

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
«Южно-Уральский государственный университет  
(национальный исследовательский университет)»  
Высшая школа электроники и компьютерных наук  
Кафедра «Информационно-измерительная техника»

РАБОТА ПРОВЕРЕНА

Рецензент,  
к.т.н., доцент кафедры ИКТ  
\_\_\_\_\_ Н.В. Вдовина  
\_\_\_\_\_ 2017 г.

ДОПУСТИТЬ К ЗАЩИТЕ

Заведующий кафедрой,  
д.т.н., профессор  
\_\_\_\_\_ А.Л. Шестаков  
\_\_\_\_\_ 2017 г.

Разработка оптической системы локальной навигации

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА  
ЮУрГУ–120401.2017.071. ВКР

Руководитель проекта,  
доцент, к.т.н

\_\_\_\_\_ Д.А. Кацай  
\_\_\_\_\_ 2017 г.

Автор работы,  
студент группы КЭ-274

\_\_\_\_\_ Е.А. Грачев  
\_\_\_\_\_ 2017 г.

Нормоконтролер,  
доцент

\_\_\_\_\_ А.А. Лысова  
\_\_\_\_\_ 2017г.

Челябинск 2017

## АННОТАЦИЯ

Грачев Е.А. Разработка оптической системы локальной навигации. – Челябинск: ЮУрГУ, КЭ; 2017, 140 с., 38 ил., библиогр. список – 38 наим., 9 прил., 1 лист ф. А2, 1 лист ф. А3

Работа посвящена разработке оптической системы локальной навигации мобильного робота в городской среде, в состав которой входит система технического зрения, основанная на методе, использующем геометрические подсказки для анализа сцены изображения.

Разработаны математическая модель преобразования оптических сигналов, функциональная схема системы, плата приемника излучения, используя современную элементную базу электронных компонентов и современных технологий проектирования печатных плат, корпус системы, алгоритм и программное обеспечение, реализующее вышеупомянутый метод. Рассчитаны пороговые значения потока излучения и освещенности в энергетических и световых величинах для приемника излучения, используемого в разработанной системе. На основе временной оценки вычислительных затрат работы алгоритма подобран процессорный модуль. Также, подобран модуль питания отечественного производства для применения в разработанной системе.

При разработке использованы современные программные средства проектирования и моделирования такие, как MatLab R2016b, VisualStudio 2010, SolidWorks 2016 и AltiumDesigner 2016.

					12.04.01.2017.071. ВКР			
Изм	Лист	№ докум	Подпись	Дата				
Разраб.	Грачев Е.А.				Разработка оптической системы локальной навигации	Литера	Лист	Листов
Пров.	Кацай Д.А.						3	140
Рец.	Вдовина Н.В.					ЮУрГУ Кафедра ИНИТ		
Н. Контр.	Лысова А.А.							
Утв	Шестаков А.Л.							

## ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	7
1 ОБЗОР СОВРЕМЕННЫХ МЕТОДОВ ПРОЕКТИРОВАНИЯ ОПТИЧЕСКИХ СИСТЕМ НАВИГАЦИИ.....	9
1.1 АНАЛИЗ ТЕХНИЧЕСКОГО ЗАДАНИЯ.....	9
1.2 ОБЗОР ОПТИЧЕСКИХ ПРИБОРОВ, КОМПЛЕКСОВ И СИСТЕМ НАВИГАЦИИ .....	10
1.2.1 Оптический датчик координат .....	10
1.2.2 Цифровая камера RT-1000DC.....	18
1.3 ОБЗОР МЕТОДОВ АНАЛИЗА ГОРОДСКИХ СЦЕН НА ИЗОБРАЖЕНИИ .....	21
1.3.1 Методы, основанные на гауссовой сфере.....	21
1.3.2 Методы оценки поворота камеры .....	25
1.3.3 Методы, основанные на EM-алгоритме.....	27
1.3.4 Методы, основанные на RANSAC-алгоритме .....	29
1.3.5 Неитеративный метод быстрой и точной оценки точек схода.....	30
1.4 ОБЗОР МЕТОДОВ РАСПОЗНАВАНИЯ ОБЪЕКТОВ .....	32
1.4.1 Метод поиска максимума кросскорреляционной функции.....	32
1.4.2 Обнаружение устойчивых признаков изображения: метод SURF .....	34
2 ТЕОРЕТИЧЕСКАЯ РАЗРАБОТКА СИСТЕМЫ .....	37
2.1 МАТЕМАТИЧЕСКАЯ МОДЕЛЬ ПРЕОБРАЗОВАНИЯ ОПТИЧЕСКИХ СИГНАЛОВ .....	38
2.2 РАСЧЕТ ПОРОГОВЫХ ЗНАЧЕНИЙ ПОТОКА ИЗЛУЧЕНИЯ И ОСВЕЩЕННОСТИ.....	41
2.3 РАЗРАБОТКА ФУНКЦИОНАЛЬНОЙ СХЕМЫ СИСТЕМЫ.....	53
2.4 РАЗРАБОТКА ПЕЧАТНОЙ ПЛАТЫ МАТРИЧНОГО ПРИЕМНИКА ИЗЛУЧЕНИЯ .....	54
2.5 ПЛАТА ЦИФРОВОЙ ОБРАБОТКИ ИЗОБРАЖЕНИЯ .....	68
2.6 ПЛАТА ПИТАНИЯ .....	73
2.7 РАЗРАБОТКА КОРПУСА СИСТЕМЫ.....	79
3 РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ОБРАБОТКИ ЗРИТЕЛЬНЫХ ДАННЫХ.....	84
3.1 ОПИСАНИЕ МОДЕЛИ .....	84

					ЮУрГУ-120401.2017.071. ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		3

3.2 АЛГОРИТМ ВЫВОДА ДЛЯ МОДЕЛИ .....	89
3.3 ТЕХНИЧЕСКИЕ ДЕТАЛИ РЕАЛИЗАЦИИ .....	94
4 АНАЛИЗ ЭКСПЕРИМЕНТАЛЬНЫХ ДАННЫХ.....	98
ЗАКЛЮЧЕНИЕ .....	103
БИБЛИОГРАФИЧЕСКИЙ СПИСОК .....	105
ПРИЛОЖЕНИЯ	
ПРИЛОЖЕНИЕ А. АЛГОРИТМ ВЫВОДА ДЛЯ ГЕОМЕТРИЧЕСКОГО ПАРСИНГА.....	109
ПРИЛОЖЕНИЕ Б. ТЕКСТ ПРОГРАММЫ ОБРАБОТКИ ИЗОБРАЖЕНИЯ .....	111
ПРИЛОЖЕНИЕ В. ТЕКСТ ФУНКЦИЙ ОБРАБОТКИ ИЗОБРАЖЕНИЯ.....	120
ПРИЛОЖЕНИЕ Г. Спецификация.....	127
ПРИЛОЖЕНИЕ Д. Сборочный чертеж системы.....	128
ПРИЛОЖЕНИЕ Е. Электрическая схема сенсорного модуля.....	129
ПРИЛОЖЕНИЕ Ж. Перечень элементов сенсорного модуля.....	130
ПРИЛОЖЕНИЕ З. Схема включения модуля питания.....	139
ПРИЛОЖЕНИЕ И. Перечень элементов модуля питания.....	140

## СПИСОК ИСПОЛЬЗУЕМЫХ СОКРАЩЕНИЙ

АЦП – аналого-цифровой преобразователь

АЧТ – абсолютно черное тело

ВИП – вторичный источник питания

ДПП – двусторонняя печатная плата

ДКВ – двойная коррелированная выборка

ИЭТ – изделие электронной техники

КПД – коэффициент полезного действия

КМОП – комплементарная структура металл-оксид-полупроводник

МОП – структура металл-оксид-полупроводник

МК – микроконтроллер

МПП – многослойная печатная плата

МПИ – матричный приемник излучения

НИР – научно-исследовательская работа

ОС – оптическая система

ОЗУ – оперативное запоминающее устройство

ОКР – опытно-конструкторская работа

ПП – печатная плата

ПЛИС – программируемая логическая интегральная схема

ПМК – поверхностно-монтируемые компоненты

ПЗС – прибор с зарядовой связью

РЭС – радиоэлектронная система

САПР – система автоматического проектирования

СПФ – сухой пленочный фоторезист

СЭП – система электрического питания

СТЗ – система технического зрения

СКО – среднеквадратическое отклонение

Т-система – техническая система

					ЮУрГУ-120401.2017.071. ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		5

ФУ – функциональное устройство

ЦАП – цифро-аналоговый преобразователь

ШИМ – широтно-импульсная модуляция

ЭА – электронная аппаратура

ЭВМ – электронно-вычислительная машина

ЭРИ – электрорадиоизделие

Bit-Bang – технология организации последовательного соединения с использованием программной эмуляции

CPU (central processing unit) – центральное обрабатывающее устройство

EM-алгоритм (expectation-maximization) – алгоритм для нахождения оценок максимального правдоподобия

GPU (graphics processing unit) – графический процессор

I2C (inter-integrated circuit) – 8-битная шина внутрисхемной связи

MEX-функция (matlab extension) – функция, расширяющая набор функций matlab

PDA (personal digital assistant) – портативное вычислительное устройство

RANSAC-алгоритм (random sample consensus) – метод оценки параметров модели на основе случайных выборок

SURF-метод (speeded up robust features) – метод описания характерных точек на изображении

SPI (serial peripheral interface) – последовательный периферийный интерфейс

USB (universal serial bus) – универсальный последовательный интерфейс

					ЮУрГУ-120401.2017.071. ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		6

## ВВЕДЕНИЕ

Информационно-измерительные системы для робототехнических систем являются той неотъемлемой частью, которая играет очень важную роль, а именно позволяет взаимодействовать в информационном плане с внешней средой.

По своему строению и принципам функционирования оптические системы локальной навигации, которые включает в себя СТЗ отличаются от остальных датчиков информационно-измерительных, навигационных систем. В современном мире датчики робототехнических систем детально изучены, а исследования в этой сфере направлены на улучшение их функциональности за счет расширения рабочего диапазона технических параметров. Исследования же СТЗ коренным образом отличаются от вышеупомянутых систем. Это связано с модальностью той информации, которую призвана фиксировать СТЗ. Субъективность восприятия зафиксированных данных приводит к возникновению определенных трудностей у разработчика при построении таких систем. Не меньшую проблему создает отсутствие универсальной модели любого объекта наблюдения, описывающей одновременно все его аспекты. Они же являются фундаментальными проблемами не только искусственного интеллекта и информатики, но ряда смежных наук: когнитивная психология, нейрофизиология, геометрия и т.п.

Однако вернемся к проблеме, которую планируется разрешить в рамках выпускной квалификационной работы, что обуславливает ее актуальность. Эта проблема заключается в творческом подходе к созданию СТЗ. В области информационно-измерительных систем создание каждой СТЗ требует творческого подхода. Впервые эта идея была предложена А.Р. Исхаковым в целой серии изданных работ по модифицированным дескриптивным алгебрам изображений [1]. Эти работы построены на основе научных работ академика Ю.И. Журавлева и его школы по дескриптивному подходу к обработке, анализу и пониманию изображений. В данной работе приведено проектирование и реализация программного обеспечения, направленного на создание СТЗ,

					ЮУрГУ-120401.2017.071. ВКР	Лист
						7
Изм.	Лист	№ докум.	Подпись	Дата		

нацеленной для применения в системах навигации мобильных робототехнических систем в городской среде, а также разработка устройства регистрации и обработки зрительных данных.

Существующие подходы к решению задачи навигации мобильного робота в городской среде можно разделить на два больших класса [2]: методы, использующие семантические подсказки, представленные на изображении, и методы, основанные на геометрических подсказках. В первой группе методов в основном используется цветовая информация о пикселях изображения. Данную работу можно отнести к методам, использующих геометрические подсказки. В условиях городской среды, регистрируемое изображение содержит большое количество вертикальных и горизонтальных плоскостей, таких как дорога или стены. Прямые могут быть образованы разметкой на асфальте, вертикальными столбами, рядом окон на здании и другими предметами, находящимися в сцене изображения. Прямые, лежащие на одной плоскости, образуют при этом группу параллельных прямых, которую можно применить в качестве подсказки о геометрии сцены.

Объектом исследования выпускной квалификационной работы является оптическая система локальной навигации в городской среде для мобильного робота.

Предмет исследования заключается в возможности использования метода, основанного на геометрических подсказках для решения задачи локальной навигации мобильного робота.

Цель работы – разработка оптической системы локальной навигации, работа которой основана на вышеупомянутом методе.

Поставленная цель достигается решением следующих задач:

- 1) обзор современных методов проектирования оптических систем навигации;
- 2) теоретическая разработка системы;
- 3) разработка программного обеспечения обработки зрительных данных;
- 4) анализ экспериментальных данных.

					ЮУрГУ-120401.2017.071. ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		8



# 1 ОБЗОР СОВРЕМЕННЫХ МЕТОДОВ ПРОЕКТИРОВАНИЯ ОПТИЧЕСКИХ СИСТЕМ НАВИГАЦИИ

## 1.1 Анализ технического задания

Исходя из требований, предъявляемых в техническом задании на выпускную квалификационную работу можно сделать вывод, что для корректного решения поставленной задачи необходимо провести информационный поиск по теме «компьютерное зрение».

В наиболее широком смысле, компьютерное зрение – теория и технология создания машин, которые могут производить обнаружение, отслеживание и классификацию объектов.

Также, в задании определены основные функции разрабатываемой системы: поиск горизонта и распознавание объекта. Однако, для реализации этих функций необходимо решить следующие задачи:

1) получение изображения – необходимо определиться с типом приемника оптического излучения (значение пикселей МПИ обычно соответствуют интенсивности света в одной или нескольких спектральных полосах – цветные или изображения в оттенках серого);

2) предварительная обработка – перед тем, как методы компьютерного зрения могут быть применены к видеоданным с тем, чтобы извлечь определённую долю информации, необходимо обработать видеоданные, с тем чтобы они удовлетворяли некоторым условиям, в зависимости от используемого метода;

3) выделение деталей – геометрические примитивы различного уровня выделяются из видеоданных;

4) детектирование/сегментация – принимается решение о том, какие точки или участки изображения являются важными для дальнейшей обработки.

