- возможность выгрузки сохраненных данных на другое устройство: сохраненные зашифрованные данные можно копировать на другое устройство посредством протокола SSH;
- возможность загрузки трехмерных моделей для использования их комплексом: в файловой системе устройства должен присутствовать каталог с трехмерными моделями, которые комплекс будет предлагать пользователям для вставки на изображения;
- возможность перезагрузки системы: возможность мягкой перезагрузки системы администратором, имеющим доступ к удаленному управлению устройством;
- возможность выключения системы: возможность мягкого выключения системы администратором, имеющим доступ к удаленному управлению устройством;
- возможность редактирования фонового рисунка для экрана загрузки системы: в файловой системе комплекса должен находиться файл, который будет использоваться системой в качестве фонового рисунка при загрузке, с возможностью его замены на другой администратором;
- возможность редактирования шаблона отправляемых электронных писем: в файловой системе комплекса должен находиться файл с указанными темой и текстом для отправляемых

электронных писем с возможностью его замены на другой администратором.

Требования к обеспечению:

Все используемое программное обеспечение должно быть доступно для использования физическими и юридическими лицами.

1.4.5 СОСТАВ И СОДЕРЖАНИЕ РАБОТ ПО СОЗДАНИЮ(РАЗВИТИЮ) СИСТЕМЫ

- Проектирование интерактивного фото-комплекса с виртуальными объектами дополненной реальности.
 - Разработка метода проецирования виртуального трехмерного объекта дополненной реальности.
 - Проектирование информационной модели фото-комплекса
 - Проектирование архитектуры комплекса технических средств для реализации процедуры фотографирования.
 - Проектирование алгоритма поведения системы: фотографирование с виртуальным трехмерным объектом дополненной реальности.
- Разработка интерактивного фото-комплекса с виртуальными объектами дополненной реальности
 - Выбор инструментов реализации
 - Программная реализация
 - Тестирование комплекса
 - Документирование комплекса

1.4.6 ПОРЯДОК КОНТРОЛЯ И ПРИЕМКИ СИСТЕМЫ

После разработки комплекс должен пройти ряд испытаний для проверки его прочностных характеристик, соответствия требованиям пожарной безопасности, проверке эргономики интерфейса. Должна быть осуществлена проверка соответствия комплекса требованиям информационной безопасности и проверка отказоустойчивости.

1.4.7 ТРЕБОВАНИЯ К ДОКУМЕНТИРОВАНИЮ

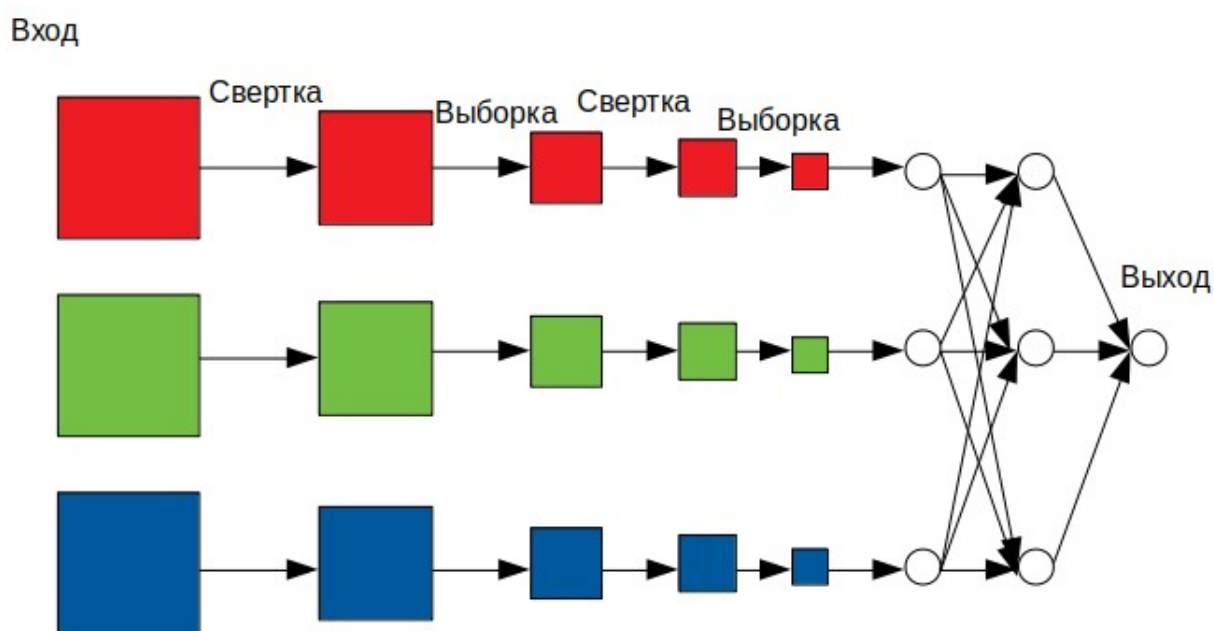
Устройство должно сопровождаться подробной документацией об особенностях эксплуатации, возможных проблемах и способах их решения.

2. ПРОЕКТИРОВАНИЕ ИНТЕРАКТИВНОГО ФОТОКОМПЛЕКСА С ВИРТУАЛЬНЫМИ ОБЪЕКТАМИ ДОПОЛНЕННОЙ РЕАЛЬНОСТИ

2.1 РАЗРАБОТКА МЕТОДА ПРОЕЦИРОВАНИЯ ВИРТУАЛЬНОГО ТРЕХМЕРНОГО ОБЪЕКТА ДОПОЛНЕННОЙ РЕАЛЬНОСТИ

Задачей комплекса является проецирование виртуальных объектов дополненной реальности на кадры видео-потока. Вставка объекта происходит в регион изображения, отмеченный ключевыми объектами — руками. Для определения местоположения ключевых объектов на кадре в данном случае необходимо прибегнуть к средствам машинного обучения и моделям, предназначенным для обнаружения объектов (Object Detection).

Большинство этих средств базируются на сверточных нейронных сетях (Convolutional Neural Network — CNN)^[18]. Суть сверточных нейронных сетей заключается в проведении операций свертки и выборки над изображением перед



попыткой распознавания на нем объектов.

Рис.2.1.1: топология простой сверточной нейронной сети.

Операция свертки заключается в поэлементном умножении скользящего окна исходной матрицы на ядро свертки.

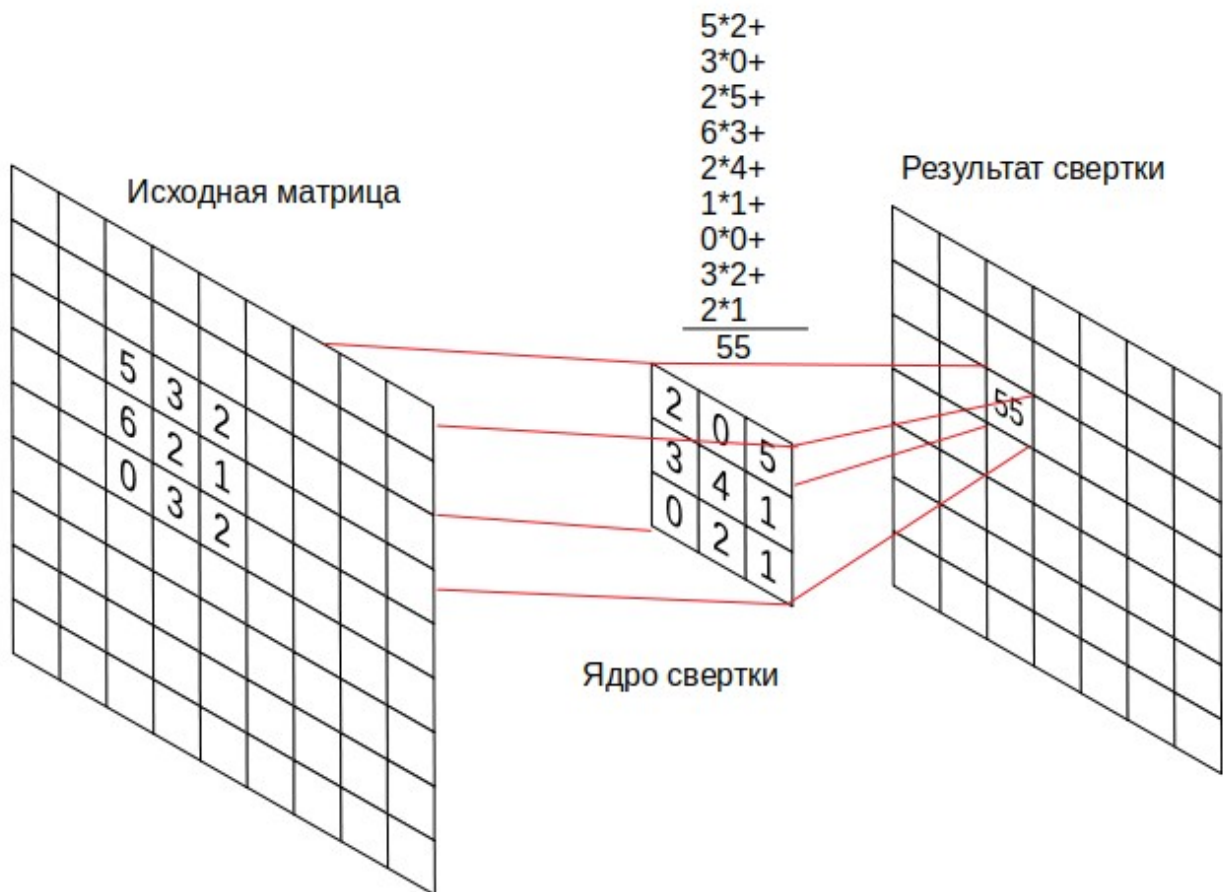


Рис.2.1.2: схема операции свертки.

Операция свертки выявляет соответствие региона изображения ядру свертки. Чем выше результирующее значение — тем сильнее выражен признак, определяемый ядром, в данном участке. При тренировке сети значения элементов в ядрах свертки меняются для минимизации ошибки.



Рис.2.1.3: в указанном регионе результат свертки будет наибольшим.

Следующим шагом идет выборка. Она применяется для «уплотнения» изображения до менее подробного, уменьшая размерность изображения и оставляя после себя признаки, обнаруженные предшествующим этапом свертки. Для выборки обычно используется операция MaxPooling — выбор максимального с размером ядра 2x2, что позволяет уменьшить размер матрицы в 2 раза.



Рис.2.1.4: операция MaxPooling с ядром 2x2.