Также, в задании оговорен тип системы – «интеллектуальная» камера. Отсюда следует, что получение, обработка изображения и принятие решения на основе обработки происходит в одном устройстве. Для этого целесообразно применить

					ЮУрГУ-120401.2017.071. ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		9

модульную структуру системы: модуль МПИ; процессорный модуль; модуль питания.

## **1.2 Обзор оптических приборов, комплексов и систем навигации**

При разработке технической системы необходимо рассмотреть аналоги в современном научно-техническом сообществе и выбрать в качестве прототипа наиболее близкое решение.

В данном разделе приводится обзор двух современных устройств машинного зрения наиболее близких в техническом плане к разрабатываемой системе.

### *1.2.1 Оптический датчик координат*

Разработанное в ИКИ РАН устройство – оптический датчик координат – предназначено для координатных измерений положения подвижного объекта, оснащенного таким датчиком, относительно неподвижной посадочной плоскости с топографическими маркерами, образующими геометрическую фигуру на этой плоскости [3].

Ближайшими аналогами по назначению и применяемым методам могут служить разработки, ведущиеся в МГТУ им. Баумана [4]. Специалистами НИИ специального машиностроения МГТУ была создана телевизионная система определения координат. Вычисления относительных координат производились по изменению размера и формы геометрической фигуры – кольца, нанесенного на поверхность посадочной плоскости в месте стыковки подвижного подводного аппарата. Результаты этой исследовательской работы представлены в нескольких публикациях [4, 5].

Схема метода измерений координат представлена на рисунке 1.

					ЮУрГУ-120401.2017.071. ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		10



Общая задача управления фактически заключается в том, чтобы совместить оси двух систем координат. Задача управления выполняется при реализации последовательности следующих процедур:

1) определение пространственного положения объекта и углов поворотов относительно неподвижной системы координат;

2) определение в каждый момент времени сглаженных значений линейных и угловых скоростей и координат;

3) построения закона терминального управления;

4) определение командных сигналов для выдачи на управление движителями аппарата.

В состав аппаратных средств системы определения координат входят оптический датчик, система освещения, топографическая мишень и вычислительные устройства. Главная задача аппаратной части системы заключается в получении качественного изображения топографической мишени. Качество изображения, получаемого в водной среде, определяется дальностью видимости отражающих объектов [4].

Структурная схема датчика координат аналогична схемам других систем машинного зрения, использующих топографическую мишень. В состав прибора входят: объектив, проецирующий изображения, платы фотоприемной матрицы, платы процессора и памяти, источник вторичного питания и порт ввода-вывода информационных потоков и сигналов. Конструктивно все узлы размещены в одном корпусе. Датчик координат является съемным устройством и крепится за иллюминатором люка шлюзовой камеры. Внешний вид датчика представлен на рисунке 2, функциональная схема приведена на рисунке 3.



Рисунок 2 – Внешний вид оптического датчика координат

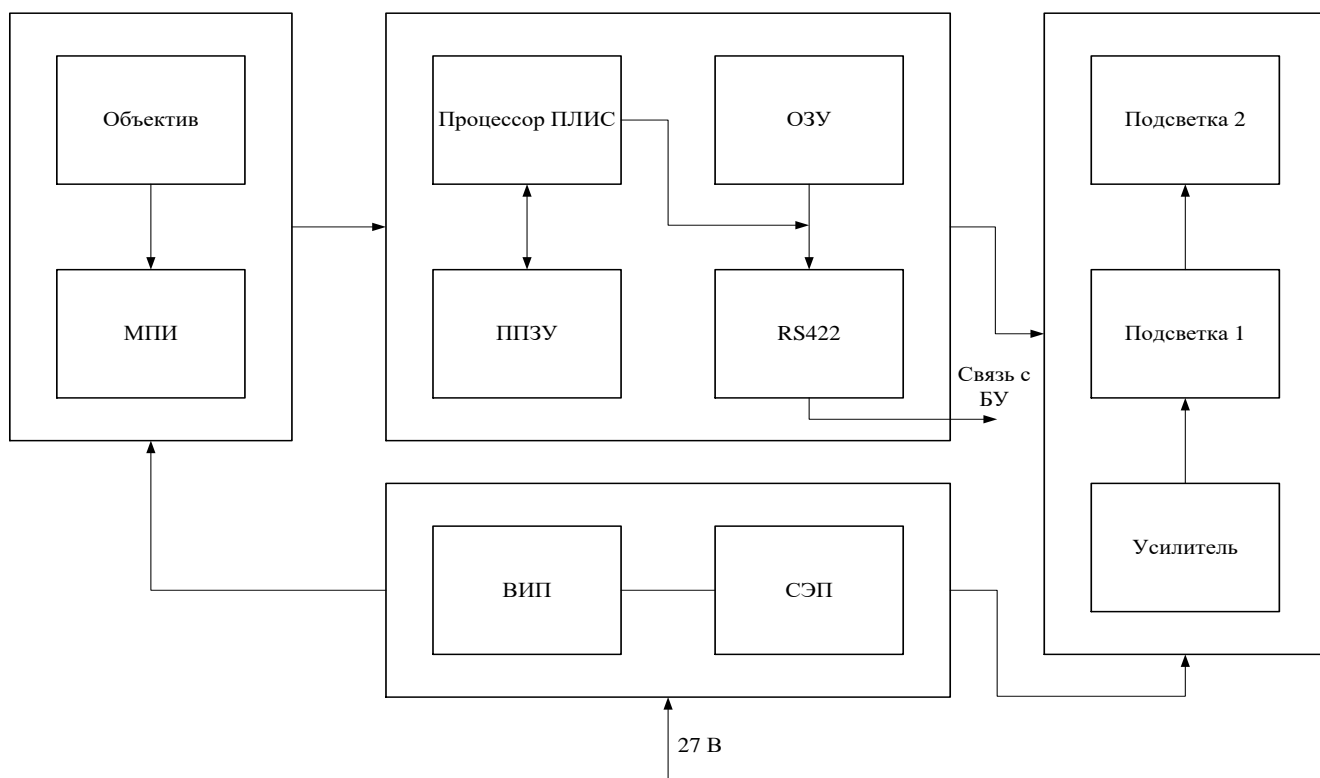


Рисунок 3 – Функциональная схема оптического датчика координат

В схемной реализации прибора можно отметить следующие особенности:

1) используется светочувствительная матрица КАС-9638 фирмы Kodak с числом используемых пикселей 1024x1024 при размере пикселя 6 мкм; матрица

Изм.	Лист	№ докум.	Подпись	Дата

изготовлена по КМОП-технологии, обеспечивающей произвольный доступ к части изображения;

2) используется программируемая матрица на основе ОЗУ фирмы Xilinx, позволяющая многократно переконфигурировать прибор в процессе отработки, а также минимизировать число электронных компонентов.

Одна из важных особенностей аппаратного обеспечения заключается в том, что оптический датчик оснащен сигнальным процессором и обработка изображения происходит в самом датчике. На вычислительное устройство системы автоматической посадки передаются только координаты изображения маркеров в приборной системе координат и прореженное изображение для отображения и визуального контроля процесса посадки. Датчик координат работает автономно, не перегружая вычислительные мощности системы управления посадки, тем самым повышая надежность всей системы.

Другой особенностью оптического датчика служит то, что в процессе поиска и анализа изображения в датчике автоматически подбираются оптимальное время экспозиции кадра и коэффициент усиления, по внутренним алгоритмам, связанным с максимальной помехоустойчивостью. Любопытно, что изображение сцены, передаваемое для визуализации на экран монитора, не соответствует оптимальному восприятию человеческим глазом и требует дополнительной гамма-коррекции, балансировки контрастности и яркости. Такая коррекция делается программным обеспечением вычислительного устройства систем отображения информации.

Алгоритмы работы, последовательность действий, критерии обнаружения маркеров мишени и методы вычисления центров координат составляют программное обеспечение датчика.

Изложение аппаратных алгоритмов и последовательность программного цикла составляют полное описание логики работы прибора, в котором далее выделим основные, ключевые моменты.

					ЮУрГУ-120401.2017.071. ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		14

Прежде всего основой для применения алгоритмов служит получение первоначального кадра изображения с максимально возможным качеством. Для достижения качества изображения на предварительном этапе работы камеры, после ее включения, производится первоначальная регулировка усиления и выбирается время экспозиции. В камере устанавливаются некоторые начальные значения времени экспозиции и коэффициента усиления. Эти значения определяются по результатам калибровок при настройке взаимодействия камеры с конкретной системой освещения. Выбор времени экспозиции производится по результату анализа гистограммы распределения освещенности, вычисляемой по всем пикселям матрицы. Гистограмма распределения интенсивности пикселей необходима для нормирования образа и обеспечения правильного выбора динамического диапазона. По гистограмме также определяется средний уровень шума фонового изображения. Заметим, что допускается насыщение (достижение максимальных значений интенсивности) для некоторого, но строго определенного объема гистограммы. Такое допущение позволяет работать датчику при наличии источников паразитных бликов, образованных блестящими деталями конструкции, стеклянными и полированными металлическими поверхностями либо другими неучтенными источниками света.

После получения кадра производится его анализ на предмет обнаружения пикселей, содержащих полезный сигнал и обнаруживаемых по критерию превышения заданного соотношения сигнал/ шум. Вообще, в приборе это соотношение задается несколькими дискретными значениями и изменяется в зависимости от состояния помеховой обстановки.

Уровень шума определяется по плавающему окну. Пиксель, содержащий полезный сигнал, но являющийся одиночным, т.е. не граничащий с подобными, считается шумовым и отбрасывается. Такая отбраковка диктуется априорным знанием о геометрических размерах изображений маркеров в заданном диапазоне дальностей до мишени. Также отбрасываются из рассмотрения фигуры – объекты,

					ЮУрГУ-120401.2017.071. ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		15

образованные группами пикселей с линейными размерами, значительно превышающими размеры изображений маркеров с учетом возможных искажений.

После селекции групп пикселей по размерам формируется массив объектов-образов. Число таких объектов, предназначенных для последующего анализа, ограничено сверху вычислительными возможностями прибора и выбирается экспериментально. При превышении количества объектов дается команда на изменение настроек камеры, реализуемых последовательно. Производится уменьшение времени экспозиции, коэффициента усиления или увеличения порогового уровня соотношения сигнал/ шум. В случае если объектов селекции меньше четырех, т.е. числа, необходимого для измерений дальности, выдается команда на противоположные действия.

Второй важный этап фильтрация массива образов основан на анализе формы образов. Очевидно, что в заданном диапазоне дальностей образы маркеров имеют вид эллипсов соответствующих размеров с учетом допустимых относительных углов наклона плоскости и дисторсии объектива. На данном этапе происходит определение полуосей имеющихся фигур, каждому отдельному образу приписывается числовой параметр, отражающий соотношение полуосей. Образу также приписывается второй числовой параметр – энергетический. Этот параметр соответствует количеству энергии в образе, т.е. это нормированная сумма интенсивностей каждого пикселя в образе. Затем вычисляются координаты центра каждого образа. Координаты центра образа вычисляются как координаты центра тяжести плоской фигуры. Плоская фигура задается контуром массива пикселей образа со значением интенсивности в каждом пикселе, равном среднему значению по образу. Физически это обоснованно тем, что профиль интенсивности идеального изображения маркера имеет плоскую вершину и резкий спад на краях.

Третий важный этап в работе алгоритмов – это объединение отдельных образов в группы. Группировка осуществляется по принципу подобия образов в части совпадения числовых параметров соотношения полуосей и энергетического параметра. Границы интервалов совпадений определяются из физических

					ЮУрГУ-120401.2017.071. ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		16



соображений и подстраиваются экспериментально. Часть образов с параметрами, не попадающими в геометрически разумный диапазон, отбраковывается.

Четвертый этап в обработке – это анализ групп образов на геометрическое подобие заданной (априорной) фигуре. При анализе используется критерий нахождения центров трех образов на одной прямой. Для произвольной пары точек, соответствующих центрам образов маркеров, находится уравнение прямой и проверяется попадание других точек на эту прямую с допуском, определяемым неточностью оптической системы и приближенным характером вычисления положений центра образов. Формируется массив прямых, содержащих по три точки, затем решаются попарно уравнения, определяющие координату пересечения прямых, и проверяется наличие в окрестности точки пересечения центра образа маркера. В случае нахождения двух таких пересекающихся прямых задача по отождествлению образа считается решенной. Вообще, для полного определения координат достаточно знать координаты четырех маркеров в приборной системе координат. Пять маркеров дают однозначность в определении образа и обладают избыточной информацией, что позволяет увеличить точность измерений. Наличие семи маркеров в мишени позволяет увеличить точность измерений и повысить надежность распознавания образов, гарантирует работоспособность системы в случае загрязнения, повреждения или утраты двух любых произвольных маркеров.

Пятый этап обработки заключается в подборе оптимального уровня экспозиции и коэффициента усиления, обеспечивающих максимальное соотношение сигнал/шум в окрестности образов маркеров. Производятся уточненные измерения формы и координат центров маркеров с учетом поправки на дисторсию. Вычисляются покадровые сдвиги центров маркеров для прогноза движения и оценки смещения. Вычисленные координаты центров маркеров передаются во внешнее устройство. Одновременно включается режим захвата и адаптивного слежения за обнаруженными маркерами. В данном режиме нет необходимости обрабатывать весь кадр, для обработки достаточно небольших

					ЮУрГУ-120401.2017.071. ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		17

окон изображения вокруг обнаруженных маркеров с учетом покадрового сдвига. КМОП-матрица и управляющая логика позволяют организовывать произвольный доступ к выделенным областям матрицы, снижая время чтения и объем обрабатываемых данных. Уменьшение времени считывания и обработки кадра позволяет повысить частоту выдачи полезной информации до величин, обеспечивающих динамику управления.

### 1.2.2 Цифровая камера RT-1000DC

Цифровая камера RT-1000 предназначена для получения высококачественного черно-белого прогрессивного телевизионного изображения, ввода в ПЭВМ и цифровой обработки его в реальном времени. Камера позволяет выводить получаемое изображение на стандартный телевизионный монитор с чересстрочной разверткой.

Камера состоит из следующих узлов: блока фотоприемника, кабеля связи и платы управления с цифровым интерфейсом RT-644, устанавливаемой в компьютере. Передача изображения производится на расстояние до 10 м с использованием канала связи Channel Link (IEEE-644).