В результате прохождения этапов свертки и выборки получается набор соответствий признакам, определяемым ядрами свертки, использовавшимися ранее. Данные разворачиваются в вектор и проходят один или несколько полносвязных слоев сети. Результатом данного этапа является классификация объекта на изображении.

Большинство методов обнаружения объектов на изображениях применяют модель классификатора к различным регионам изображения в разном масштабе. Регионы с высоким откликом классификатора считаются содержащими в себе определяемый им объект.

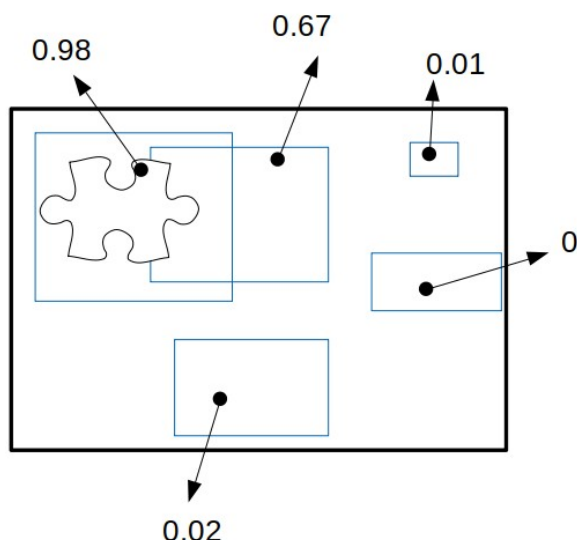


Рис.2.1.5: обнаружение объекта на изображении с помощью классификатора в выбранных алгоритмом регионах.

В зависимости от алгоритма выборки регионов изображения для дальнейшей классификации, обнаружение объектов может иметь разную производительность и точность. Часть алгоритмов используют скользящее окно, содержимое которого проходит классификацию. Другие — выделяют регионы более сложными методами.

YOLO (You Only Look Once)^{[19][20]} — один из алгоритмов обнаружения объектов. Его отличительной особенностью является применение одной нейронной сети ко всему изображению для получения регионов, содержащих объекты. Такой подход позволяет значительно повысить скорость работы алгоритма.

Входное изображение разделяется сеткой с фиксированным размером ячеек, для каждой ячейки выбирается несколько прямоугольников (число фиксировано для всех ячеек), для каждого из которых определяется уровень «доверия» и условная вероятность нахождения в нем объекта определенного класса в том случае, если в нем присутствует объект. Число прямоугольников в таком случае заметно ниже, чем число регионов, выделяемых другими алгоритмами.

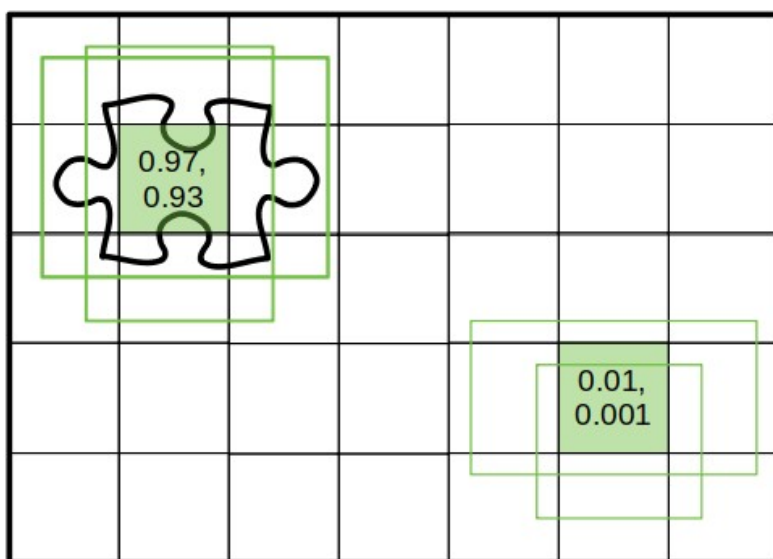


Рис.2.1.6: разбиение изображения на ячейки и выделенные прямоугольники для некоторых из них (отображены прямоугольники лишь для двух ячеек, алгоритм создает фиксированное число прямоугольников для каждой ячейки).

На рисунке в ячейки записан показатель «доверия» прямоугольникам, принадлежащим им. На следующем этапе все прямоугольники, чьи показатели «доверия» не превосходят некоторого порогового значения будут отсеяны и не будут участвовать в дальнейшем анализе.

Следующий этап — алгоритм Non Max Supression — используется для отсеивания еще части прямоугольников.

Оставшиеся прямоугольники считаются содержащими в себе объект того класса, условная вероятность для которого в данном прямоугольнике выше.

Данный алгоритм дает высокую точность, малое число ложных срабатываний и большую производительность. Существуют также различные вариации алгоритма, использование которых позволяет еще сильнее ускорить его работу.

Обнаружение ключевого объекта на изображении с камер позволяет нам определить его координаты x_0 и y_0 на сцене. После этого необходимо определить оставшуюся координату — z_0 .

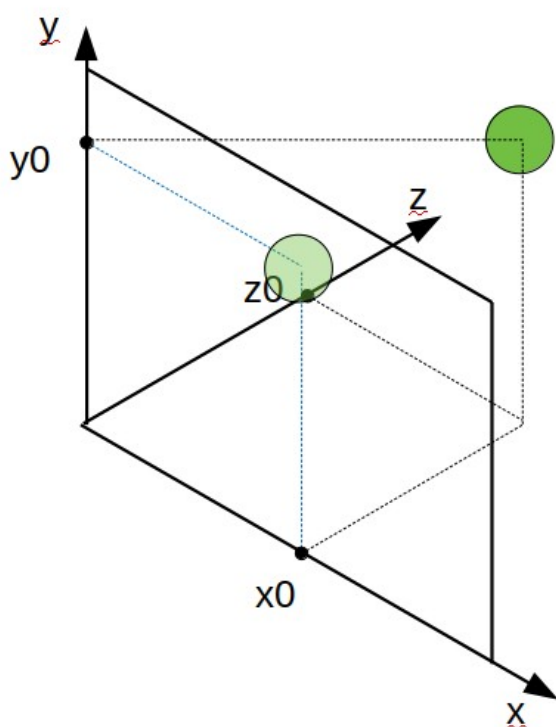


Рис.2.1.7: по изображению с камеры мы можем определить положение объекта по осям x и y , но не знаем его удаленность по оси z .

Используя специализированные библиотеки необходимо построить карту глубины изображения^[21]. Карта глубины — это изображение, на котором для каждого пикселя хранится его расстояние до камеры. Для этого необходимо использовать стерео-фотографию двумя камерами.

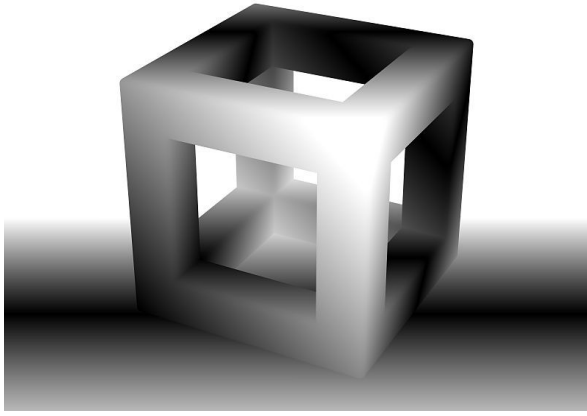


Рис.2.1.8: карта глубины трехмерного объекта. Светлые области находятся ближе к наблюдателю.

Имея построенную карту глубины и выделенный на изображении прямоугольный регион, содержащий в себе ключевой объект, можно определить недостающую координату.

Т.к. выделенный регион помимо ключевого объекта может содержать дополнительно участки фона и других объектов необходимо определить какая именно его часть относится к искомому объекту. В случае описываемого комплекса, можно предположить, что ближайшим к камере объектом в выделенном регионе будет используемый ключевой объект. Таким образом, можно считать, что координата z_0 равна минимальному значению глубины в выделенном регионе.

Вставка трехмерного объекта происходит по координатам одной из найденных рук. Затем происходит поворот объекта по направлению ко второй руке. После этого необходимо масштабировать объект для совпадения его размеров с пространством между рук.

В целях предотвращения искажений первоначальной концепции дизайнеров объекта, предполагается производить масштабирование пропорционально для всех осей.

2.2 ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ МОДЕЛИ ИНТЕРАКТИВНОГО ФОТО-КОМПЛЕКСА

В виду использования на устройстве операционной системы семейства Unix, необходимо спроектировать информационную модель комплекса для файловой системы Unix^[22].

Файловая система Unix имеет древовидную структуру. В качестве листьев выступают файлы и файлы устройств.

В корневом каталоге - / находится набор стандартных каталогов системы. Т.к. доступ к содержимому большинства из них при обычном использовании устройства не нужен, конфигурационные файлы такие, как трехмерные модели, фоновое изображение для экрана загрузки и шаблон для электронных писем, должны находиться в пределах каталога /home/*user*, где *user* - профиль заранее созданного в системе пользователя, не наделенного правами супер-пользователя для доступа к системным файлам(предположительно «User»). Это позволит избежать случайных изменений системных файлов администратором, что может привести к прекращению работы комплекса или его неправильной работе.

Возможный вариант структуры каталогов представлен на схеме:

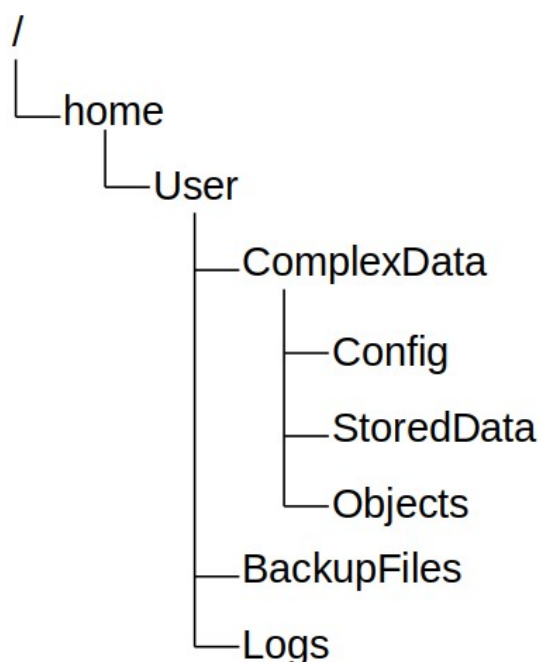


Рис. 2.2.1

На приведенной схеме:

- каталог BackupFiles предназначен для хранения архивированных резервных копий системы;
- в каталог Logs будут сохраняться логи программного обеспечения комплекса;
- каталог Config хранит в себе файл фонового изображения экрана загрузки операционной системы и шаблон электронных писем;
- каталог StoredData предназначен для хранения собранных комплексом данных о посетителях;
- каталог Objects является библиотекой трехмерных объектов, доступных пользователю комплекса для вставки на кадры видео-потока.