Внешний вид камеры в прямоугольном корпусе с объективом «Гелиос44» изображен на рисунке 4, функциональная схема приведена на рисунке 5



а)



б)

Рисунок 4 – Внешний вид цифровой камеры RT-1000DC:

а – вид спереди; б – вид сзади

Изм.	Лист	№ докум.	Подпись	Дата



Рисунок 5 – Функциональная схема цифровой камеры RT-1000DC

Видеокамера RT-1000DC состоит из следующих основных блоков и узлов:

1) МПИ – ПЗС матрица. Имеет два аналоговых выхода считывания заряда: X и Y. Каждый из выходов подключен к области матрицы размером 512x1024 элементов;

2) схема ДКВ – каскады, осуществляющие двойную коррелированную выборку преобразование выходного сигнала матрицы в видеосигнал для последующего аналого-цифрового преобразования;

3) нормирующий видеоусилитель – устраняет рассогласование усиления в каналах X и Y, возникающее из-за технического разброса параметров и обеспечивает согласование уровня видеосигнала с динамическим диапазоном АЦП;

4) драйверы считывания заряда – преобразуют цифровые тактовые сигналы управления в сигналы с необходимыми для ПЗС матрицы уровнями напряжения;

5) АЦП – 12-ти разрядный аналого-цифровой преобразователь;

6) блок цифровой обработки сигналов – обеспечивает динамическое и статическое выравнивание каналов X и Y, выполняет DSP обработку изображения в реальном времени;

7) внутренний буфер памяти (8 МБ) – функционально разделен на 4 банка по 2 МБ. Используется для хранения промежуточных результатов при цифровой обработке, а также для вывода изображения на внешний монитор;

8) подсистема вывода изображения – осуществляет преобразование прогрессивного изображения матрицы в стандартный ТВ сигнал и обеспечивает синхронизацию ввода и вывода;

9) ЦАП – 10-ти разрядный цифро-аналоговый преобразователь;

10) генератор ТВ сигнала – формирует аналоговый сигнал по стандарту CCIR;

11) интерфейс связи IEEE 644 – обеспечивает связь фотоприемника с платой управления по стандарту Chanel Link;

12) схема внешнего запуска – синхронизирует ввод изображения с внешним сигналом начала кадра. Имеет оптическую развязку.

Цифровая камера RT-1000 DC имеет следующие основные режимы работы:

1) режим внутренней синхронизации, при котором осуществляется непрерывный ввод телевизионных кадров в компьютер с одновременным контролем вводимого изображения на внешнем мониторе (ВКУ). Частота ввода кадров составляет 25 Гц (50 Гц при объединении строк) или определяется временем экспозиции, если это время превышает 40 мс (20 мс при объединении строк);

2) режим внешней синхронизации, при котором осуществляется одиночный или потоковый ввод телевизионных кадров в компьютер с одновременным контролем телевизионного изображения на ВКУ. Сигнал внешней синхронизации должен иметь частоту запуска не более 25 Гц (50 Гц при объединении строк);

3) режим передачи тестового изображения, при котором осуществляется проверка канала связи между блоком фотоприемника и платой управления (ПЗС матрица не задействована).

### **1.3 Обзор методов анализа городских сцен на изображении**

Задача поиска геометрических примитивов для анализа геометрии городских сцен довольно хорошо изучена, о чём говорит обилие статей по данной тематике. Несмотря на большое количество опубликованных методов, пока не существует никакой общепринятой классификации. Однако, можно выделить некоторые характеристики для классификации методов:

1) предположение Манхэттэнского мира: нахождение трёх точек схода, соответствующих трём взаимно перпендикулярным направлениям / нахождение произвольного количества точек схода;

2) параметры камеры: использование внутренних параметров камеры / использование неоткалиброванной камеры;

3) используемые геометрические примитивы: использование карты градиентов / карты краёв в качестве входных данных, определение отрезков / прямых.

Рассмотрим основные группы методов, предложенных для нахождения различных геометрических примитивов и оценки геометрии городских сцен. Методы сгруппированы по схожести подхода, лежащего в основе решения задачи. При этом для каждого метода даётся характеристика, согласно выделенным признакам.

#### *1.3.1 Методы, основанные на гауссовой сфере*

Одним из базовых методов к определению точек схода основан на использовании гауссовой сферы, предложенной впервые в работе Barnard 1983 [6, 7]. Иллюстрация этого метода представлена на рисунке 6.



Представление плоскости изображения гауссовой сферой оказалось настолько удачной идеей, что в дальнейшем было предложено очень много методов, которые используют различные вариации гауссовой сферы.

Главная проблема в первых подходах, использующих гауссову сферу, заключалась в поточности вычислений, связанных с дискретизацией сферы па ячейки. В следующей работе – Collins, Weiss 1990 [9, 10], использующей гауссову сферу, этот недостаток был устранён. Авторы предложили статистический подход к решению данной задачи, в котором при вычислении используется исключительно произведение векторов, что не ведёт к снижению точности.

Метод работает с откалиброванной камерой, также подразумевается, что с самого начала известны сгруппированные по направлениям семейства отрезков. Таким образом, единственная задача, решаемая методом – оценка точек схода по заданным группам отрезков.

Метод основан на наблюдении, что нормали плоскостей, образованных отрезками из одной группы и оптическим центром, формируют в свою очередь большую окружность на сфере, и задача нахождения точки схода сводится к нахождению нормали к этой плоскости (Рисунок 7).

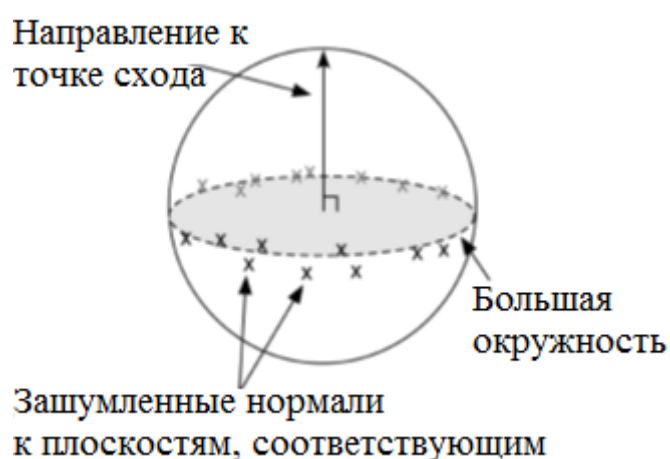


Рисунок 7 – Распределение нормалей, соответствующих прямым

Нормали окружностей, соответствующих отрезкам из одной группы, представляются как случайные величины на поверхности сферы. Далее делается предположение, что зашумлённые нормали имеют экваториальное распределение относительно положения истинной окружности. Далее производится оценка максимального правдоподобия направления точки схода. Эта задача сводится к нахождению собственных векторов матрицы:

$$T = \sum_{i=1}^n x_i x_i^T, \quad (1.1)$$

где  $x_i$  – единичная нормаль, соответствующая  $i$ -ому отрезку из данной группы.

Ответом к задаче будет служить собственный вектор, соответствующий наименьшему собственному значению матрицы  $T$ .

У этого метода присутствует ряд достоинств: нет никакой дискретизации, вес вычисления производятся в непрерывном пространстве, при этом вычисления нетрудоёмки: используется только вычисление собственных векторов матрицы размера  $3 \times 3$ . Однако этот метод решает только часть задачи: определение точек схода по группам отрезков. Метод хорошо применим для нахождения точек схода на размеченных данных.

Ещё один способ применения гауссовой сферы представлен в статье Boulanger 2006 [11]. Данный алгоритм подразумевает, что выполнено предположение Манхэттэнского мира (т.е. находятся не более трёх точек схода, соответствующих трём взаимно перпендикулярным направлениям), также предполагается, что все внутренние параметры камеры известны.

Алгоритм инициализируется картой краёв, посчитанной для входного изображения. Далее с использованием модифицированного преобразования Хафа находятся основные прямые изображения, и затем происходит группировка линий и вычисление соответствующих точек схода на модифицированной гауссовой сфере.



Гауссова сфера используется в качестве аккумуляторной сетки, как и в первоначальных методах. Основные отличия метода заключаются в двух модификациях сферы:

1) вместо сферы используется полусфера с диаметром, равным наименьшей из сторон изображения, касающаяся середины изображения (для ускорения вычислений);

2) предлагается новый способ деления сферы на ячейки таким образом, чтобы площадь ячеек была одинаковой (для повышения точности).

Таким образом, в данном методе полусфера делится на ячейки специальным образом, для каждой прямой вычисляется уравнение полуокружности, соответствующей данной прямой на полусфере, и для каждой ячейки, через которую проходит данная полуокружность, прибавляется 1. Далее точки схода находятся как направления, соответствующие ячейкам полусферы с максимальным значением.

Модификация алгоритма позволила работать с такими сложными изображениями, как фотографии, сделанные на природе и нарисованные картины. По сравнению с классической гауссовой сферой, данный метод обладает большей точностью и работает быстрее.

### *1.3.2 Методы оценки поворота камеры*

В следующих методах решается задача нахождения поворота камеры, что эквивалентно нахождению трёх точек схода, соответствующих взаимно перпендикулярным направлениям. Один из первых алгоритмов, решающих данную задачу в такой формулировке, описан в [12].

Главная особенность алгоритма заключается в том, что он не требует никакой предобработки, такой как определение отрезков или преобразование Хафа. Однако подразумевается, что известны параметры камеры.

Алгоритм рассматривает карту градиентов для всех пикселей изображения. При этом считается, что каждый пиксель может быть отнесён к одному из пяти классов (пиксель относится к отрезку, описывающему одну из трёх основных

точек схода, при этом каждая точка схода определяет свой класс; пиксель является краевым, но не относится ни к одному из отрезков, описывающих три основные точки схода; пиксель относится к шуму в данных).

Угол поворота камеры находится как оценка апостериорной вероятности  $P(\psi|E_u)$ . Здесь стоит заметить, что, хотя все расчёты производятся в непрерывном пространстве, в результате всё равно используется дискретизация, и для нахождения ответа рассматриваются все дискретные значения  $\psi$  в диапазоне от  $-45^\circ$  до  $45^\circ$ . В качестве ответа берётся угол, для которого апостериорная вероятность максимальна.

Как утверждают авторы, алгоритм работает достаточно хорошо, за исключением случаев, когда на изображении много прямых, не соответствующих трём основным направлениям (например, присутствует машина). Также отмечается, что алгоритм работает слишком долго (по минуте на изображении  $640 \times 480$  пикселей).

В более поздней работе [13] представлен ряд улучшений предыдущего алгоритма. Данная работа показывает, что алгоритмы, основанные на картах краёв, дают более точные результаты и более эффективны. В этой работе также подразумевается откалиброванная камера. Кроме того, в работе представлена база фотографий, содержащая изображения, удовлетворяющие предположению Манхэттэнского мира, для которых пользователем была произведена разметка основных групп отрезков.

В данной работе, в отличие от предыдущей, рассматриваются не все пиксели, а только краевые пиксели, полученные детектором краёв. Все пиксели делятся на 4 класса: либо пиксел описывает одно из трёх основных направлений, либо пиксел относится к шуму. Итоговый угол поворота также вычисляется через максимизацию апостериорной вероятности. Авторам удалось улучшить модель из предыдущего метода с помощью представленной базы, по которой производилась настройка статистической модели. Таким образом, эта модель более адекватно описывает реальные данные.

Кроме того, в данной статье представлен новый метод для максимизации апостериорной оценки поворота угла. Предлагается два метода, не подразумевающие дискретизации: метод градиентного спуска, а также использование EM-алгоритма.

Эксперименты показали, что использование карты краёв, улучшенной статистической модели, а также метода градиентного спуска позволяют значительно улучшить производительность метода.

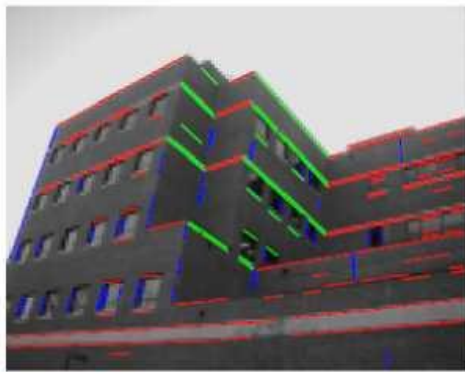
### *1.3.3 Методы, основанные на EM-алгоритме*

Довольно большую распространённость получили методы, основанные на EM-алгоритме. Одной из первых статей, предложившей данный подход, является работа [14, 15] Алгоритм находит не более трёх точек схода и подразумевает использование некалиброванной камеры.

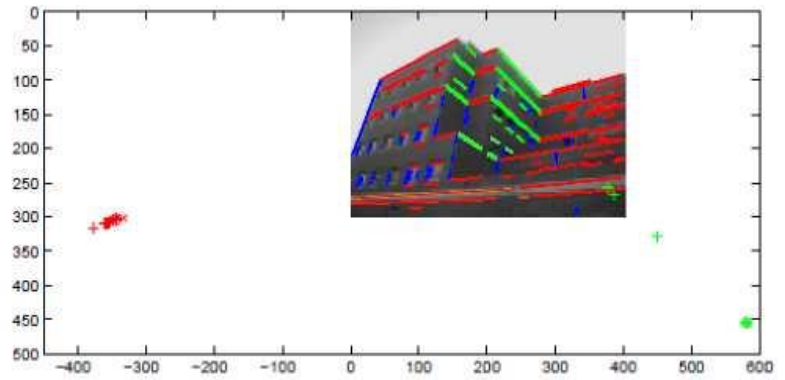
В качестве предварительного шага вычисляется карта краёв фильтром Канни, затем по карте краёв вычисляются отрезки. Далее для группировки отрезков и вычисления точек схода используется EM-алгоритм. В данной статье представлен также дополнительный шаг нормализации всех данных, а также предлагается новый способ инициализации EM-алгоритма.

На этапе инициализации EM алгоритма все отрезки группируются по углу наклона к оси OX, и для каждой такой группы вычисляется точка схода. При таком подходе на шаге инициализации получается большое количество неточных точек схода, а также лишних точек, однако уже такая инициализация позволяет алгоритму сходиться к довольно хорошему решению.

Сам EM-алгоритм работает следующим образом: на E шаге для каждого отрезка вычисляется вероятность принадлежности к каждой точке схода, на M шаге с учётом новых вероятностей, посчитанных на E шаге, пересчитываются координаты точек схода, а также, если несколько точек лежат слишком близко, то такие точки объединяются. Иллюстрация этого процесса представлена на рисунке 8.



а)



б)

Рисунок 8 – Иллюстрация двух шагов EM-алгоритма:  
*a* – группировка отрезков; *б* – нахождение точек схода

Алгоритм сходится довольно быстро, примерно за 2-5 итерации, в зависимости от начального приближения. Стоит отметить, что тот же алгоритм можно использовать и для нахождения любого количества точек схода без значительных изменений.

Другой пример использования EM-алгоритма представлен в работе [16]. Подобно работе [14] в качестве входных данных используется карта градиентов  $G$ , при этом никак не вычисляются ни отрезки, ни прямые. Результатом работы алгоритма является набор точек схода, а также разметка пикселей  $M$ . Каждый пиксель может быть отнесён к одному из следующих классов:

- 1) пиксель не является краевым пикселем;
- 2) пиксель относится к одной из точек схода;
- 3) пиксель является краевым, но не относится ни к одной из точек схода.

В работе предлагаются две различные модели: первая модель описывает изображения Манхэттэнского мира, т.е. изображения, на которых представлена только одна пара взаимно ортогональных горизонтальных точек схода, вторая модель описывает изображения, на которых представлено несколько пар взаимно ортогональных горизонтальных точек схода. Параметры камеры считаются

неизвестными, более того, показано, как модель может быть расширена для определения фокусного расстояния одновременно с нахождением точек схода,

В предложенной работе оптимизация производится не по положению точек схода, а по вектору параметров  $\psi$ , который определяет положение точек схода и вид которого зависит от выбранной модели мира. EM-алгоритм при этом работает следующим образом: на E шаге при зафиксированном значении  $\psi$  вычисляется разметка пикселей M, на M шаге при известной разметке пикселей находится новая оценка параметров  $\psi$ . Тестирование показало, что метод способен работать на довольно сложных изображениях городских сцен, содержащих несколько точек схода. Алгоритм обычно сходится за 20-30 шагов и довольно чувствителен к начальному приближению, из-за чего не для всех изображений находится верное решение.

#### *1.3.4 Методы, основанные на RANSAC-алгоритме*

RANSAC-алгоритм успешно применялся для решения различных задач компьютерного зрения. Например, в работе [17] показано, как этот подход может быть использован для нахождения точек схода на изображении,

RANSAC-алгоритм используется в данной работе для кластеризации отрезков (или прямых) в параллельные семейства и нахождения соответствующих точек схода. На вход алгоритму подаются отрезки, для каждой пары отрезков находится точка пересечения – гипотеза на положение точки схода. Далее каждая такая гипотеза проверяется на соответствие остальным отрезкам. Каждая точка оценивается с помощью количества отрезков, расстояние от которых до точки достаточно мало. Точка, получившая большее количество голосов отрезков, запоминается, а все отнесённые к ней отрезки удаляются, после чего процесс повторяется заново.

Как видно, в этом алгоритме не требуется знание параметров камеры, а также не делается предположение Манхэттэнского мира. Однако в работе [17] после применения RANSAC-алгоритма из полученных точек выбираются с помощью

дополнительной процедуры точки, отвечающие трём взаимно перпендикулярным направлениям.

Пример ещё одной работы, в которой используется RANSAC-алгоритм – [18]. В этой работе представлен более сложный алгоритм для нахождения точек схода. На вход алгоритму также подаются отрезки. Сначала отрезки группируются с помощью адаптивного RANSAC-алгоритма (его отличие заключается в том, что он позволяет изменять параметры во время работы алгоритма), затем происходит уточнение найденных групп на гауссовой сфере опять же с помощью RANSAC-алгоритма, после чего для каждой уточнённой группы отрезков вычисляется точка схода с помощью метода наименьших квадратов. После этого полученные точки схода уточняются с помощью специальной процедуры, которая заключается в минимизации сумм площадей треугольников, образованных отрезками и соответствующей точкой схода. Метод используется для нахождения трёх взаимно ортогональных точек схода и подразумевает известное фокусное расстояние, необходимое для определения положения гауссовой сферы.

### *1.3.5 Неитеративный метод быстрой и точной оценки точек схода*

Одним из самых последних и более точных методов является подход, представленный в статье [19]. Данный алгоритм находит произвольное количество точек схода, не используя значения параметров камеры, а также может находить три точки, соответствующие трём взаимно перпендикулярным направлениям в пространстве при заданном значении внутренних параметров камеры. Алгоритм сначала находит все отрезки на изображении, используя карту краёв, затем разделяет их на группы отрезков, соответствующим одной точке схода, и для каждой группы определяет соответствующую точку схода.

Наибольший интерес представляют шаги кластеризации отрезков и нахождения точек схода. В статье определяются две специальные функции:  $D(v, E_j)$  и  $V(S, w)$ , первая функция задаёт метрику расстояния от точки схода  $v$  до отрезка  $E_j$ , как среднее расстояние от прямой, проходящей через середину отрезка

					ЮУрГУ-120401.2017.071. ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		30

и точку схода, до концов отрезка; вторая функция в качестве результата выдаёт точку схода для списка отрезков  $S$ ,  $w$  – веса отрезков, обозначающие значимость каждого отрезка. В статье рассматриваются несколько способов вычисления функции  $V$ , однако, как показали эксперименты, самый эффективный способ – это нахождение точки, минимизирующей суммарное расстояние от точки до всех отрезков (по метрике  $D$ ).

Алгоритм можно представить в виде последовательности следующих шагов:

1) случайно выбирается  $M$  пар рёбер, для которых с помощью функции  $V(S, 1)$  определяется точка схода;

2) происходит построение булевой матрицы соответствия  $P (N \times M)$ . Для каждого отрезка  $v_i$  вычисляется расстояние до каждой найденной точки  $p_j$ . Если это расстояние меньше заданного порога, то соответствующему элементу матрицы  $P(i, j)$  присваивается значение 1, иначе - значение 0;

3) каждому  $i$ -ому отрезку определяется в соответствие  $i$ -ая строка матрицы – характеристическая функция отрезка. Затем проводится кластеризация всех отрезков по значению соответствующим им характеристических функций; для группы отрезков характеристическая функция определяется как пересечение строк матрицы, соответствующих отрезкам;

4) процесс повторяется итеративно:

- первоначально каждый отрезок определяет отдельный кластер;
- на каждом шаге определяются кластеры с наименьшим расстоянием и затем происходит их объединение;
- процесс продолжается, пока есть кластеры, находящиеся на расстоянии друг от друга меньше 1;

5) кластеры, содержащие меньше двух отрезков, отбрасываются – в результате получается порядка 3-7 кластеров;

6) для полученных кластеров опционально проводится процесс уточнения с помощью EM-алгоритма.

Также в случае, когда заданы параметры камеры, после выполнения основного алгоритма, возможен выбор трёх точек схода, соответствующих трём взаимно перпендикулярным направлениям. Алгоритм был протестирован на базе изображений и показал лучшее качество работы, среди существующих методов.

## 1.4 Обзор методов распознавания объектов

### 1.4.1 Метод поиска максимума кросскорреляционной функции

Дискретной кросс-корреляцией функций  $f(t)$  и  $g(t)$ , определенных на множестве целых чисел  $Z$ , называется следующая операция:

$$\text{corr}(f, g)[n] = \sum_{-\infty}^{+\infty} f[m]g[n + m]. \quad (1.2)$$

Кросс-корреляция чаще всего применяется в обработке сигналов, при этом  $f$  считается образцом, а  $g$  – сигналом, содержащим образец [20]. Результат – это вектор чисел, показывающих, насколько сильно образец выражен в сигнале. В нашем случае мы можем представить наши изображения в виде двух дискретных функций яркости и применить к ним операцию кросскорреляции. Максимум этой функции будет находиться в той области где изображения совпадают больше всего. По сути операция кросскорреляции является сверткой двух функций.

Свёртка функций – операция в функциональном анализе, показывающая «схожесть» одной функции с отражённой и сдвинутой копией другой.

Как следствие, используется одно из свойств свертки функции, а именно свойство Фурье-образа:

$$F(f * g) = F(f) \cdot F(g), \quad (1.3)$$

где  $F(f)$  – преобразование Фурье функции  $f$ .

Преобразование Фурье функции  $f$  вещественной переменной является интегральным преобразованием и задается следующей формулой:

					ЮУрГУ-120401.2017.071. ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		32



$$f(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} f(x) e^{-ix\omega} dx. \quad (1.4)$$

Метод подразумевает замену операции вычисления свертки операцией умножения Фурье-образов функций яркости изображений, что позволяет значительно повысить производительность вычислений.

Применительно к обработке изображений используется дискретное преобразование Фурье.

Прямое преобразование:

$$x_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N} kn}, k = 0 \dots N - 1, \quad (1.5)$$

и обратное преобразование:

$$x_n = \frac{1}{N} \sum_{k=0}^{N-1} x_k e^{\frac{2\pi i}{N} kn}, n = 0 \dots N - 1. \quad (1.6)$$

Применив это преобразование к функциям яркости изображений, мы получим вектора комплексных чисел. Перемножая поэлементно, мы получим вектор комплексных чисел. Применив обратное преобразование Фурье получим значения функции свертки. Найдя максимум среди этих значений, мы находим точку, в которой наши два изображения наиболее похожи [20].

Можно отметить следующие недостатки данного метода:

- 1) т. к. метод ищет глобальный максимум, то он может найти только 1 совпадение;
- 2) не устойчив к поворотам;
- 3) относительно неустойчив к изменению общего уровня яркости изображений.

Также выделим преимущества:

- 1) простота реализации;
- 2) высокая производительность вычислений;
- 3) качественное детектирование объектов на изображении;
- 4) устойчивость к масштабированию.

#### *1.4.2 Обнаружение устойчивых признаков изображения: метод SURF*

Суть метода заключается в следующем: для изображения сцены и изображения эталона с помощью метода SURF находятся особые точки и уникальные дескрипторы для них. Сравнивая эти наборы дескрипторов, мы можем выделить эталонный объект на сцене [20].

Метод ищет особые точки с помощью матрицы Гессе. Детерминант матрицы Гессе (т.н. гессиан) достигает экстремума в точках максимального изменения градиента яркости. Он хорошо детектирует пятна, углы и края линий.

Гессиан инвариантен относительно вращения. Но не инвариантен масштабу. Поэтому SURF использует разномасштабные фильтры для нахождения гессианов.

Для каждой ключевой точки определяется направление максимального изменения яркости (градиент) и масштаб, взятый из масштабного коэффициента матрицы Гессе. Градиент в точке вычисляется с помощью фильтров Хаара.

После нахождения ключевых точек, SURF формирует их дескрипторы. Дескриптор представляет собой набор из 64 (либо 128) чисел для каждой ключевой точки. Эти числа отображают флуктуации градиента вокруг ключевой точки. Поскольку ключевая точка представляет собой максимум гессиана, то это гарантирует, что в окрестности точки должны быть участки с разными градиентами. Таким образом, обеспечивается дисперсия (различие) дескрипторов для разных ключевых точек.

Флуктуации градиента окрестностей ключевой точки определяются относительно направления градиента вокруг точки в целом (по всей окрестности ключевой точки). Таким образом, достигается инвариантность дескриптора относительно вращения. Размер же области, на которой считается дескриптор,

					ЮУрГУ-120401.2017.071. ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		34

определяется масштабом матрицы Гессе, что обеспечивает инвариантность относительно масштаба. Флуктуации градиента также считаются с помощью фильтра Хаара.

Таким образом, обзор современных технических отечественных решений в области оптических систем навигации показал, что задачи навигации с применением систем технического зрения являются актуальными на сегодняшний день. Детально рассмотрены технические преимущества и недостатки оптического датчика координат, разработанного Институтом космических исследований РАН.

В разделе приводится краткое, но достаточное для понимания описание методов распознавания объектов на двумерных изображениях

Также, видно, что для решения задачи анализа геометрии городских сцен и поиска геометрических примитивов было представлено довольно много различных алгоритмов. Однако главный недостаток всех этих методов заключается в том, что определение примитивов различного уровня производится последовательно, т.е. алгоритм чаще всего независимо решает последовательность следующих задач:

- 1) чаще всего алгоритм инициализируется картой краёв изображения;
- 2) затем краевые пиксели группируются в отрезки или прямые;
- 3) далее прямые / отрезки группируются в семейства с последующим определением точек схода для каждой группы;
- 4) и уже в самом конце по найденным точкам схода определяется горизонт и зенит.

Главный недостаток данного подхода в том, что на каждом шаге присутствует ошибка вычисления. Так, карта краёв обычно достаточно сильно зашумлена пикселями, не относящимися ни к каким отрезкам или прямым изображения, т.е. уже на первом шаге могут найтись лишние прямые, далее при группировке прямых в семейства из-за наличия лишних прямых могут неправильно определиться точки схода, и затем уже на последнем шаге из-за неправильно

определённых точек схода вычисленный горизонт может оказаться очень неточным. Таким образом, ошибка на каждом шаге ведёт к ухудшению оценки положения горизонта на последнем шаге.

Некоторые методы используют немного модифицированный подход: вместо последовательного выполнения шагов некоторые шаги повторяются итерационно (например, группировка прямых и определение точек схода) до сходимости метода. Однако, чаще всего, такие методы склонны сходиться к локальному решению.

					ЮУрГУ-120401.2017.071. ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		36

## 2 ТЕОРЕТИЧЕСКАЯ РАЗРАБОТКА СИСТЕМЫ

Рассмотрим типовую последовательность разработки t-систем (технических систем) на рисунке 9.