Таким образом, конфигурация комплекса для использования его определенной компанией на определенном мероприятии сводится к загрузке необходимых файлов в соответствующие каталоги.

Выгрузка резервных копий и собранных данных производится копированием данных из соответствующих каталогов. После получения данных о собранных устройством адресах они остаются в зашифрованном виде и их расшифровка возможна с использованием ключа, передающегося ответственному лицу.

В случае возникновения неполадок отследить место их появления можно с использованием хранящихся логов работы.

Администратор устройства не нуждается в доступе к другим файлам системы.

2.3 ПРОЕКТИРОВАНИЕ АРХИТЕКТУРЫ КОМПЛЕКСА ТЕХНИЧЕСКИХ СРЕДСТВ ДЛЯ РЕАЛИЗАЦИИ ПРОЦЕДУРЫ ФОТОГРАФИРОВАНИЯ

Структура комплекса технических средств:

Структура комплекса выбрана для обеспечения работы функций устройства с необходимым качеством и производительностью.

В качестве центрального аппаратного устройства был выбран Raspberry Pi Compute Module 3 с расширением StereoPi т. к. он предназначен для работы с двумя камерами посредством интерфейса CSI. Камеры, подключенные к нему данным образом, транслируют видео-поток напрямую в процессор, ускоряя обработку кадров в несколько раз.

В комплексе будут использоваться две камеры модели Raspberry Pi Camera Module v2. Они способны снимать видео-поток в разрешении Full HD с максимальной частотой кадров — 30 кадров в секунду и подключаются к устройству по интерфейсу CSI.

На устройство будет установлена свободно распространяющаяся операционная система Raspbian. Она оптимизирована специально для работы с одноплатными компьютерами семейства Raspberry Pi, что повысит эффективность работы комплекса по сравнению с использованием других операционных систем.

Корпус устройства может быть спроектирован и создан под заказ.

Примерный внешний вид комплекса в собранном состоянии:

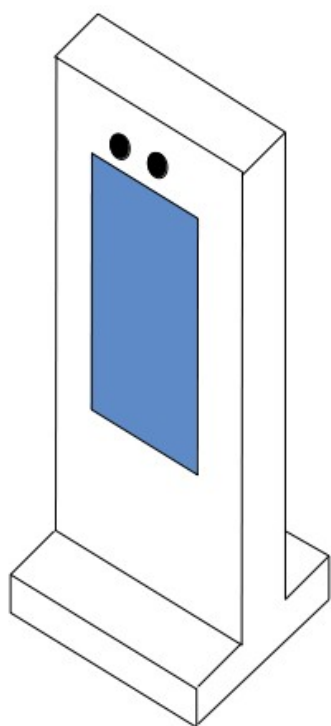


Рис. 2.3.1: эскиз варианта внешнего вида комплекса в собранном состоянии

Функционирование комплекса осуществляется по схеме на рис. 2.3.2 и 2.3.3:

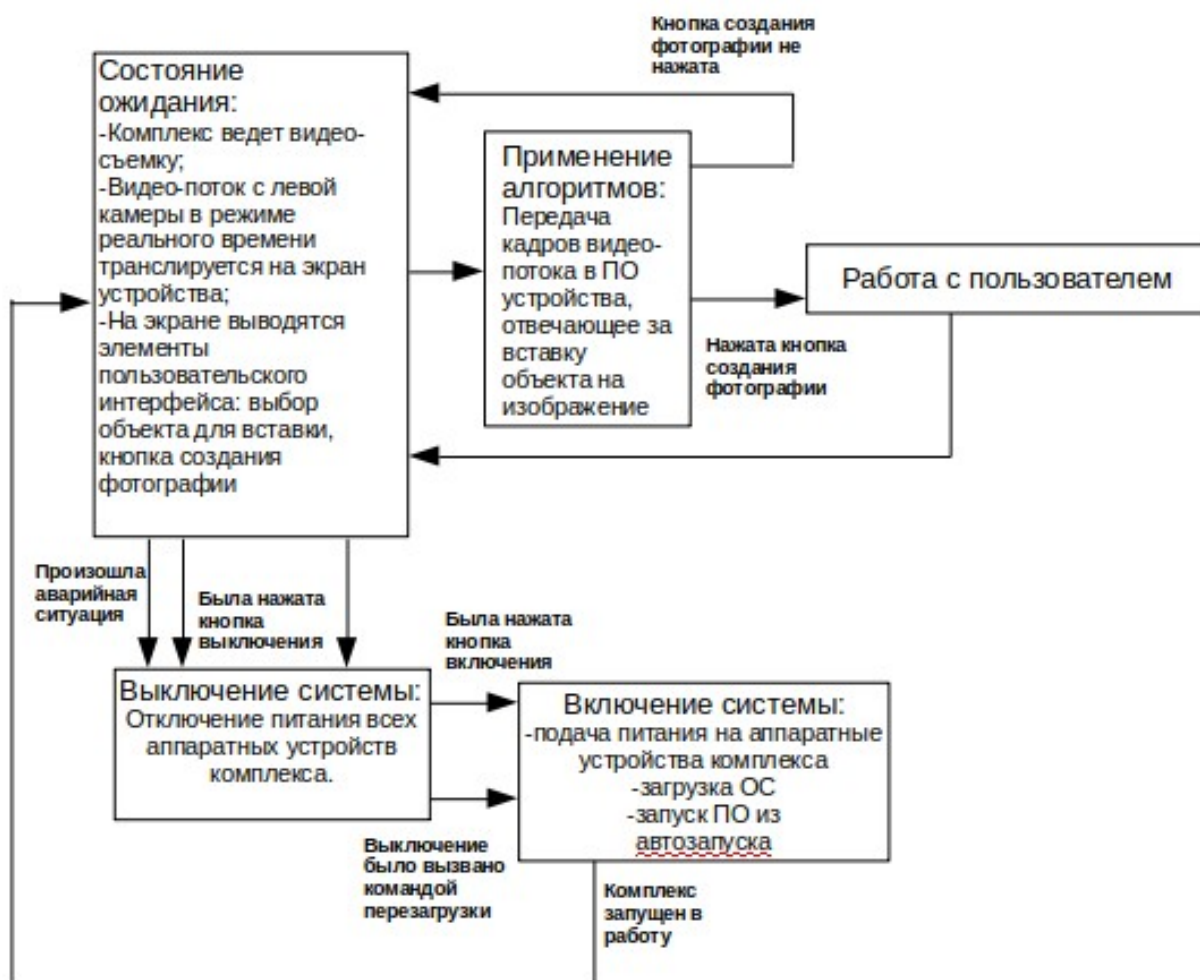


Рис.2.3.2: схема функционирования комплекса

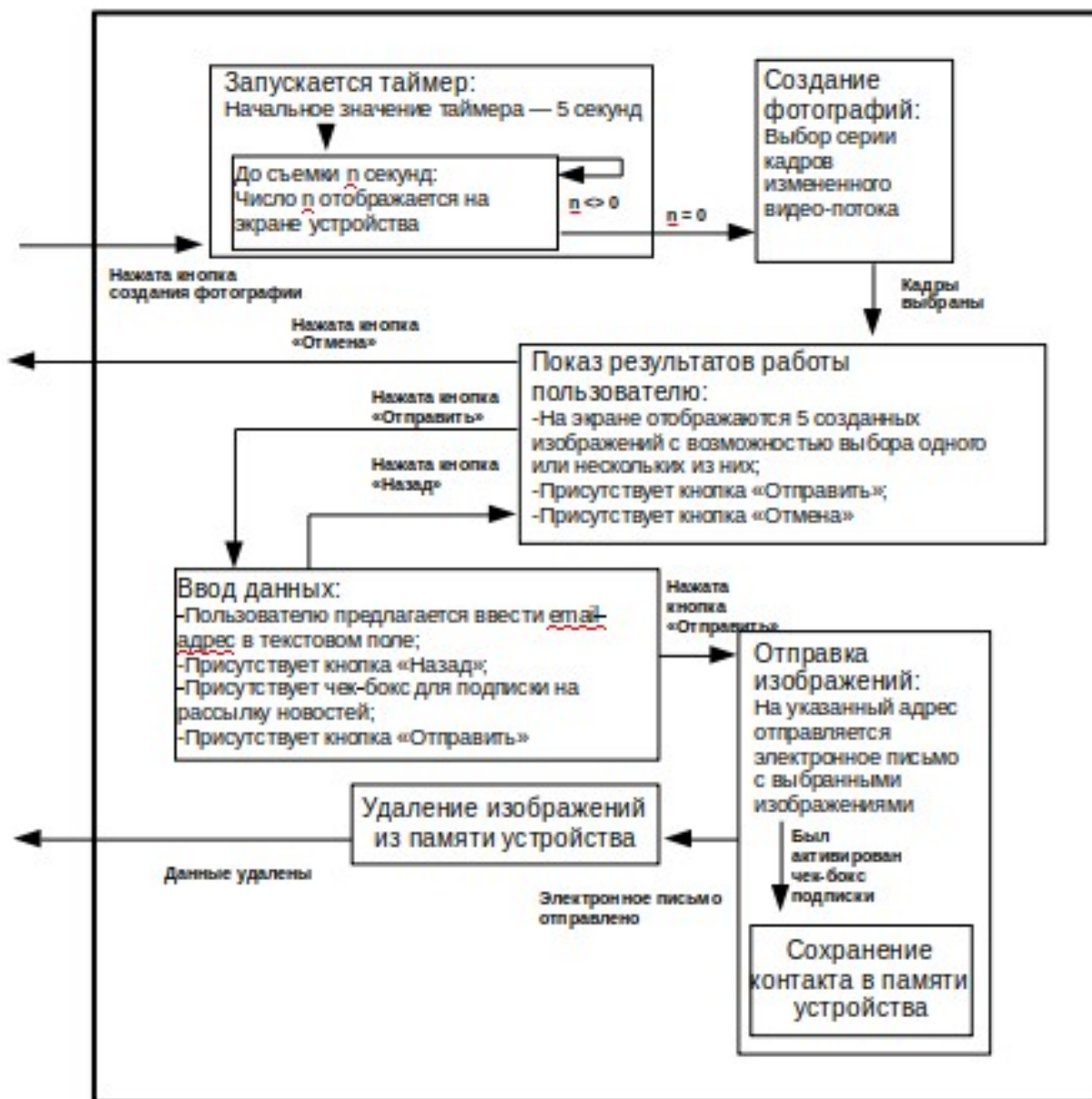


Рис. 2.3.3: декомпозиция состояния «Работа с пользователем» из рис. 2.3.2

На объектах комплекс должен располагаться в защищенных от дождя и прямых солнечных лучей местах, отдаленных от источников тепла. Не допускается попадание влаги на корпус и детали комплекса. Комплекс не должен загромождать проход посетителям мероприятия и преграждать пути эвакуации. Для лучшей работы алгоритмов распознавания рекомендуется установить фоновый стенд на расстоянии 2-3 метра от комплекса напротив камер.