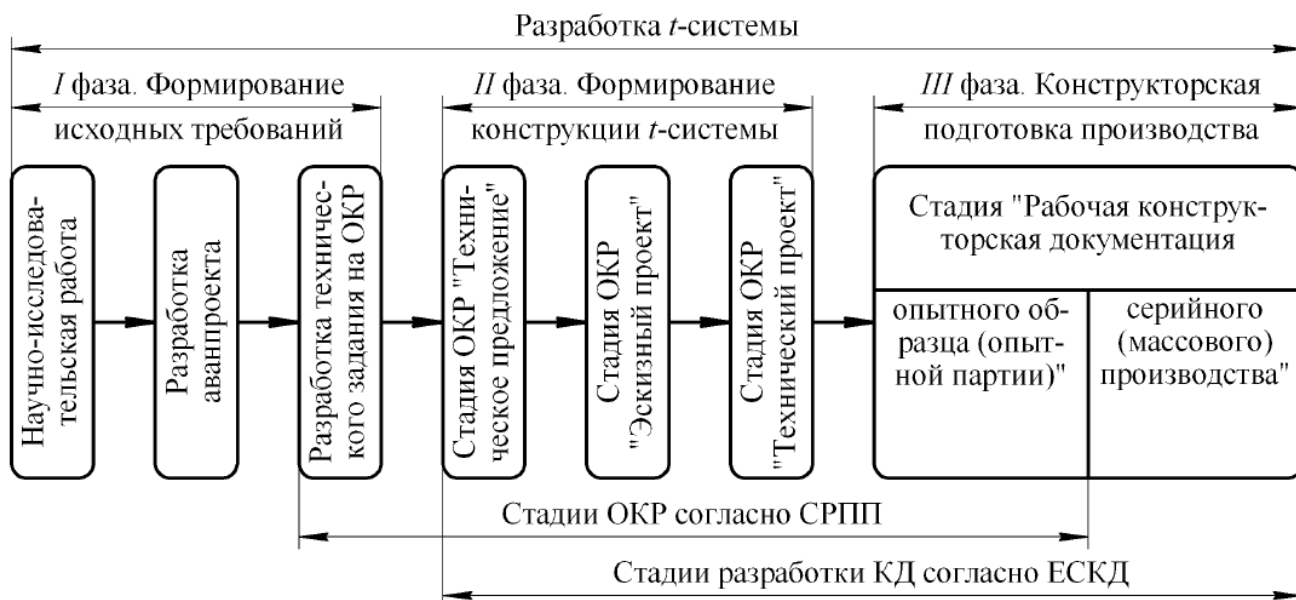


Рисунок 9 – Типовая последовательность разработки t-систем

Основным и, как правило, обязательным видом работ по разработке t-систем является опытно-конструкторская работа (ОКР).

ОКР – комплекс работ по разработке конструкторской и технологической документации на опытный (головной) образец создаваемой (модернизируемой, модифицируемой) t-системы, изготовлению и испытаниям опытного (головного) образца или опытной партии этой t-системы.

В рамках данной выпускной квалифицированной работы разрабатывается сложная t-система, для которой перед ОКР необходимо выполнить научно-исследовательскую работу (НИР).

НИР – комплекс теоретических и (или) экспериментальных исследований, проводимых с целью получения обоснованных исходных данных, изыскания принципов и путей создания (модернизации) t-систем.

В данной главе приведена последовательная разработка системы, которая включает в себя все стадии разработки технической системы.

## 2.1 Математическая модель преобразования оптических сигналов

Для корректного решения поставленной задачи необходимо составить математическую модель преобразования оптических сигналов от наблюдаемых объектов в разрабатываемой системе.

В общем случае входной оптический сигнал характеризуется приращением спектрального распределения яркости объекта  $L_{\lambda}^0(\lambda, x, y)$  к спектральной яркости  $L_{\lambda}^{\Phi}(\lambda)$  пространственно-равномерного фона, т.к. по сравнению с периодом считывания кадров в матричном приемнике излучения объект перемещается медленно, а именно [21]:

$$\Delta L(x, y, \lambda; x_0, y_0, \varphi) = L_{\lambda}^0(x, y, \lambda; x_0, y_0, \varphi) - L_{\lambda}^{\Phi}(\lambda), \quad (2.1)$$

где  $x_0, y_0, \varphi$  – параметры, описывающие ориентацию объекта, а также его положение и подлежащие измерению.

Объектив преобразует входной оптический сигнал в приращение распределения спектральной освещённости, которая описывается известным соотношением [22]:

$$\Delta E(x', y', \lambda, x'_0, y'_0, \varphi) = k_1 \tau_0(\lambda) \Delta L\left(\frac{x'}{\beta}, \frac{y'}{\beta}, \lambda; x'_0, y'_0, \varphi\right) \otimes H_{N\lambda}(x', y'), \quad (2.2)$$

где  $\beta$  – линейное увеличение объектива;

$$k_1 = \pi \sin^2 \sigma'_{A'};$$

$\sigma'_{A'}$  – задний апертурный угол объектива;

$\tau_0(\lambda)$  – спектральный коэффициент пропускания объектива;

$H_{N\lambda}(x', y')$  – монохроматическая функция рассеяния ОС.

С учётом дискретизации регистрируемого оптического сигнала и эффектов пространственной фильтрации, и если предположить, что выполняются условия, при которых МПИ линеен относительно входного сигнала, и пренебречь разбросом характеристик и параметров чувствительных элементов, то электрический сигнал на выходе МПИ определяется формулой [21]:

$$f(x_1, y_1) = U(x_1, y_1; x'_0, y'_0, \varphi) \cdot \text{comb} \left( \frac{x_1}{T_x}, \frac{y_1}{T_y} \right) \cdot \text{rect} \left( \frac{x_1}{l_x}, \frac{y_1}{l_y} \right) = f(mT_x, nT_y), \quad (2.3)$$

где  $U(x_1, y_1; x'_0, y'_0, \varphi) = S_{\lambda M} A_{\text{ПИ}} \cdot \Delta E_1(x_1, y_1, \lambda, x'_0, y'_0, \varphi) * H_{\text{ПИ}}(x_1, y_1); \quad (2.4)$

$$\Delta E_1(x_1, y_1, \lambda, x'_0, y'_0, \varphi) = \int_0^\infty \Delta E(x_1, y_1, \lambda, x'_0, y'_0, \varphi) d\lambda; \quad (2.5)$$

$$H_{\text{ПИ}}(x_1, y_1) = \frac{1}{A_{\text{ПИ}}} \text{rect} \left( \frac{x_1}{\alpha_x}, \frac{y_1}{\alpha_y} \right) - \text{импульсный пространственный отклик}$$

элементов МПИ;

$S_{\lambda M}$  – спектральная чувствительность пикселя МПИ на длине волны  $\lambda_M$ , соответствующей максимальной чувствительности;

$A_{\text{ПИ}} = \alpha_x \alpha_y$  – площадь пикселя МПИ;

$T_x$  и  $T_y$  – периоды расположения пикселей МПИ;

$l_x$  и  $l_y$  – размеры чувствительного слоя МПИ.

Если пренебречь искажениями сигнала, которые обусловлены квантованием, то массив значений сигналов, которые записываются в память ЭВМ для каждого из кадров, регистрируемых оптической камерой, представляется двумерным массивом:

$$f(i, j) = f(mT_x, nT_y) = \begin{bmatrix} f(0,0) & \dots & f(0, N-1) \\ \vdots & \ddots & \vdots \\ f(M-1,0) & \dots & f(M-1, N-1) \end{bmatrix}. \quad (2.6)$$

Правая часть выражения (2.6) описывает цифровое изображение. Элемент изображения – пикселя, имеющего размеры  $T_x \times T_y$  соответствует каждому элементу этой матрицы.

Полученная математическая модель найдена без учёта действия помех, которые возникают в матричном приемнике излучения при преобразовании изображения в электрический сигнал.

Для модельного описания помех, присутствующих в выходном изображении, примем следующие допущения [21]:

1) помехи являются аддитивными и подчиняются нормальному закону распределения вероятностей;

2) корреляция помех в близлежащих пикселях изображения отсутствует.

С учётом вышеперечисленных допущений для описания помех необходимо использовать модель белого шума. Корреляционная функция, описывающая случайное поле помехи в цифровом изображении, описывается выражением:

$$K(\Delta x, \Delta y) = \tilde{K}_0 \delta(\Delta x, \Delta y), \quad (2.7)$$

где  $\tilde{K}_0 = \sigma_N^2 T_x T_y$ ;

$\sigma_N^2$  – дисперсия помехи на выходе каждого из элементов МПИ.

Для оценки уровня помех в изображении необходимо использовать показатель в виде пикового отношения сигнала к шуму, который определяется как:

$$\mu = \frac{L_M}{\sigma_N},$$

где  $L_M$  – максимальная амплитуда уровня сигнала в изображении;

$\sigma_N$  – среднеквадратическое отклонение (СКО) шума.





среднеквадратического отклонения (СКО) шума. Расчет будем проводить как для световых величин (привязанных к спектральной чувствительности глаза), так и для энергетических величин.

В общем случае, излучение источника, подсвечивающего объект, имеет сложный спектральный состав, который у поверхности объекта характеризуется спектральной плотностью освещенности  $E_\lambda(\lambda)$ .

Соответственно выражение для интегральной освещенности в плоскости объекта будет иметь вид [24, 25]:

$$E_\lambda = \int_0^\infty E_\lambda(\lambda) \cdot d\lambda \left[ \frac{\text{Вт}}{\text{м}^2} \right]. \quad (2.8)$$

Ей будет соответствовать световая освещенность:

$$E_{\text{св}} = 683 \cdot \int_0^\infty E_\lambda(\lambda) \cdot V(\lambda) \cdot d\lambda, \quad (2.9)$$

где:  $E_\lambda(\lambda)$  – спектральная плотность энергетической освещенности объекта;

$E_\lambda$  – интегральная энергетическая освещенность объекта;

$E_{\text{св}}$  – световая освещенность объекта;

$V(\lambda)$  – относительная спектральная чувствительность глаза;

683 – пересчетный коэффициент энергетических величин в световые,  $\left[ \frac{\text{лм}}{\text{Вт}} \right]$ ;

$\lambda$  – длина волны излучения.

Для случая, когда угловые размеры объекта  $\omega_{\text{об}}$  значительно превышают угловой предел разрешения объектива оптической системы  $\delta_{\text{ос}}$ , и объект удален от объектива на расстояние  $L$ , значительно превышающее фокусное расстояние  $f'$  объектива:

$$\begin{cases} \omega_{\text{об}} > 10 \cdot \delta_{\text{ос}} \\ L > 10 \cdot f' \end{cases}, \quad (2.10)$$

					ЮУрГУ-120401.2017.071. ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		42

справедливо следующее выражение, связывающее спектральную освещенность в плоскости изображения со спектральной освещенностью объекта:

$$E'_\lambda(\lambda) = \frac{\tau_{\text{атм}}(\lambda) \cdot \tau_{\text{ос}}(\lambda) \cdot \rho_{\text{об}}(\lambda)}{4} \cdot \left(\frac{D}{f'}\right) \cdot E_\lambda(\lambda) = \frac{\tau_{\text{атм}}(\lambda) \cdot \tau_{\text{ос}}(\lambda) \cdot \rho_{\text{об}}(\lambda)}{4 \cdot k^2} \cdot E_\lambda(\lambda), \quad (2.11)$$

где:  $\tau_{\text{атм}}(\lambda)$  – спектральный коэффициент пропускания атмосферы;

$\tau_{\text{ос}}(\lambda)$  – спектральный коэффициент пропускания оптической системы (объектива);

$\rho_{\text{об}}(\lambda)$  – коэффициент диффузного отражения объекта;

$E_\lambda(\lambda)$  – спектральная плотность освещенности объекта;

$\frac{D}{f'}$  – относительное отверстие объектива;

$k$  – диафрагменное число объектива.

В этом случае энергетическая освещенность в плоскости светочувствительного слоя матричного приемника излучения, будет определяться выражением [24, 25]:

$$E'_\lambda = \int_0^\infty E'_\lambda(\lambda) \cdot d\lambda = \frac{1}{4 \cdot k^2} \int_0^\infty E_\lambda(\lambda) \cdot \tau_{\text{атм}}(\lambda) \cdot \tau_{\text{ос}}(\lambda) \cdot \rho_{\text{об}}(\lambda) \cdot d\lambda. \quad (2.12)$$

Будем считать, что объект подсвечивается источником излучения, относительная спектральная плотность светимости которого, совпадает с относительной спектральной плотностью светимости абсолютно черного тела (АЧТ) с температурой  $T$ .

$$E_\lambda(\lambda) = E_{\lambda_{\text{макс}}} \cdot \frac{M_\lambda(\lambda, T)}{M_{\lambda_{\text{макс}}}(T)} = E_{\lambda_{\text{макс}}} \cdot X(\lambda, T), \quad (2.13)$$

$$M_\lambda(\lambda, T) = \frac{2 \cdot \pi \cdot c^2 \cdot h}{\lambda^5} \cdot \frac{1}{\exp\left(\frac{h \cdot c}{\lambda \cdot k \cdot T}\right) - 1}, \quad (2.14)$$

$$\lambda_{\text{макс}} = \frac{2.8978 \cdot 10^{-6}}{T}, \quad (2.15)$$

где:  $E_{\lambda_{\max}}$  – максимальное значение спектральной плотности освещенности;  
 $M_{\lambda}(\lambda, T)$  – спектральная плотность светимости АЧТ, рассчитываемая по формуле Планка;

$M_{\lambda_{\max}}(T)$  – максимальное значение спектральной плотности светимости АЧТ;

$X(\lambda, T)$  – относительная спектральная плотность светимости АЧТ;

$\lambda$  – длина волны излучения;

$\lambda_{\max}$  – длина волны излучения, соответствующая максимуму спектральной плотности светимости АЧТ, рассчитываемая в соответствии с законом смещения Вина;

$T$  – температура АЧТ;

$h$  – постоянная планка,  $h = 6.626176 \cdot 10^{-34} \frac{\text{Дж}}{\text{Гц}}$ ;

$c$  – скорость света в вакууме,  $c = 2.99792458 \cdot 10^8 \frac{\text{м}}{\text{с}}$ ;

$k$  – постоянная Больцмана,  $k = 1.380662 \cdot 10^{-23} \frac{\text{Дж}}{\text{К}}$ .

Интегральная светимость АЧТ в диапазоне длин волн  $0 < \lambda < \infty$ :

$$M^0 = \int_0^{\infty} M_{\lambda}(\lambda, T) \cdot d\lambda = M_{\lambda_{\max}}(T) \cdot \int_0^{\infty} X(\lambda, T) \cdot d\lambda = \sigma \cdot T^4, \quad (2.16)$$

где:  $M^0$  – интегральная светимость АЧТ;

$\sigma$  – постоянная Стефана-Больцмана,  $\sigma = 5.6697 \cdot 10^{-8} \frac{\text{Вт}}{\text{м}^2 \cdot \text{К}^4}$ ;

Подставляем (2.15) в (2.14) и получаем выражение для расчета максимального значения спектральной плотности светимости АЧТ:

$$M_{\lambda_{\max}}(T) = 1.2865 \cdot 10^{-11} \cdot T^5, \left[ \frac{\text{Вт}}{\text{м}^2 \cdot \text{мкм}} \right], \quad (2.17)$$

где:  $M_{\lambda_{\max}}(T)$  – максимальное значение спектральной плотности светимости АЧТ;

$T$  – температура АЧТ, °К.

С учетом подстановки (2.15) и (2.14) в (2.13) получаем формулу для вычисления относительной спектральной плотности светимости АЧТ  $X(\lambda, T)$ :

$$X(\lambda, T) = \frac{2.9081 \cdot 10^{19}}{(\lambda \cdot T)^5} \cdot \frac{1}{\exp\left(\frac{14388}{\lambda \cdot T}\right) - 1}, \quad (2.18)$$

где:  $\lambda$  – длина волны излучения, мкм;

$T$  – температура АЧТ, °К.

На рисунке 11 приведены графики относительной спектральной плотности светимости АЧТ при температурах  $T = 2856$  °К и  $T = 6000$  °К.

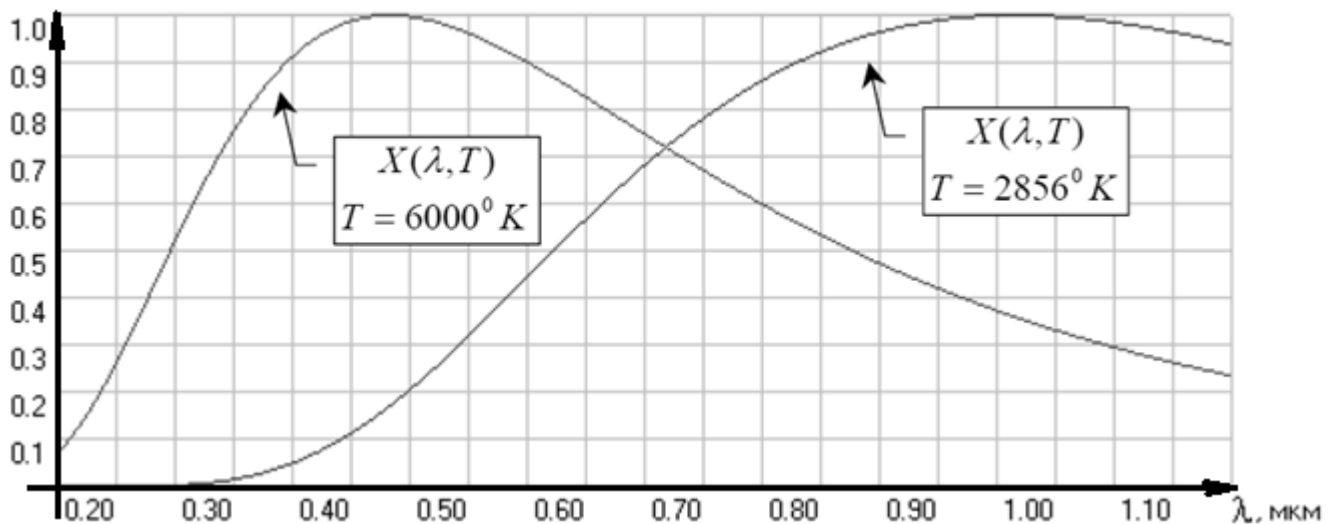


Рисунок 11 – Относительная спектральная плотность светимости АЧТ при температурах  $T = 2856$  °К и  $T = 6000$  °К

Из выражения (2.16) с учетом (2.17) можно получить численное значение для интеграла  $\int_0^\infty X(\lambda, T) \cdot d\lambda$ :

$$\int_0^\infty X(\lambda, T) \cdot d\lambda = \frac{\sigma \cdot T^4}{M_{\lambda_{\max}}(T)} = \frac{4407}{T}, \quad (2.19)$$

где:  $T$  – температура АЧТ, °К.

Окончательно для спектральной плотности освещенности в плоскости изображения получаем:

$$E'_\lambda(\lambda) = \frac{\tau_{\text{атм}}(\lambda) \cdot \tau_{\text{ос}}(\lambda) \cdot \rho_{\text{об}}(\lambda)}{4 \cdot k^2} \cdot E_{\lambda_{\text{макс}}} \cdot X(\lambda, T). \quad (2.20)$$

Соответственно выражение (2.12) для интегральной энергетической освещенности в плоскости изображения можно переписать в виде:

$$E'_3 = \int_0^\infty E'_\lambda(\lambda) \cdot d\lambda = \frac{E_{\lambda_{\text{макс}}}}{4 \cdot k^2} \int_0^\infty X(\lambda, T) \cdot \tau_{\text{атм}}(\lambda) \cdot \tau_{\text{ос}}(\lambda) \cdot \rho_{\text{об}}(\lambda) \cdot d\lambda. \quad (2.21)$$

Выражение для интегрального энергетического потока излучения, попадающего на одиночный пиксель матричного приемника излучения, будет иметь вид:

$$\Phi'_3 = A_{\text{пи}} \cdot E'_3 = \frac{A_{\text{пи}} \cdot E_{\lambda_{\text{макс}}}}{4 \cdot k^2} \cdot \int_0^\infty X(\lambda, T) \cdot \tau_{\text{атм}}(\lambda) \cdot \tau_{\text{ос}}(\lambda) \cdot \rho_{\text{об}}(\lambda) \cdot d\lambda, \quad (2.22)$$

где:  $\Phi'_3$  – интегральный энергетический поток излучения, попадающий на одиночный пиксель матричного приемника излучения;

$A_{\text{пи}}$  – площадь пикселя.

Выражения для световой освещенности и светового потока излучения будут иметь вид (2.23) и (2.24) соответственно:

$$E'_{\text{св}} = 683 \cdot \int_0^\infty E'_\lambda(\lambda) \cdot V(\lambda) \cdot d\lambda = \frac{683 \cdot E_{\lambda_{\text{макс}}}}{4 \cdot k^2} \times \\ \times \int_0^\infty X(\lambda, T) \cdot V(\lambda) \cdot \tau_{\text{атм}}(\lambda) \cdot \tau_{\text{ос}}(\lambda) \cdot \rho_{\text{об}}(\lambda) \cdot d\lambda \quad (2.23)$$

$$\Phi'_{\text{св}} = A_{\text{пи}} \cdot E'_{\text{св}} = \frac{683 \cdot A_{\text{пи}} \cdot E_{\lambda_{\text{макс}}}}{4 \cdot k^2} \times \\ \times \int_0^\infty X(\lambda, T) \cdot V(\lambda) \cdot \tau_{\text{атм}}(\lambda) \cdot \tau_{\text{ос}}(\lambda) \cdot \rho_{\text{об}}(\lambda) \cdot d\lambda. \quad (2.24)$$

Поток излучения (2.22) вызовет появление в потенциальной яме пикселя матрицы  $n$  сигнальных электронов:

$$n = t_{\text{нак}} \cdot A_{\text{пи}} \cdot \int_0^{\infty} \frac{E_{\lambda}'(\lambda) \cdot \eta(\lambda)}{E_{\phi}(\lambda)} \cdot d\lambda = t_{\text{нак}} \cdot A_{\text{пи}} \cdot \int_0^{\infty} E_{\lambda}'(\lambda) \cdot \frac{\eta(\lambda) \cdot \lambda}{h \cdot c} \cdot d\lambda \quad (2.25)$$

$$E_{\phi}(\lambda) = \frac{h \cdot c}{\lambda}, \quad (2.26)$$

где:  $n$  – количество сигнальных электронов;

$\eta(\lambda)$  – квантовая эффективность матрицы;

$E_{\phi}(\lambda)$  – энергия фотона на длине волны  $\lambda$ ;

$t_{\text{нак}}$  – время накопления;

$h$  – постоянная Планка;

$c$  – скорость света в вакууме.

Величина  $\frac{\eta(\lambda) \cdot \lambda}{h \cdot c}$  фактически представляет собой спектральную чувствительность матрицы:

$$S(\lambda) = \frac{\eta(\lambda) \cdot \lambda}{h \cdot c} = \frac{\max[\eta(\lambda) \cdot \lambda]}{h \cdot c} \cdot \frac{\eta(\lambda) \cdot \lambda}{\max[\eta(\lambda) \cdot \lambda]} = S_{\text{макс}} \cdot S_{\text{отн}}(\lambda), \quad (2.27)$$

где:  $S_{\text{макс}}$  – значение максимальной чувствительности матрицы;

$S_{\text{отн}}(\lambda)$  – относительная спектральная чувствительность матрицы.

Чувствительности матрицы  $S(\lambda)$  в данном случае имеет размерность  $[\text{Дж}^{-1}]$ , что соответствует количеству электронов, возникающих в потенциальной яме при поглощении 1 Дж лучистой энергии.

С учетом выражений (2.26) и (2.27) выражение (2.25) запишется в виде:

$$n = \frac{t_{\text{нак}} \cdot A_{\text{пи}} \cdot S_{\text{макс}} \cdot E_{\lambda \text{макс}}}{4 \cdot k^2} \times \int_0^{\infty} X(\lambda, T) \cdot S_{\text{отн}}(\lambda) \cdot \tau_{\text{атм}}(\lambda) \cdot \tau_{\text{ос}}(\lambda) \cdot \rho_{\text{об}}(\lambda) \cdot d\lambda. \quad (2.26)$$

Суммарный уровень шума, возникающий при фотоэлектрическом преобразовании, состоит из фотонного шума потока излучения от объекта и собственных шумов сенсора ПЗС матрицы и ее электронного тракта [25].

$$N_{\text{ш}} = \sqrt{N_{\text{ф}}^2 + N_{\text{пасп}}^2}, \quad (2.27)$$

где:  $N_{\text{ш}}$  – суммарное значение СКО шума;

$N_{\text{ф}}$  – СКО фотонного шума;

$N_{\text{пасп}}$  – паспортное значение СКО шума ПЗС матрицы.

Фотонный шум является следствием дискретной природы света и подчиняется закону распределения (статистике) Пуассона [25]. Согласно этой статистике, фотонный шум равен квадратному корню из числа сигнальных фотонов. Таким образом, отношение сигнал/шум в потоке фотонов, падающем на пиксель, будет также равно корню квадратному из числа фотонов. Приемник излучения дополнительно ухудшает соотношение сигнал/шум за счет наличия собственных шумов.

Учитывая, что по условию расчета, количество сигнальных электронов должно быть равно значению СКО шумовых электронов, получаем:

$$n = N_{\text{ш}} = \sqrt{(\sqrt{n})^2 + N_{\text{пасп}}^2} = \frac{1 + \sqrt{1 + 4 \cdot N_{\text{пасп}}^2}}{2} \approx N_{\text{пасп}}. \quad (2.28)$$

Соответственно выражение для пороговой величины максимальной спектральной плотности освещенности будет иметь вид:

$$E_{\lambda_{\text{макс}}}^{\text{пор}} = \frac{4 \cdot k^2 \cdot N_{\text{ш}}}{t_{\text{нак}} \cdot A_{\text{пи}} \cdot S_{\text{макс}} \cdot \int_0^{\infty} X(\lambda, T) \cdot S_{\text{отн}}(\lambda) \cdot \tau_{\text{атм}}(\lambda) \cdot \tau_{\text{ос}}(\lambda) \cdot \rho_{\text{об}}(\lambda) \cdot d\lambda}. \quad (2.29)$$



Сделаем следующие допущения:

1) Будем считать, что дальность до объекта наблюдения невелика, в этом случае можно пренебречь потерями в атмосфере, то есть  $\tau_{\text{атм}}(\lambda) \approx 1$ .

2) Рассматриваемый матричный приемник излучения имеет область спектральной чувствительности в диапазоне  $\Delta\lambda = 0.35 \div 1.0$  мкм (рисунок 2.3). За его пределами относительная спектральная чувствительность приемника  $S_{\text{отн}}(\lambda) = 0$ . Область спектральной чувствительности человеческого глаза еще уже и располагается в диапазоне  $\Delta\lambda = 0.4 \div 0.8$  мкм. То есть на практике, область интегрирования будет ограничена областью спектральной чувствительности приемника (глаза). В этом диапазоне коэффициент пропускания оптической системы можно считать постоянным  $\tau_{\text{ос}}(\lambda) \approx \text{const}$ .