Технические средства комплекса защищены от механических воздействий жестким корпусом. При перегреве

аппаратных устройств комплекс будет аварийно отключен. Защита данных обеспечивается путем их шифрования. Несанкционированные подключения блокируются активным файрволом.

Средства вычислительной техники:

Аппаратные устройства выбираются исходя из необходимых вычислительных ресурсов для работы комплекса, возможности синхронной стерео-видео-съемки удовлетворительного качества, хранения собранных данных и подключения к сети интернет.

Для работы комплекса необходимо устойчивое соединение с сетью интернет со скоростью, достаточной для быстрой передачи нескольких изображений объемом 2-10 Мб.

Необходимый персонал для комплекса: оператор-помощник, осуществляющий справочную функцию для пользователей; мастер по настройке комплекса.

Для возможности настройки комплекса мастера необходимо оборудовать ПК с необходимым ПО.

В случае возникновения аварийной ситуации, системы защиты комплекса отключают все аппаратные устройства до дальнейшего ручного включения.

2.4 ПРОЕКТИРОВАНИЕ АЛГОРИТМА ПОВЕДЕНИЯ СИСТЕМЫ: ФОТОГРАФИРОВАНИЕ С ВИРТУАЛЬНЫМ ТРЕХМЕРНЫМ ОБЪЕКТОМ ДОПОЛНЕННОЙ РЕАЛЬНОСТИ

План взаимодействия пользователя с комплексом:

1. Пользователь подходит к комплексу со стороны экрана на расстояние 0-2 метра;
2. На экране транслируется видео-поток, на котором присутствует пользователь;
3. Пользователь выбирает трехмерный объект для вставки;
4. На экран начинает транслироваться аугментированный видео-поток с трехмерным объектом, вставленным между рук пользователя;
5. Пользователь нажимает кнопку создания изображения на экране комплекса;
6. На экране начинают высвечиваться цифры обратного отсчета;
7. Пользователь занимает позу для фотографии. Руки пользователя находятся в зоне видимости камер;
8. Обратный отсчет заканчивается;
9. На экране отображаются созданные изображения;
10. Пользователь выбирает одно или несколько изображений и нажимает кнопку «Отправить», или нажимает кнопку «Отмена»;
11. Если нажата кнопка «Отправить», появляется окно ввода контактных данных с возможностью оформить подписку на рассылку;
12. Пользователь вводит контактные данные и, опционально, активирует чек-бокс для оформления подписки, или возвращается на предыдущий этап нажатием кнопки «Назад»;
13. Если нажата кнопка «Отправить», изображения отправляются по указанному email-адресу;
14. Пользователь может повторить работу с комплексом или уйти.

3. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ ИНТЕРАКТИВНОГО ФОТО-КОМПЛЕКСА С ВИРТУАЛЬНЫМИ ОБЪЕКТАМИ ДОПОЛНЕННОЙ РЕАЛЬНОСТИ

3.1 ВЫБОР ИНСТРУМЕНТОВ ДЛЯ РАЗРАБОТКИ

Приблизительная схема работы комплекса:



Рис. 3.1.1: схема взаимодействия модулей комплекса

Каждый этап использует свои технологии, инструменты и библиотеки для упрощения разработки. Для поиска объектов на кадрах видео-потока необходимо задействовать стек технологий машинного обучения и компьютерного зрения в частности. Определение глубины на изображение задействует технологии, применяемые в разработке роботов и средств фотограмметрии.

Для поиска ключевых объектов предполагается использование одного из нескольких вариантов:

1. Использование свободного фреймворка Darknet^[23]. Данный фреймворк позволяет использовать алгоритм YOLO (You Only Look Once) для детектирования и классификации объектов на изображении.

2. Использование свободного фреймворка Darkflow^[25]. Это реализация алгоритма YOLO с использованием другого свободного фреймворка — Tensorflow^[24].

3. Использование фреймворка Tensorflow Object Detection API напрямую. Предоставляет наибольшую гибкость, однако тренировка собственного детектора может занимать несколько суток даже с использованием нескольких GPU. С учетом необходимости подбора гиперпараметров эмпирическим путем, такой подход может сильно замедлить разработку.

4. Использование форка фреймворка Darknet от пользователя GitHub AlexeyAB^[26]. Обладает всеми положительными чертами оригинального фреймворка и добавляет к нему дополнительные возможности.

Из перечисленных подходов однозначно необходимо исключить непосредственное использование Tensorflow Object Detection API из-за больших потерь во времени на тренировку детектора. Darkflow показал низкую скорость обучения моделей во время экспериментальных попыток. Из двух оставшихся вариантов был выбран клон оригинального фреймворка от AlexeyAB из-за имеющихся у него дополнительных опций.

Для построения карты глубины стерео-изображения было решено использовать библиотеку OpenCV^[27], доступную по лицензии BSD и предоставляющую возможность использования множества алгоритмов компьютерного зрения.

Имея данные о расположении ключевого объекта на двумерном изображении с камеры и его удаленности, определяемой по карте глубины, необходимо реализовать алгоритм метода смещения координат для определения положения объекта в пространстве.

Для вставки 3d-модели на изображение будут использоваться библиотеки PyOpenGL^[28] и PyGame^[29]. Лицензии — BSD и GNU GPL соответственно.

3.2 МОДУЛЬ РАСПОЗНАВАНИЯ РУК

Для распознавания рук был выбран один из фреймворков, реализующих алгоритм YOLO. Для его использования необходимо сперва скачать исходные коды проекта из репозитория AlexeyAB/darknet на сайте GitHub.

После этого необходимо отредактировать файл Makefile: для ускорения работы программы на системах с GPU необходимо проставить значения: «GPU=1», «CUDNN=1»; для ускорения работы на CPU: «AVX=1», «OPENMP=1».

Использование GPU ускоряет обработку изображений в несколько раз.

После этого необходимо собрать и скомпилировать проект с помощью утилиты Make.

Установив фреймворк на устройство, необходимо создать конфигурационные файлы для модели, которая будет использована.

В данной работе будет использована модель yolov3-tiny.

Необходимо дважды копировать конфигурационный файл соответствующей модели в каталоге cfg собранного проекта для тренировки модели и для её использования. В первой копии необходимо изменить значение параметра batch с 1 на 64 и значение параметра subdivisions с 1 на 8. Далее в каждой из копий необходимо изменить значение параметра classes для каждого из [yolo]-слоев модели, сделав его равным числу классов для распознавания: 1 для рассматриваемого случая. Для каждого [convolutional]-слоя, предшествующего [yolo]-слою, необходимо выставить значение filters по формуле: $\text{filters} = (\text{число классов} + 5) * 3$. В случае, рассматриваемом в работе, данное значение будет равным 18.

Файл darknet/data/obj.names необходимо создать, чтобы он содержал только один лейбл — hands.

В каталоге cfg необходимо создать файл obj.data со следующим содержимым:

```
classes = 1
ntrain = dataset_folder/train/train.txt
ntest = dataset_folder/test/test.txt
```

```
nnames = data/obj.names
nbackup = backup
```

Где `dataset_folder/train` — каталог, содержащий изображения и аннотации к ним для тренировки модели; `train.txt` — файл, содержащий в себе имена всех файлов изображений, участвующих в тренировке; `dataset_folder/test` — каталог с тестовым набором, построенным аналогично обучающему; `backup` — каталог, в который будут сохраняться контрольные точки при обучении модели.

После того, как все этапы конфигурации были выполнены, необходимо загрузить файл претренированных весов для соответствующей модели с сайта проекта `darknet`.^[30]

Далее этот файл весов необходимо отредактировать посредством выполнения команды: `darknet partial config weights new_weights 15`, где: `config` — изначальный файл конфигурации модели, который был скопирован ранее; `weights` — файл весов, загруженный на данном этапе; `new_weights` — имя для файла, в который будет сохранен результат выполнения команды.

На этом процесс конфигурации заканчивается.

Для обучения модели необходим набор данных, на котором будет проходить тренировка.

В процессе обучения модели для данной работы был использован сборник нескольких открытых наборов данных, доступный для скачивания с сайта Оксфордского университета^[31]. Набор содержит более 4000 изображений и аннотации к ним в формате `matplotlib`.

`Darknet` принимает на вход файлы аннотаций, составленные в строго определенном виде и записанные в формате `txt`.

В ходе работы был написан скрипт для автоматического преобразования файлов аннотаций из указанного набора данных в принимаемый фреймворком вид, а также записывающий имена файлов, для аннотаций к которым было проведено преобразование, в файл `train.txt`.

На этом подготовка к обучению модели окончена.

Для запуска тренировки используется команда: `darknet detector train cfg/obj.data config weights 1 -dont_show`, где:

`config` — конфигурационный файл для тренировки, созданный ранее; `weights` — автоматически отредактированный файл весов, созданный ранее; `-dont_show` — опция, отключающая показ графиков с показателями обучения. В случае, если обучение было прервано и необходимо его продолжить — можно использовать сохранение, сделанное во время прохождения последней контрольной точки. Для этого вместо файла весов, использовавшегося при первом запуске, необходимо использовать файл, сохраненный в папке `backup`.

В ходе процесса обучения в консоль выводятся показатели функции ошибки. Чем меньше данный показатель — тем точнее работает модель. Однако, в случае использования недостаточно большого набора данных, может возникнуть ошибка переобучения — когда модель показывает хорошую точность на обучающей выборке, но не дает удовлетворительного результата на данных, которые ей не показывались в процессе тренировки.

Тренировку можно прервать в любой момент. Например, если функция ошибки перестала снижаться.

Посмотреть результаты работы модели можно, воспользовавшись командой: `darknet detector demo cfg/obj.data config weights video -dont_show -out_filename result`, где: `config` — конфигурационный файл для использования модели; `weights` — файл весов, полученный при прохождении контрольной точки; `video` — входной видео-файл для проверки распознавания на нем; `result` - имя файла, в который будет записан результат.