3) Будем также считать постоянным в диапазоне коэффициент диффузного отражения объекта  $\rho_{\text{об}}(\lambda) \approx \text{const}$ .

$$\begin{cases} \tau_{\text{атм}}(\lambda) \approx \tau_{\text{атм}} \approx 1 \\ \tau_{\text{ос}}(\lambda) \approx \tau_{\text{ос}} \approx \text{const} \\ \rho_{\text{об}}(\lambda) \approx \rho_{\text{об}} \approx \text{const} \end{cases} \quad (2.30)$$

С учетом допущений (2.30) выражение (2.29) запишется в виде:

$$E_{\lambda_{\text{макс}}}^{\text{пор}} = \frac{4 \cdot k^2 \cdot N_{\text{ш}}}{t_{\text{нак}} \cdot A_{\text{пи}} \cdot S_{\text{макс}} \cdot \tau_{\text{ос}}(\lambda) \cdot \rho_{\text{об}}(\lambda) \cdot \int_0^{\infty} X(\lambda, T) \cdot S_{\text{отн}}(\lambda) \cdot d\lambda} \quad (2.31)$$

Подставляя выражение (2.31) в выражения (2.6) и (2.7) с учетом (2.11) получаем выражения для интегральных значений энергетической и световой освещенности на объекте:

$$\begin{aligned} E_{\text{э}}^{\text{пор}} &= E_{\lambda_{\text{макс}}}^{\text{пор}} \cdot \int_0^{\infty} X(\lambda, T) \cdot d\lambda = \\ &= \frac{4 \cdot k^2 \cdot N_{\text{ш}}}{t_{\text{нак}} \cdot A_{\text{пи}} \cdot S_{\text{макс}} \cdot \tau_{\text{ос}} \cdot \rho_{\text{об}}} \cdot \frac{\int_0^{\infty} X(\lambda, T) \cdot d\lambda}{\int_0^{\infty} X(\lambda, T) \cdot S_{\text{отн}}(\lambda) \cdot d\lambda'} \end{aligned} \quad (2.32)$$

					ЮУрГУ-120401.2017.071. ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		49

$$E_{\text{св}}^{\text{пор}} = 683 \cdot E_{\lambda_{\text{макс}}}^{\text{пор}} \cdot \int_0^{\infty} X(\lambda, T) \cdot V(\lambda) \cdot d\lambda =$$

$$= \frac{683 \cdot 4 \cdot k^2 \cdot N_{\text{ш}}}{t_{\text{нак}} \cdot A_{\text{пи}} \cdot S_{\text{макс}} \cdot \tau_{\text{ос}} \cdot \rho_{\text{об}}} \cdot \frac{\int_0^{\infty} X(\lambda, T) \cdot V(\lambda) \cdot d\lambda}{\int_0^{\infty} X(\lambda, T) \cdot S_{\text{отн}}(\lambda) \cdot d\lambda}. \quad (2.33)$$

Выражение для порогового значения интегральной энергетической освещенности в плоскости изображения (2.19) с учетом (2.31) имеет вид:

$$E_{\text{э}}^{\prime \text{пор}} = \frac{N_{\text{ш}}}{t_{\text{нак}} \cdot A_{\text{пи}} \cdot S_{\text{макс}} \cdot \tau_{\text{ос}} \cdot \rho_{\text{об}}} \cdot \frac{\int_0^{\infty} X(\lambda, T) \cdot \tau_{\text{ос}} \cdot \rho_{\text{об}} \cdot d\lambda}{\int_0^{\infty} X(\lambda, T) \cdot S_{\text{отн}}(\lambda) \cdot d\lambda}. \quad (2.34)$$

К сожалению, для интеграла в числителе выражения (2.34) невозможно получить численное значение, так как функция  $X(\lambda, T)$  имеет отличное от нуля значение в диапазоне  $0 < \lambda < \infty$ , а соответствующие данные по функциям  $\tau_{\text{ос}}(\lambda)$  и  $\rho_{\text{об}}(\lambda)$  отсутствуют. Тем не менее, применим к выражению (2.34) допущение (2.30) и в этом случае, получим оценочное (завышенное) значение пороговой освещенности. На самом деле такая ситуация эквивалентна случаю, когда матричный приемник без объектива подсвечивается тем же источником и при тех же условиях, что и объект.

$$E_{\text{э}}^{\prime \text{пор}} = \frac{N_{\text{ш}}}{t_{\text{нак}} \cdot A_{\text{пи}} \cdot S_{\text{макс}}} \cdot \frac{\int_0^{\infty} X(\lambda, T) \cdot d\lambda}{\int_0^{\infty} X(\lambda, T) \cdot S_{\text{отн}}(\lambda) \cdot d\lambda}. \quad (2.35)$$

Пороговое значение интегрального энергетического потока излучения:

$$\Phi_{\text{э}}^{\prime \text{пор}} = E_{\text{э}}^{\prime \text{пор}} \cdot A_{\text{пи}} = \frac{N_{\text{ш}}}{t_{\text{нак}} \cdot S_{\text{макс}}} \cdot \frac{\int_0^{\infty} X(\lambda, T) \cdot d\lambda}{\int_0^{\infty} X(\lambda, T) \cdot S_{\text{отн}}(\lambda) \cdot d\lambda}. \quad (2.36)$$

Пороговая световая освещенность в плоскости изображения:

$$E'_{\text{св}}^{\text{пор}} = \frac{683 \cdot N_{\text{ш}}}{t_{\text{нак}} \cdot A_{\text{пи}} \cdot S_{\text{макс}}} \cdot \frac{\int_0^{\infty} X(\lambda, T) \cdot V(\lambda) \cdot d\lambda}{\int_0^{\infty} X(\lambda, T) \cdot S_{\text{отн}}(\lambda) \cdot d\lambda}, \quad (2.37)$$

$$\Phi'_{\text{св}}^{\text{пор}} = E'_{\text{св}}^{\text{пор}} \cdot A_{\text{пи}} = \frac{683 \cdot N_{\text{ш}}}{t_{\text{нак}} \cdot S_{\text{макс}}} \cdot \frac{\int_0^{\infty} X(\lambda, T) \cdot V(\lambda) \cdot d\lambda}{\int_0^{\infty} X(\lambda, T) \cdot S_{\text{отн}}(\lambda) \cdot d\lambda}. \quad (2.38)$$

Величина обратная отношению интегралов в выражениях (2.32), (2.35) и (2.36) представляет собой коэффициент использования излучения источника матричным приемником.

$$K_{\text{пи}} = \frac{\int_0^{\infty} X(\lambda, T) \cdot S_{\text{отн}}(\lambda) \cdot d\lambda}{\int_0^{\infty} X(\lambda, T) \cdot d\lambda}. \quad (2.39)$$

Для человеческого глаза выражение для коэффициента использования будет иметь вид:

$$K_{\text{гл}} = \frac{\int_0^{\infty} X(\lambda, T) \cdot V(\lambda) \cdot d\lambda}{\int_0^{\infty} X(\lambda, T) \cdot d\lambda}. \quad (2.40)$$

С учетом (2.39) и (2.40) выражения (2.37) и (2.38) можно записать в виде:

$$E'_{\text{св}}^{\text{пор}} = 683 \cdot \frac{N_{\text{ш}}}{t_{\text{нак}} \cdot A_{\text{пи}} \cdot S_{\text{макс}}} \cdot \frac{K_{\text{гл}}}{K_{\text{пи}}}, \quad (2.41)$$

$$\Phi'_{\text{св}}^{\text{пор}} = 683 \cdot \frac{N_{\text{ш}}}{t_{\text{нак}} \cdot S_{\text{макс}}} \cdot \frac{K_{\text{гл}}}{K_{\text{пи}}}. \quad (2.42)$$

Вычисление пороговых значений потока излучения и освещенности проведем согласно выражениям 2.32, 2.33, 2.35, 2.36, 2.37, 2.38.

Вычисление интегралов  $\int X(\lambda, T) \cdot S_{\text{отн}}(\lambda) \cdot d\lambda$  и  $\int X(\lambda, T) \cdot V(\lambda) \cdot d\lambda$  произведем методом прямоугольников в диапазоне  $\Delta\lambda = 0.35 \div 1.0$  мкм с шагом 0.01 мкм.

Значение  $X(\lambda, T)$  вычислим по формуле 2.18. Значения  $S_{\text{макс}}$  и  $S_{\text{отн}}(\lambda_i)$  вычисляются на основе табличных паспортных значений квантовой эффективности  $\eta(\lambda_i)$ .

Вычисление интеграла  $\int X(\lambda, T) \cdot d\lambda$  осуществим по формуле 2.19.

При расчете будем использовать следующие исходные данные:

- диафрагменное число объектива:  $k = 1.4$ ;
- коэффициент пропускания оптической системы:  $\tau_{\text{ос}} = 0.8$ ;
- цветовая температура источника излучения:  $T = 6000 \text{ К}$ ;
- коэффициент диффузного отражения объекта:  $\rho_{\text{об}} = 0.8$ ;

Результаты расчета представлены в таблице 1.

Таблица 1 – Результаты расчета

Параметр	Значение	Параметр	Значение
$S_{\text{макс}}, \text{Дж}^{-1}$	$1.23 \cdot 10^{18}$	$E_{\text{э}}^{\text{пор}}, \frac{\text{Вт}}{\text{м}^2}$	$5.47 \cdot 10^{-5}$
$A_{\text{ПИ}}, \text{м}^2$	$1.64 \cdot 10^{-10}$	$\Phi_{\text{э}}^{\text{пор}}, \text{Вт}$	$8.97 \cdot 10^{-15}$
$N_{\text{ш}}$	40	$E_{\text{св}}^{\text{пор}}, \text{лк}$	$9.02 \cdot 10^{-4}$
$t_{\text{нак}}, \text{с}$	0.033	$\Phi_{\text{св}}^{\text{пор}}, \text{лм}$	$1.48 \cdot 10^{-13}$
$\int_0^{\infty} X(\lambda, T) \cdot d\lambda$	1.54	$E_{\text{э}}^{\text{пор}}, \frac{\text{Вт}}{\text{м}^2}$	$7.15 \cdot 10^{-4}$
$\int_0^{\infty} X(\lambda, T) \cdot V(\lambda) \cdot d\lambda$	0.037	$E_{\text{св}}^{\text{пор}}, \text{лк}$	$1.17 \cdot 10^{-2}$
$K_{\text{ГЛ}}$	0.024		
$\int_0^{\infty} X(\lambda, T) \cdot S_{\text{отн}}(\lambda) \cdot d\lambda$	0.169		
$K_{\text{ПИ}}$	0.11		

Также, были сделаны следующие выводы: за счет увеличения времени накопления заряда, можно добиться значительного уменьшения пороговой освещенности; меньшее значение квантовой эффективности, размера пикселя и

коэффициента использования излучения источника следует меньшим значениям пороговой освещенности и потока излучения.

### 2.3 Разработка функциональной схемы системы

Схемы функциональные разъясняют определенные процессы, протекающие в отдельных функциональных цепях изделия или в изделии в целом.

Функциональные схемы являются основным техническим документом, определяющим функционально-блочную структуру отдельных узлов автоматического контроля, управления и регулирования технологического процесса и оснащение объекта управления приборами и средствами автоматизации (в том числе средствами телемеханики и вычислительной техники).

На рисунке 12 показана разработанная функциональная схема.

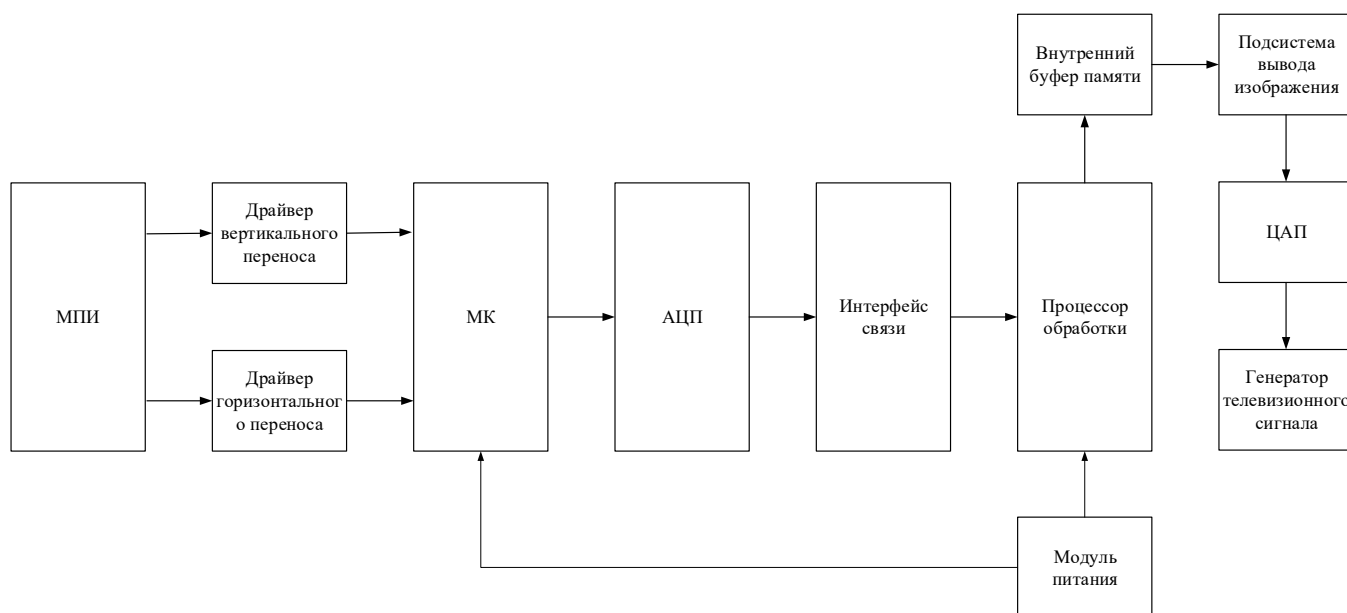


Рисунок 12 – Функциональная схема

В разрабатываемой системе решено применить матричный приемник излучения, изготовленный по ПЗС-технологии с межстрочным переносом заряда.

ПЗС-матрица – аналоговая интегральная микросхема, состоящая из светочувствительных фотодиодов, выполненная на основе кремния.

Для переноса заряда необходимы специализированные микросхемы, обозначенные на схеме как Драйвер вертикального переноса и Драйвер горизонтального переноса, которые в свою очередь управляются микроконтроллером. АЦП, предназначенный для выборки выходов трехлинейной цветной ПЗС-матрицы представляет собой отдельную микросхему. Блок Интерфейс связи предназначен для реализации обмена данными, передаваемыми на Процессор обработки. Для реализации системы технического зрения необходим процессор, обеспечивающий непрерывную обработку данных (на схеме Процессор обработки). Также данный модуль выполняет функцию подготовки данных для передачи на устройства отображения.

## **2.4 Разработка печатной платы матричного приемника излучения**

В качестве центрального элемента, разрабатываемого ФУ следует применить микросхему FTDI-FT2232HL. Применение микросхемы FT2232HL позволит реализовать высокоскоростной мост между USB и различными типами доступных интерфейсов, в которые можно передавать (и откуда можно принимать) данные через USB.

Микросхема FT2232HL (принадлежащая 5 поколению USB устройств FTDI) – преобразователь интерфейса USB 2.0 High Speed 480 мегабит/сек (поддерживается совместимость с USB 2.0 High Speed 480 мегабит/сек и Full Speed 12 мегабит/сек) в интерфейсы UART/FIFO с возможностью конфигурирования в различные стандартные последовательные или параллельные интерфейсы. Протокол USB полностью обрабатывается внутри микросхемы, не требуется дополнительное firmware. Далее приведены типичные применения данной микросхемы:

- мост USB – 2 канала UART (RS232, RS422 или RS485);
- мост USB – 2 канала FIFO;
- мост USB – 2 канала JTAG;
- мост USB – 2 канала SPI;

- мост USB – 2 канала I2C;
- мост USB – 2 канала Bit-Bang;
- мост USB – комбинация из вышеперечисленных интерфейсов;
- мост USB – быстрый оптический последовательный интерфейс (Fast Serial Optic Interface);
- мост USB – двойной, независимый интерфейс CPU (в качестве памяти);
- мост USB – эмуляция Host Bus (в качестве CPU);
- подключение PDA к USB;
- ридеры для смарт-карт, подключаемые по USB;
- подключение плееров mp3 к USB;
- использование шины USB для промышленного управления и инструментальное применение USB;
- ридеры для FLASH-карт, подключаемые по USB;
- подключение цифровых видеокамер к USB;
- ридеры для баркода (штриховой код), подключаемые по USB.