Рис. 3.2.1: фрагмент демонстрационной видео-записи с результатом работы алгоритма

Фреймворк Darknet имеет возможность компиляции в качестве библиотеки для использования его функций в других программах.

Процесс обучения столь сложной модели является затратным по аппаратным ресурсам и времени. Поэтому для ускорения тренировки была использована платформа Google Colaboratory, предоставляющая бесплатный доступ к виртуальной GPU и возможность использования серверов компании Google для тренировки моделей.

3.3 МОДУЛЬ ПОСТРОЕНИЯ КАРТ ГЛУБИН

Для построения карт глубин будут использованы функции библиотеки OpenCV, доступной для использования на нескольких языках программирования.

Класс StereoBM позволяет рассчитать смещение каждого пикселя на правом изображении относительно левого. Использование формулы триангуляции к полученным данным позволяет получить расстояние до объекта в абсолютном значении.



Рис. 3.3.1: искусственно созданная стерео-пара и результат работы алгоритма для неё

Данный алгоритм производит вычисление смещений пикселей на двух изображениях строго по горизонтали. Это означает, что использующиеся камеры должны быть одинаковыми, расположенными строго параллельно, с нулевым углом поворота относительно друг-друга. В реальности добиться подобного невозможно.

Для решения данной проблемы для закреплённой пары камер проводится калибровка: производится серия снимков заранее известного объекта, автоматическими методами высчитываются отклонения и информация о них сохраняется.

В дальнейшем, все изображения проходят этап «выпрямления» перед передачей их в функцию построения карты смещений. Таким образом удастся добиться достаточной точности в неидеальных условиях.

Для данной калибровки можно использовать библиотеку StereoVision^[33], распространяющуюся по свободной лицензии GNU v3.

Построение карт смещений библиотекой OpenCV использует два дополнительных параметра:

-numDisparities — радиус поиска смещения. Для каждого пикселя алгоритм будет искать смещение от 0 до numDisparities.

-blockSize — размер блоков, сравниваемых алгоритмом. Размер блока должен быть нечётным т. к. центром блока считается текущий рассматриваемый пиксель. Большое значение blockSize делает результат более гладким, но менее точным; малое — напротив, делает результат более детализированным, но повышает вероятность, что алгоритм найдет неправильное соответствие.

Для получения желаемых результатов необходимо подобрать значения этих двух параметров, которые их обеспечат.

3.4 РАЗРАБОТКА ПРОЕЦИРОВАНИЯ ВИРТУАЛЬНОГО ТРЕХМЕРНОГО ОБЪЕКТА ДОПОЛНЕННОЙ РЕАЛЬНОСТИ

Имея три координаты ключевых объектов в пространстве, его можно отобразить на экране. Для этого в работе использовались библиотеки PyOpenGL и PyGame.

Для проецирования объекта необходимо разработать соответствующие правила.

Координаты на изображении отличаются от соответствующих им мировых координат. Для библиотек, обрабатывающих изображения, свойственно иметь систему координат с точкой начала отсчета в левом верхнем углу и осью Y , направленной вертикально вниз. OpenGL использует систему координат с началом координат в левом нижнем углу, осью Y , направленной вверх и осью Z — в сторону наблюдателя.

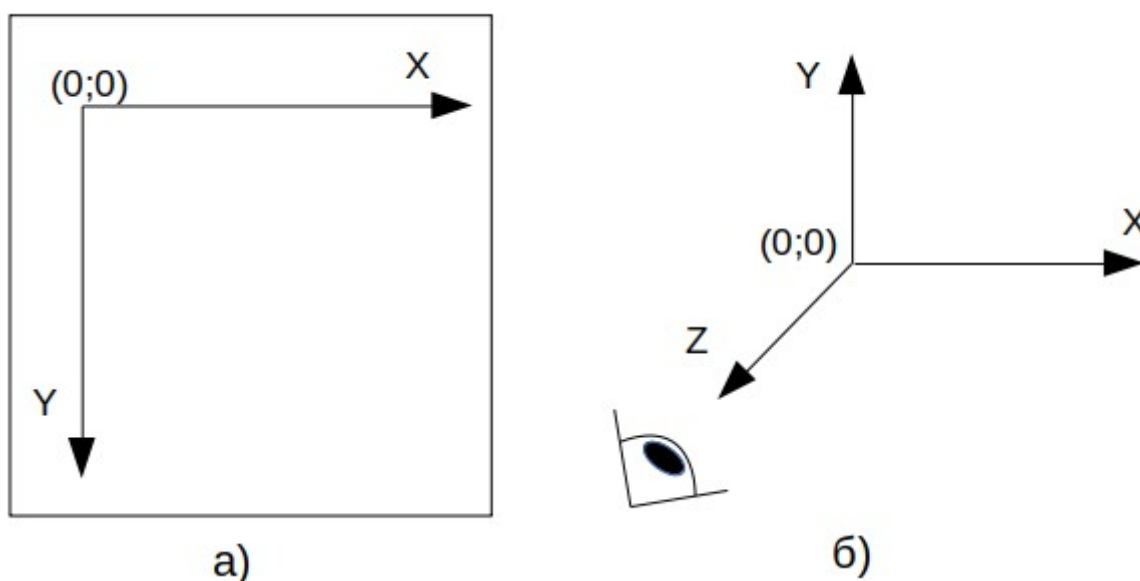


Рис. 3.4.1: отличия в системах координат изображения (а) и трехмерного пространства на экране (б)

Привести координату Y к соответствующему виду можно, воспользовавшись формулой: $Y^* = \text{Высота_Изображения} - Y$.

Следующим шагом необходимо отобразить полученные координаты на интервал мировых координат, видимый на экране. Для этого необходимо провести операции переноса и масштабирования.

В результате получаются координаты ключевых объектов в мировых координатах.

Так как описанный в файле трехмерный объект имеет фиксированный размер, перед проецированием его на изображение необходимо провести его масштабирования для совпадения его размеров с пространством между руками.

В связи с функциями, выполняемыми комплексом, коэффициент масштабирования будет вычисляться исходя из длины объекта.

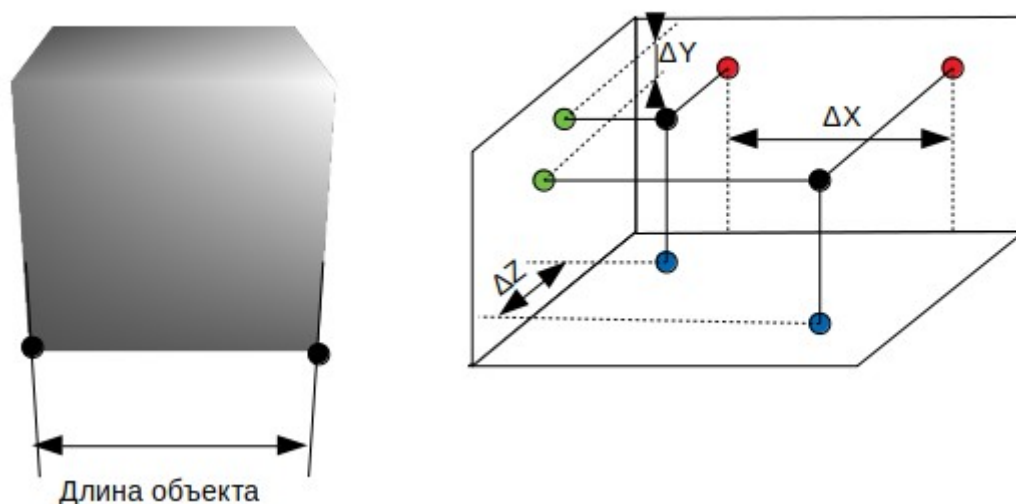


Рис. 3.4.2: объект, описанный в файле, и его расположение в пространстве при проецировании.

Коэффициент масштабирования определяется путем деления известной длины объекта на расстояние между руками в пространстве:

$$K = \text{Длина_Объекта} / \sqrt{(\Delta X * \Delta X + \Delta Y * \Delta Y + \Delta Z * \Delta Z)}$$

Умножение координат каждой из вершин на найденный коэффициент дает информацию об объекте в нужном масштабе.

Для дальнейших преобразований центральной точкой объекта будет считаться его крайняя левая-нижняя точка, максимально удаленная от наблюдателя.

Объект необходимо поместить так, чтобы его центральная точка совпала с крайним левым найденным на кадре ключевым объектом — рукой.

После этого необходимо повернуть объект на нужные углы вокруг осей OX и OZ .

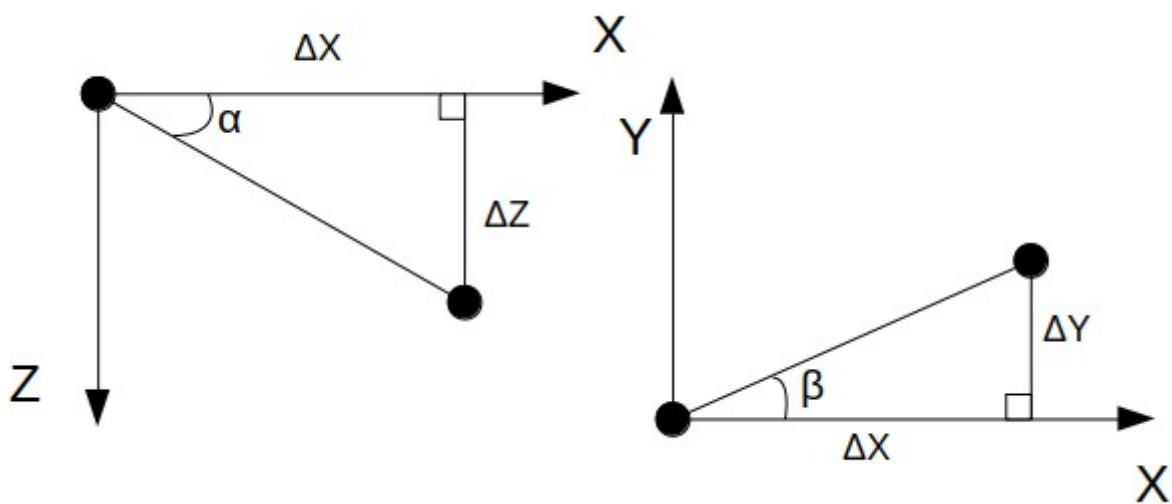


Рис. 3.4.3: геометрия углов поворота объекта

Необходимый угол поворота вокруг оси oY вычисляется по формуле:

$$\alpha = -\text{arctg}(\Delta Z/\Delta X);$$

вокруг оси oZ :

$$\beta = \text{arctg}(\Delta Y/\Delta X).$$

Поворот осуществляется против часовой стрелки.

После этого объект проецируется на экран.



Рис. 3.4.4: результат проецирования объекта

3.5 РЕАЛИЗАЦИЯ АЛГОРИТМА ПОВЕДЕНИЯ СИСТЕМЫ: ФОТОГРАФИРОВАНИЕ С ВИРТУАЛЬНЫМ ТРЕХМЕРНЫМ ОБЪЕКТОМ ВИРТУАЛЬНОЙ РЕАЛЬНОСТИ

Для обеспечения работы пользователя с комплексом необходимо спроектировать графический интерфейс пользователя для всех экранов, используемого программного обеспечения.

Основной экран должен содержать на себе список доступных объектов для вставки и кнопку создания фотографии. Все элементы должны располагаться так, чтобы пользователь находил их при первом использовании комплекса. Пользователь должен иметь возможность понять управление комплексом за минимальное время. Для этого интерфейс должен использовать привычные образы.

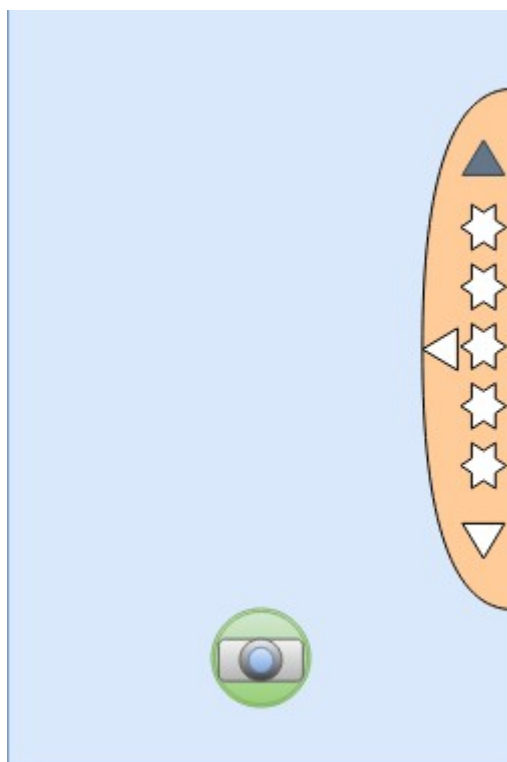


Рис. 3.5.1: эскиз интерфейса главного окна программы.

Для снижения пространства на экране, занимаемого списком объектов, он был помещен в проматываемую полосу. Стрелки сверху и снизу сигнализируют о возможности проматывания. Стрелка слева показывает возможность

выдвинуть список в бок, если пользователю будет удобнее увидеть больше объектов одновременно:

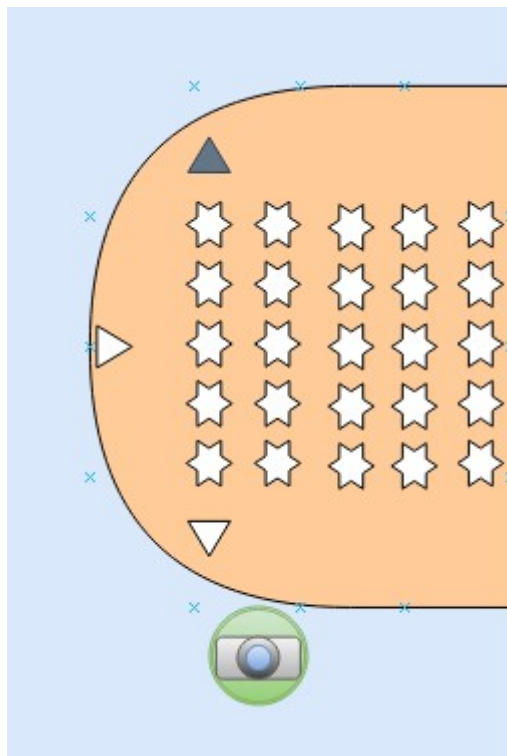


Рис. 3.5.2: эскиз интерфейса главного окна с выдвинутым в бок списком объектов.

При нажатии на кнопку создания фотографии запускается таймер обратного отсчета. Текущее его состояние отображается на месте кнопки:

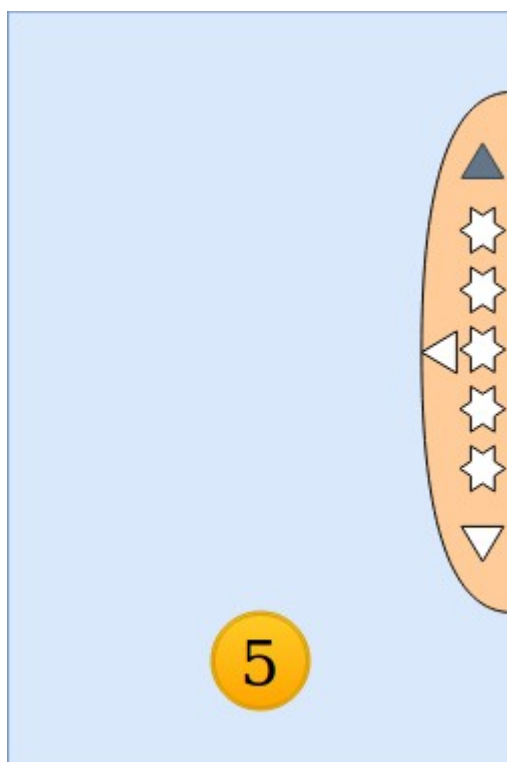


Рис. 3.5.3: эскиз интерфейса главного окна с запущенным таймером.

После истечения таймера программа проводит серию из четырех фото-снимков. На полученные кадры проецируется выбранный объект. После этого аугментированные изображения показываются пользователю с возможностью выбрать одну или несколько путем нажатия с удержанием на них:

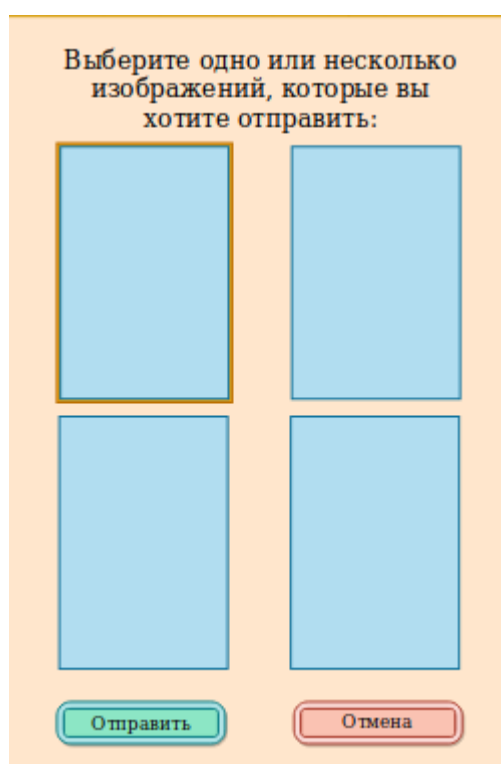


Рис. 3.5.4: эскиз интерфейса окна выбора изображений для отправки.

Пользователь может выбрать желаемые изображения и продолжить или вернуться на главный экран. В случае нажатия на кнопку «Отправить» пользователь попадает на экран ввода email-адреса:

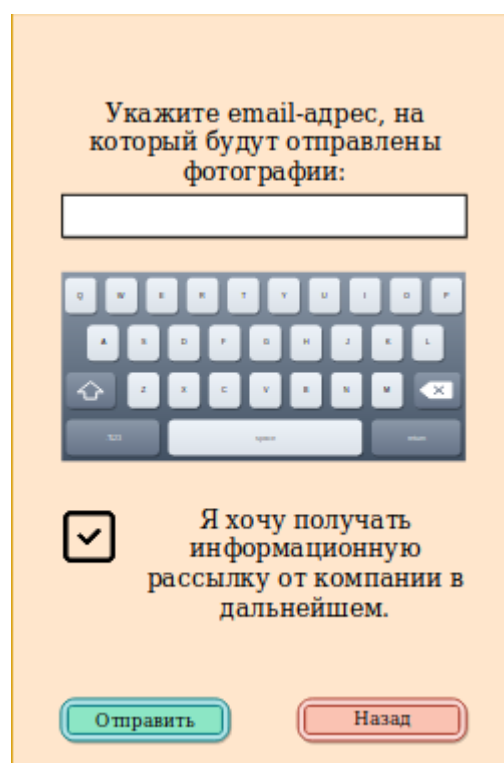


Рис. 3.5.5: эскиз интерфейса ввода email-адреса.

Пользователь вводит адрес электронной почты в текстовое поле, используя экранную клавиатуру. Ниже он может поставить отметку о том, что он желает получать информационную рассылку от компании. В таком случае, после нажатия на кнопку «Отправить» введенный email-адрес будет сохранен в зашифрованном виде.

Выбранные пользователем изображения будут отправлены на введенный адрес нажатием кнопки «Отправить». После этого программа вновь вернется на главный экран.

Нажатие кнопки «Назад» вернет пользователя на экран выбора изображений.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Бэйлесс М. "Just How 'Blind' Are We to Advertising Banners on the Web?"(2010)
2. Raspberry Pi [Электронный ресурс]//Список моделей.-Режим доступа: <https://www.raspberrypi.org/products/> .- (Дата обращения: 10.05.19).
- 3.Raspberry Pi [Электронный ресурс]//Модель Raspberry Pi 3 Model A+.-Режим доступа: <https://www.raspberrypi.org/products/raspberry-pi-3-model-a-plus/> .- (Дата обращения: 10.05.19).
- 4.Raspberry Pi [Электронный ресурс]//Модель Raspberry Pi 3 Model B+.-Режим доступа: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/> .- (Дата обращения: 10.05.19).
- 5.Orange Pi [Электронный ресурс]//Модель Orange Pi PC Plus.- Режим доступа: <http://www.orangepi.org/orangepipcplus/> .- (Дата обращения: 10.05.19).
- 6.Banana Pi [Электронный ресурс]//Модель Banana Pi BPI-F1.- Режим доступа: <http://www.banana-pi.org/f2.html> .- (Дата обращения: 10.05.19).
- 7.Stereo Pi Pi [Электронный ресурс]//Главная страница.-Режим доступа: <http://www.stereopi.com/> .- (Дата обращения: 10.05.19).
- 8.Raspberry Pi [Электронный ресурс]//Плата Compute Module 3.- Режим доступа: <https://www.raspberrypi.org/products/compute-module-3/> .- (Дата обращения: 10.05.19).
- 9.Raspberry Pi [Электронный ресурс]//Плата Compute Module 3+.-Режим доступа: <https://www.raspberrypi.org/products/compute-module-3-plus-32gb/> .- (Дата обращения: 10.05.19).
- 10.Raspberry Pi [Электронный ресурс]//Плата Compute Module 3+ Lite.-Режим доступа: <https://www.raspberrypi.org/products/compute-module-3-plus-lite/> .- (Дата обращения: 10.05.19).
- 11.Raspberry Pi [Электронный ресурс]//Процессор BCM2835.- Режим доступа: <https://www.raspberrypi.org/documentation/hardware/raspberrypi/bcm2835/README.md> .- (Дата обращения: 10.05.19).
- 12.Raspberry Pi [Электронный ресурс]//Процессор BCM2837.- Режим доступа: <https://www.raspberrypi.org/documentation/hardware/raspberrypi/bcm2837/README.md> .- (Дата обращения: 10.05.19).
- 13.Wikipedia the free encyclopedia [Электронный ресурс]//Список архитектур ARM.-Режим доступа:

https://ru.wikipedia.org/wiki/%D0%A1%D0%BF%D0%B8%D1%81%D0%BE%D0%BA%D0%B0%D1%80%D1%85%D0%B8%D1%82%D0%B5%D0%BA%D1%82%D1%83%D1%80_ARM .- (Дата обращения: 10.05.19).

14. Raspberry Pi [Электронный ресурс]//Raspberry Pi Camera Module V.2.-Режим доступа: <https://www.raspberrypi.org/products/camera-module-v2/> .- (Дата обращения: 10.05.19).

15. WaveShare [Электронный ресурс]//RPi Camera (D).-Режим доступа: <https://www.waveshare.com/rpi-camera-d.htm> .- (Дата обращения: 10.05.19).

16. Wikipedia the free encyclopedia [Электронный ресурс]//Список Linux-дистрибутивов для архитектуры ARM.- Режим доступа: https://en.wikipedia.org/wiki/Category:ARM_Linux_distributions .- (Дата обращения: 10.05.19).

17. Хартли Р., Зиссерман А. "Multiple View Geometry in computer vision"(2003).

18. Хабиби Х. А., Джахани Э. Х. "Guide to convolutional neural networks : a practical application to traffic-sign detection and classification."(2017).

19. Редмон Д. "You only look once: Unified, real-time object detection"//ArXiv.-2016.

20. Редмон Д. "Yolov3: An incremental improvement"//ArXiv.-2018.

21. Wikipedia the free encyclopedia [Электронный ресурс]//Карта глубин.-Режим доступа: https://en.wikipedia.org/wiki/Depth_map .- (Дата обращения: 10.05.19).

22. Керниган Б. "The UNIX Programming Environment"(1984).

23. GitHub [Электронный ресурс]//Информация, исходные коды и информация о лицензии проекта Darknet.-Режим доступа: <https://github.com/pjreddie/darknet> .- (Дата обращения: 10.05.19).

24. GitHub [Электронный ресурс]//Информация, исходные коды и информация о лицензии проекта Tensorflow.-Режим доступа: <https://github.com/tensorflow> .- (Дата обращения: 10.05.19).

25. GitHub [Электронный ресурс]//Информация, исходные коды и информация о лицензии проекта Darkflow.-Режим доступа: <https://github.com/thtrieu/darkflow> .- (Дата обращения: 10.05.19).

26. GitHub [Электронный ресурс]//Информация, исходные коды и информация о лицензии проекта AlexeyAB/Darknet.-Режим

доступа: <https://github.com/AlexeyAB/darknet> .- (Дата обращения: 10.05.19).

27.OpenCV [Электронный ресурс]//Главная страница.-Режим доступа: <https://opencv.org/> .- (Дата обращения: 10.05.19).

28.SourceForge [Электронный ресурс]//Информация, исходные коды, документация и информация о лицензии проекта PyOpenGL .-Режим доступа: <http://pyopengl.sourceforge.net/> .- (Дата обращения: 10.05.19).

29.PyGame [Электронный ресурс]//Главная страница.-Режим доступа: <https://pygame.org/> .- (Дата обращения: 10.05.19).

30.Darknet [Электронный ресурс]//YOLO: Real-Time Object Detection.-Режим доступа: <https://pjreddie.com/darknet/yolo/> .- (Дата обращения: 10.05.19).

31.University of Oxford [Электронный ресурс]//Hand dataset.-Режим доступа: www.robots.ox.ac.uk/~vgg/data/hands .- (Дата обращения: 10.05.19).

32.Habr [Электронный ресурс]//Изучаем OpenCV на StereoPi: карта глубин по видео.-Режим доступа: <https://habr.com/ru/post/446872/> .- (Дата обращения: 10.05.19).

33.GitHub [Электронный ресурс]//Информация, исходные коды и информация о лицензии проекта StereoVision.-Режим доступа: <https://github.com/erget/StereoVision> .- (Дата обращения: 10.05.19).

ПРИЛОЖЕНИЕ

1. Инструкция по обучению сети для распознавания рук с использованием фреймворка AlexeyAB/darknet и платформы Google Colaboratory:

1. Авторизоваться на сайте <https://www.google.com/drive/>;
2. С помощью контекстного меню по клику правой кнопкой мыши создать новый файл с использованием приложения Collaboratory;
3. Открыть созданный файл, дважды кликнув на нем левой кнопкой мыши;
4. Ввести и выполнить последовательность команд для установки и первичной настройки фреймворка:

```
%cd /content
!git clone https://github.com/AlexeyAB/darknet
%cd darknet
!sed '1s/0/1/; 4s/0/1/' Makefile > Makefile1
!cat Makefile1 > Makefile
!make
!wget https://pjreddie.com/media/files/yolov3-
tiny.weights
%cd /content/darknet
%cd cfg

!sed '3s/1/64/; 4s/1/8/; 177s/80/1/; 171s/255/18/;
135s/80/1/; 127s/255/18/' yolov3-tiny.cfg > tempCfg.cfg

!echo 'hand' > /content/darknet/data/obj.names

!echo -e 'classes= 1\ntrain =
/content/hand_dataset/training_dataset/training_data/images/train.
txt\nvalid =
/content/hand_dataset/training_dataset/training_data/images/test.t
xt\nnames = data/obj.names\nbackup = backup' >
/content/darknet/cfg/obj.data

%cd /content/darknet

!./darknet partial cfg/yolov3-tiny.cfg yolov3-tiny.weights
yolov3-tiny.conv.15 15
```


5. Ввести и выполнить последовательность команд для скачивания и распаковки набора данных:

```
%cd /content
```

```
!wget
```

```
http://www.robots.ox.ac.uk/~vgg/data/hands/downloads/hand_data  
set.tar.gz
```

```
!tar -xzf hand_dataset.tar.gz
```

```
%cd
```

```
/content/hand_dataset/training_dataset/training_data/
```

```
%cd
```

```
/content/hand_dataset/training_dataset/training_data
```

```
!mkdir labels
```

6. Выполнить скрипт, подготавливающий файлы аннотаций для их передачи во фреймворк. Листинг кода скрипта на языке Python3:

```
import os  
from lxml import etree  
import xml.etree.cElementTree as ET  
import cv2  
import scipy.io  
import numpy as np
```

```
for wrongAnnotation in
```

```
os.scandir('/content/hand_dataset/training_dataset/training_data/a  
nnotations'):
```

```
    if(wrongAnnotation.path.endswith('.mat') == False):
```

```
continue
```

```
    path = wrongAnnotation.path  
    path = path.replace('.mat', '.txt')  
    path = path.replace('annotations', 'images')  
    imgPath = path  
    imgPath = imgPath.replace('.txt', '.jpg')  
    imgName = os.path.basename(imgPath)
```

```
    image = cv2.imread(imgPath)  
    height, width, depth = image.shape  
    data = scipy.io.loadmat(wrongAnnotation.path)  
    for x in data.get('boxes')[0]:  
        xTemp = x[0][0]  
        bottom = np.min([y[0][0] for y in xTemp.item()[:4]])
```

```

left = np.min([y[0][1] for y in xTemp.item()[:4]])
right = np.max([y[0][1] for y in xTemp.item()[:4]])
top = np.max([y[0][0] for y in xTemp.item()[:4]])
if (left/width > 1): print('something wrong')
if (bottom/height > 1): print ('something wrong')

with open(path, 'a') as newAnnotation:
    newAnnotation.write('0 ' + str(left/width) + ' ' +
str(bottom/height) + ' ' + str((right - left)/width) + ' ' + str((top -
bottom)/height) + '\n')
    with
open('/content/hand_dataset/training_dataset/training_data/images
/train.txt', 'a') as trainFile:

trainFile.write('/content/hand_dataset/training_dataset/training_dat
a/images/' + imgName + '\n')

```

7. Запустить тренировку, выполнив команды:

```

%cd /content/darknet
!./darknet detector train cfg/obj.data cfg/tempCfg.cfg
yolov3-tiny.conv.15 -clear 1 -dont_show | grep -v '^Region'

```

8. Остановить тренировку, когда показатель Avg. Loss опустится до желаемого значения или перестанет снижаться.

9. Сохранить файл весов последней контрольной точки(/content/darknet/backup/tempCfg_last.weights) на свое устройство.

2. Инструкция по сборке фреймворка AlexeyAB/darknet в качестве подключаемой библиотеки для использования его в других программных продуктах в среде Raspbian:

1. Скачать и установить необходимые компоненты из репозитория apt посредством выполнения команд в терминале:

```

sudo apt install g++
sudo apt install build-essential
sudo apt install git

```

2. Скачать и скомпилировать фреймворк в качестве библиотечки, выполнив команды в терминале:

```

git clone https://github.com/AlexeyAB/darknet
cd darknet
sed '7s/0/1/; 4s/0/1/' Makefile > Makefile1
cat Makefile1 > Makefile
make

```

Файл libdarknet.so является библиотекой, которую можно подключать в коде других программ для использования функций фреймворка.

3. Установка необходимых модулей для создания карт глубин:

1. Установить менеджер модулей python 3 - pip, выполнив команду в терминале:

```
sudo apt install python3-pip
```

2. Загрузить необходимые модули, выполнив в терминале команды:

```
pip3 install opencv-python  
pip3 install matplotlib
```

4. Листинг кода для демонстрации построения карты глубин на языке Python3:

```
import matplotlib.pyplot as plt  
import cv2
```

```
left = cv2.imread('test1.jpg', cv2.IMREAD_COLOR)  
right = cv2.imread('test2.jpg', cv2.IMREAD_COLOR)
```

```
left = cv2.cvtColor(left, cv2.COLOR_RGB2GRAY)  
right = cv2.cvtColor(right, cv2.COLOR_RGB2GRAY)
```

```
stereo = cv2.StereoBM_create(numDisparities=128,  
blockSize=21)
```

```
disparity = stereo.compute(left,right)
```

```
norm_image = cv2.normalize(disparity, None, alpha = 0, beta  
= 256, norm_type=cv2.NORM_MINMAX, dtype=cv2.CV_8U)
```

```
plt.imshow(norm_image)
```

5. Установка необходимых библиотек для проецирования объекта и отображения на экране:

1. Выполнить в терминале команды:

```
pip3 install pygame  
pip3 install PyOpenGL
```

6. Листинг кода для демонстрационной сцены проецирования объекта:

1. Модуль загрузки трехмерного объекта из файла objloader.py:

```
import pygame
from OpenGL.GL import *

def MTL(filename):
    contents = {}
    mtl = None
    for line in open(filename, "r"):
        if line.startswith('#'): continue
        values = line.split()
        if not values: continue
        if values[0] == 'newmtl':
            mtl = contents[values[1]] = {}
        elif mtl is None:
            raise ValueError
        elif values[0] == 'map_Kd':
            # load the texture referred to by this declaration
            mtl[values[0]] = values[1]
            surf = pygame.image.load(mtl['map_Kd'])
            image = pygame.image.tostring(surf, 'RGBA', 1)
            ix, iy = surf.get_rect().size
            texid = mtl['texture_Kd'] = glGenTextures(1)
            glBindTexture(GL_TEXTURE_2D, texid)
            glTexParameteri(GL_TEXTURE_2D,
GL_TEXTURE_MIN_FILTER,
GL_LINEAR)
            glTexParameteri(GL_TEXTURE_2D,
GL_TEXTURE_MAG_FILTER,
GL_LINEAR)
            glTexImage2D(GL_TEXTURE_2D, 0, GL_RGBA, ix,
iy, 0,
GL_RGBA,
GL_UNSIGNED_BYTE, image)
        else:
            mtl[values[0]] = map(float, values[1:])
    return contents

class OBJ:
    def __init__(self, filename, swapyz=False):
        """Loads a Wavefront OBJ file. """
        self.vertices = []
```

```

self.normals = []
self.texcoords = []
self.faces = []
self.minx = None
self.maxx = None

material = None
for line in open(filename, "r"):
    if line.startswith('#'): continue
    values = line.split()
    if not values: continue
    if values[0] == 'v':
        if self.minx == None:
            self.minx = float(values[1])
        if float(values[1]) < self.minx:
            self.minx = float(values[1])
        if self.maxx == None:
            self.maxx = float(values[1])
        if float(values[1]) > self.maxx:
            self.maxx = float(values[1])

        v = list(map(float, values[1:4]))
        if swapyz:
            v = v[0], v[2], v[1]
        self.vertices.append(v)
    elif values[0] == 'vn':
        v = list(map(float, values[1:4]))
        if swapyz:
            v = v[0], v[2], v[1]
        self.normals.append(v)
    elif values[0] == 'vt':
        self.texcoords.append(list(map(float,
values[1:3])))

    elif values[0] in ('usemtl', 'usemat'):
        material = values[1]
    elif values[0] == 'mtllib':
        self.mtl = MTL(values[1])
    elif values[0] == 'f':
        face = []
        texcoords = []
        norms = []
        for v in values[1:]:
            w = v.split('/')

```

```

        face.append(int(w[0]))
        if len(w) >= 2 and len(w[1]) > 0:
            texcoords.append(int(w[1]))
        else:
            texcoords.append(0)
        if len(w) >= 3 and len(w[2]) > 0:
            norms.append(int(w[2]))
        else:
            norms.append(0)
        self.faces.append((face, norms, texcoords,
material))

```

```

    def get_data(self):
        return self.vertices, self.normals, self.faces,
self.minx, self.maxx

```

2. Основная программа:

```

import pygame
from pygame.locals import *

from OpenGL.GL import *
from OpenGL.GLU import *
from OpenGL.GLUT import *
import objloader
import math

backgroundVerts = (
    (-11, -16.5, -30., 0.0, 0.0),
    (-11, 16.5, -30., 0.0, 1.0),
    (11, 16.5, -30., 1.0, 1.0),
    (11, -16.5, -30., 1.0, 0.0)
)

global image

pygame.init()
display = (720, 1080)
sceneDisplay = pygame.display.set_mode(display,
DOUBLEBUF|      OPENGL)

image = pygame.image.load('image.png')

```

```

    texid = glGenTextures(1)
    glBindTexture(GL_TEXTURE_2D, texid)

    glLightModelfv(GL_LIGHT_MODEL_AMBIENT, (1.0, 1.0,
1.0, 1))

    glEnable(GL_LIGHTING)
    glEnable(GL_LIGHT0)
    glLightfv(GL_LIGHT0, GL_POSITION, (0.0, 0.5, -15.0))

    gluPerspective(45, display[0]/display[1], 0.1, 50.0)
    glTranslatef(0.0, 0.0, -10)

    glEnable(GL_DEPTH_TEST)
    glDepthFunc(GL_LESS)

    def background():
        glEnable(GL_TEXTURE_2D)
        glTexEnvf(GL_TEXTURE_ENV,
GL_TEXTURE_ENV_MODE,          GL_DECAL)

        glPushMatrix()
        width = image.get_width()
        height = image.get_height()

        textureData = pygame.image.tostring(image, "RGBA",
1)

        glTexImage2D(GL_TEXTURE_2D, 0, GL_RGBA, width,
height,
                    0, GL_RGBA, GL_UNSIGNED_BYTE,
textureData)

        glTexParameterf(GL_TEXTURE_2D,
GL_TEXTURE_WRAP_S,          GL_REPEAT)
        glTexParameterf(GL_TEXTURE_2D,
GL_TEXTURE_WRAP_T,          GL_REPEAT)
        glTexParameterf(GL_TEXTURE_2D,
GL_TEXTURE_MAG_FILTER, GL_LINEAR)
        glTexParameterf(GL_TEXTURE_2D,
GL_TEXTURE_MIN_FILTER, GL_LINEAR)

        glBegin(GL_QUADS)

```

```

    for v in backgroundVerts:
        glTexCoord2f(v[3], v[4])
        glVertex3f(v[0], v[1], v[2])
    glEnd()

    glPopMatrix()
    glDisable(GL_TEXTURE_2D)

global obj

def load_obj(filename):
    global obj
    model = objloader.OBJ(filename)
    obj = model.get_data()
    print(len(obj[2]))

def draw_obj_on_pixels(xF, yF, zF, xS, yS, zS, imgWidth,
imgHeight,
                    bckgMinX, bckgMaxX, bckgMinY,
bckgMaxY):

    bckgWidth = bckgMaxX - bckgMinX
    bckgHeight = bckgMaxY - bckgMinY

    xFInWindow = xF
    yFInWindow = imgHeight - yF

    normalizedXF = (xFInWindow/imgWidth ) * bckgWidth
+      bckgMinX
    normalizedYF = (yFInWindow/imgHeight) *
bckgHeight +      bckgMinY

    xSInWindow = xS
    ySInWindow = imgHeight - yS

    normalizedXS = (xSInWindow/imgWidth ) * bckgWidth
+      bckgMinX
    normalizedYS = (ySInWindow/imgHeight) *
bckgHeight + bckgMinY

    objMinX = obj[3]
    objMaxX = obj[4]

    objLen = objMaxX - objMinX

```



```

objWndLen = math.sqrt((normalizedXS -
normalizedXF)**2 +          (normalizedYS - normalizedYF)**2 +
(zS-zF)**2)

```

```

scale = objWndLen/objLen

```

```

zAngleR = math.atan( (normalizedYS - normalizedYF) /
(normalizedXS - normalizedXF) )
zAngle = math.degrees(zAngleR)

```

```

if (zS-zF) == 0.:
    yAngleR = -math.atan( (0.0000000000001) /
(normalizedXS -          normalizedXF) )
    else:
        yAngleR = -math.atan( (zS - zF) / (normalizedXS -
normalizedXF) )
        yAngle = math.degrees(yAngleR) % 360

draw_obj(scale, (normalizedXF, normalizedYF, 0.), (0,
yAngle,          zAngle))

```

```

def draw_obj(scale, move, rotation):

```

```

    scaled = scale_obj(scale)

```

```

    glPushMatrix()

```

```

    glTranslatef(move[0], move[1], move[2])

```

```

    glRotatef(rotation[0], 1.0, 0.0, 0.0)

```

```

    glRotatef(rotation[1], 0.0, 1.0, 0.0)

```

```

    glRotatef(rotation[2], 0.0, 0.0, 1.0)

```

```

    for face in scaled[2]:

```

```

        vertices, normals, texture_coords, material = face

```

```

        glBegin(GL_POLYGON)

```

```

        glColor3fv((0.9, 0.0, 0.0))

```

```

        for i in range(len(vertices)):

```

```

            if normals[i] > 0:

```

```

                glNormal3fv(scaled[1][normals[i] - 1])

```

```
        glVertex3fv(scaled[0][vertices[i] - 1])
    glEnd()
```

```
glPopMatrix()
```

```
def scale_obj(multiplier):
```

```
    res = [ [], [], [], 0, 0 ]
```

```
    res[1] = obj[1]
```

```
    res[2] = obj[2]
```

```
    res[3] = obj[3]
```

```
    res[4] = obj[4]
```

```
    for v in range(len(obj[0])):
```

```
        res[0].append([])
```

```
        for coord in range(len(obj[0][v])):
```

```
            res[0][v].append([])
```

```
            res[0][v][coord] = obj[0][v][coord] * multiplier
```

```
    return res
```

```
load_obj('LowPolyTry.obj')
```

```
xr=0.
```

```
yr=0.
```

```
zr=0.
```

```
angle = 2.0
```

```
xpos = 0.
```

```
ypos = 0.
```

```
while True:
```

```
    for event in pygame.event.get():
```

```
        if event.type == pygame.QUIT:
```

```
            pygame.quit()
```

```
            quit()
```

```
        glClear(GL_COLOR_BUFFER_BIT |
GL_DEPTH_BUFFER_BIT)
        background()
```

```
draw_obj_on_pixels(100, 450, -1, 540, 400, 0, 720,  
1080, -2.75, 2.75, -4.12, 4.12)  
pygame.display.flip()  
pygame.time.wait(10)
```