Отличительной особенностью микросхемы является режим Bit-Bang, при котором время переключения между выводами около 65 нс.

Микросхема рассчитана на напряжение питания 3.3 вольт, и в соответствии с технической документацией к ней, в разрабатываемой схеме конфигурируется с помощью подключения внешней микросхемы EEPROM 93C46.

Микросхема выпускается в 64-выводных корпусах LQFP или QFN, которые имеют одинаковую нумерацию контактов по своим выполняемым функциям (Рисунок 13).

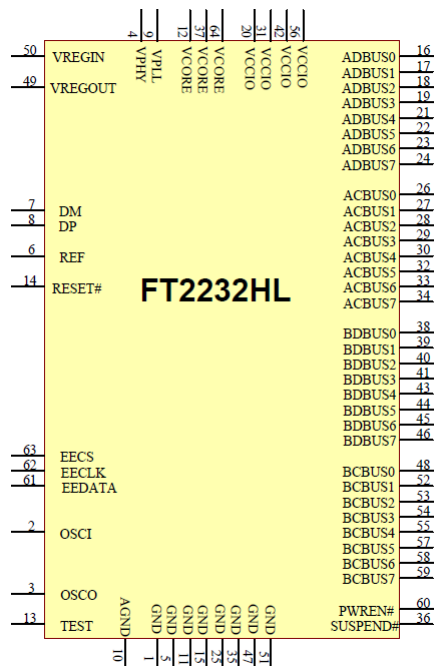


Рисунок 13 – Цоколевка микросхемы FT2232HL

16-разрядный АЦП (процессор обработки сигналов изображения) AD9826KRSZ, который имеет 3-канальную архитектуру, предназначенную для выборки выходов трехлинейных цветных ПЗС-матриц подходит для применения в разрабатываемой сенсорной плате устройства.

Рекомендуемая конфигурация цепи представлена на рисунке 14.

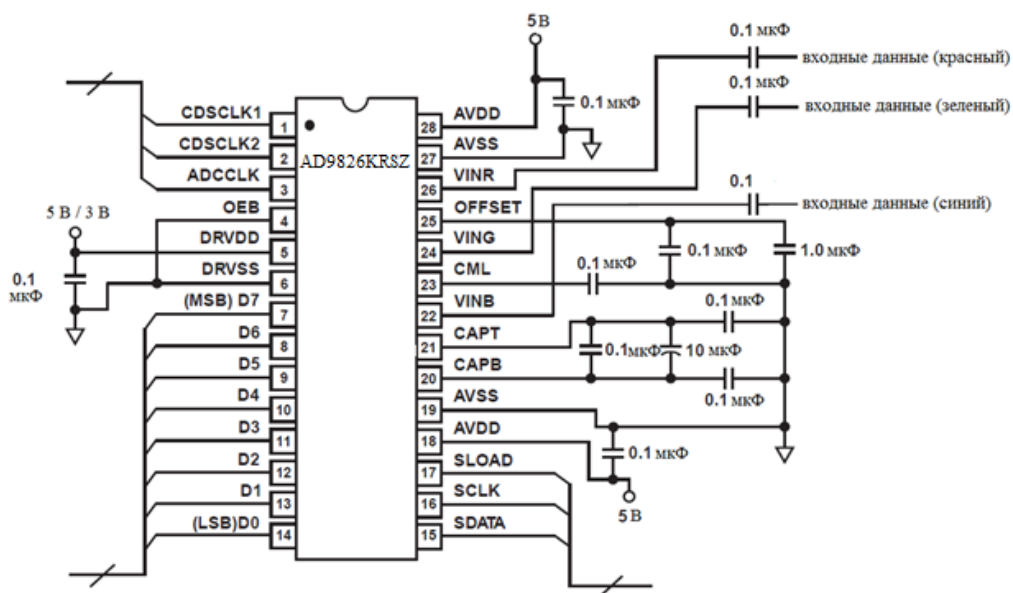


Рисунок 14 – Рекомендуемая конфигурация цепи AD9826KRSZ

Изм.	Лист	№ докум.	Подпись	Дата
------	------	----------	---------	------



В качестве чувствительного элемента в разрабатываемой системе, применим ПЗС-матрицу Sony ICX493AQ с межстрочным переносом заряда. Перечислим назначения ее выводов:

- 5, 7, 10, 14 ( $V_{\phi 1} - V_{\phi 4}$ ) – входы тактовых импульсов переноса вертикального регистра;
- 2, 9, 16, 21, 29, 32 (GND) – корпус;
- 1 ( $V_L$ ) – вход смещения защитного транзистора;
- 26 (CSUB) – вход тактового сигнала смещения подложки;
- 20 ( $\phi RG$ ) – вход стробирующих импульсов;
- 24, 25, 27, 28, 30, 31, 33, 34 ( $H_{\phi 2D}, H_{\phi 1D}, H_{\phi 2C}, H_{\phi 1C}, H_{\phi 2B}, H_{\phi 1B}, H_{\phi 2A}, H_{\phi 1A}$ ) – входы тактовых импульсов переноса горизонтального переноса.

Структура и расположение выводов микросхемы ICX493AQ приведены на рисунке 15.

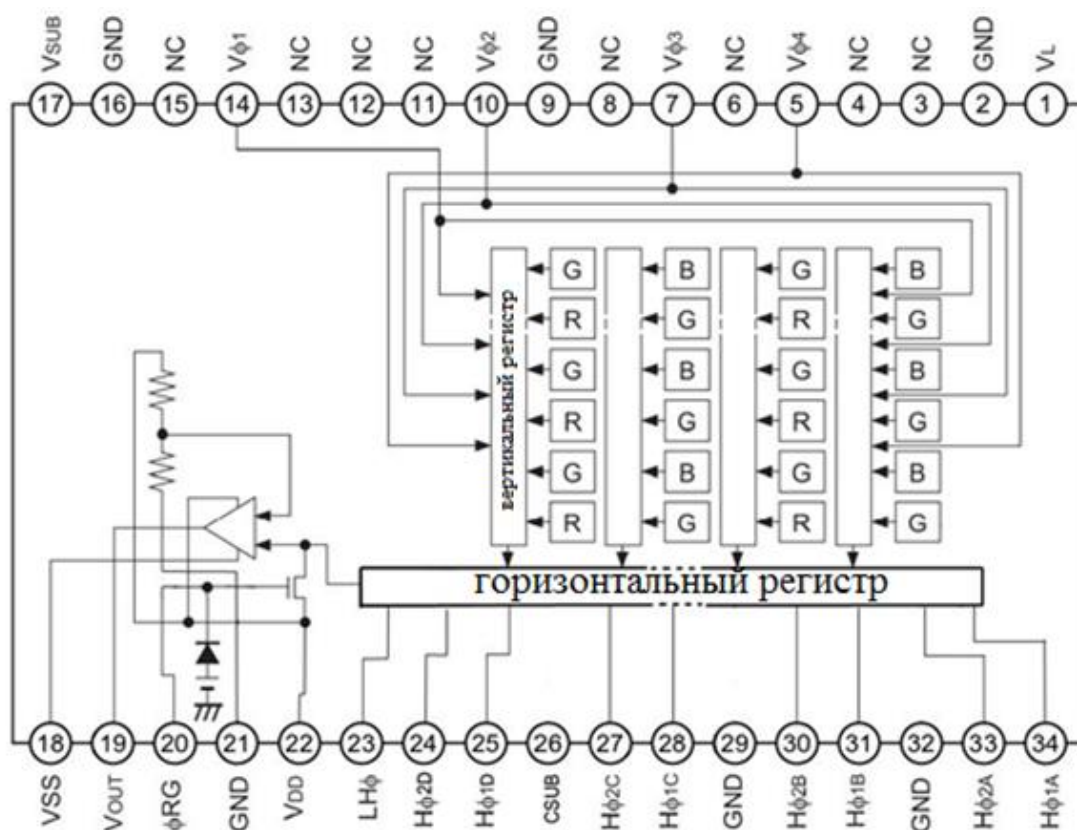


Рисунок 15 – Структура микросхемы ICX493AQ

Изм.	Лист	№ докум.	Подпись	Дата



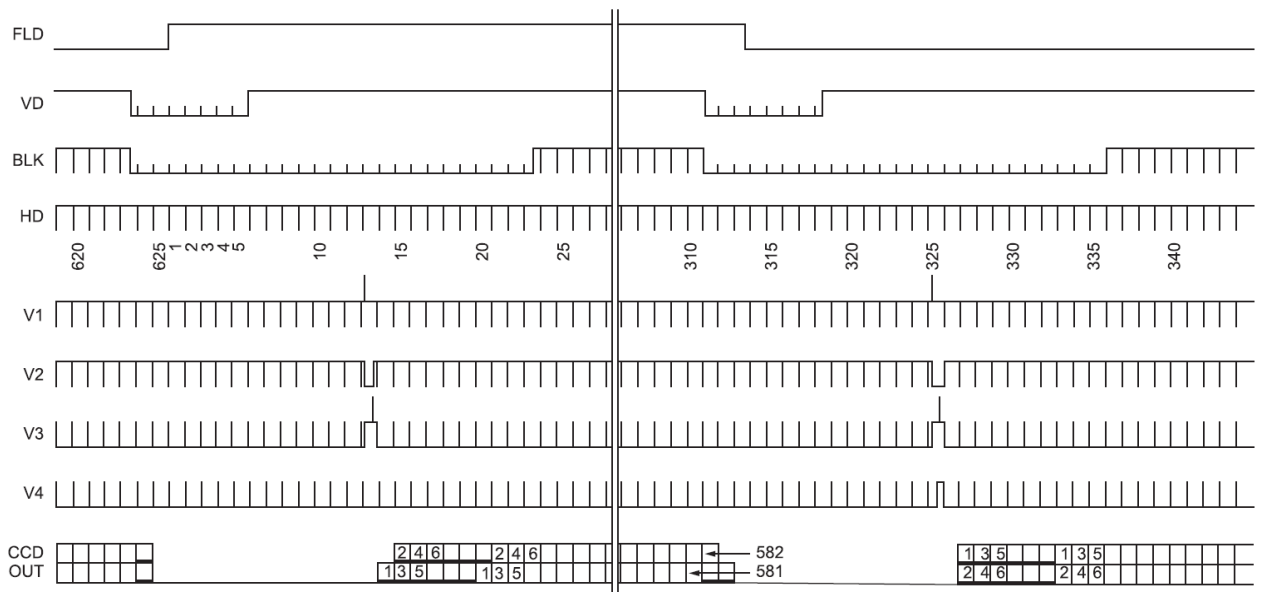


Рисунок 17 – Временные диаграммы тактовых сигналов ПЗС матрицы ICX459AQ в масштабе кадров

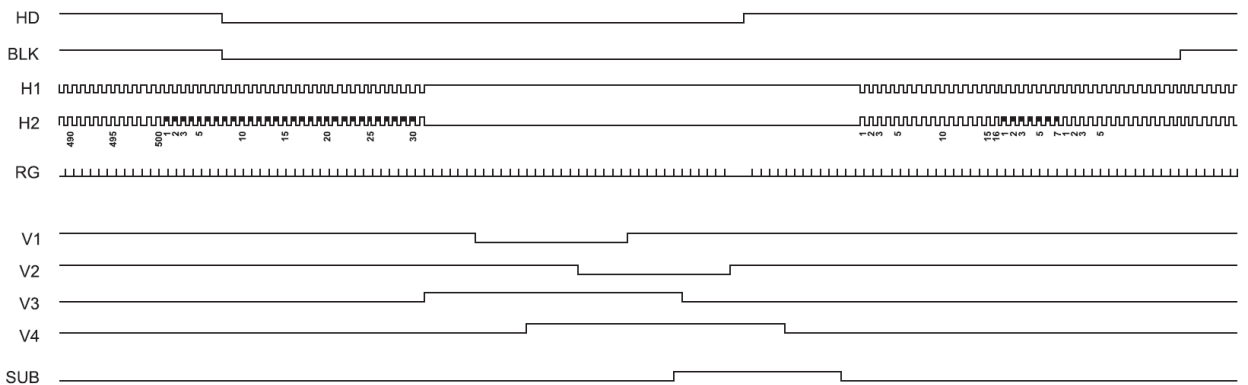


Рисунок 18 – Временные диаграммы тактовых сигналов ПЗС матрицы ICX459AQ в масштабе строк

На этих диаграммах (см. рисунки 17, 18) обозначены следующие сигналы:

1)  $H1 (H_{\phi 1}), H2 (H_{\phi 2})$  – тактовые сигналы горизонтального переноса (Horizontal transfer clock), типовая длительность импульсов на интервалах изображения (During imaging) 46 нс, на рис. 10 под диаграммой  $H2$  показаны номера ячеек в строке;

2)  $RG$  – сигналы сброса (Reset gate clock), типовая длительность импульсов 14 нс;

3)  $V_1-V_4$  ( $V_{\varphi 1}-V_{\varphi 4}$ )— тактовые сигналы вертикального переноса (Vertical transfer clock), сигналы формируются инвертированием сигналов XSUB.

Из комбинаций сигналов XV1, XSG1, XV3, XSG2 формируются синхронизирующие импульсы по строкам и кадрам HD, VD, гасящие и полевые импульсы BLK, FLD, налагаемые на выходной видеосигнал ПЗС-матрицы в соответствии с требованиями стандарта ТВ 625/50. В качестве задающего тактового генератора может быть использована микросхема CXD1156Q, подключаемая к выводам 4–10 микросхемы CXD1267AN.

Для управления работой регистров в схеме предусмотрен микроконтроллер ATmega 328.

Для питания микросхем необходимо использование импульсных стабилизаторов напряжения. Для этой цели было решено применить микросхему MAX743. Электрическая схема периферии питания приведена на рисунке 19.

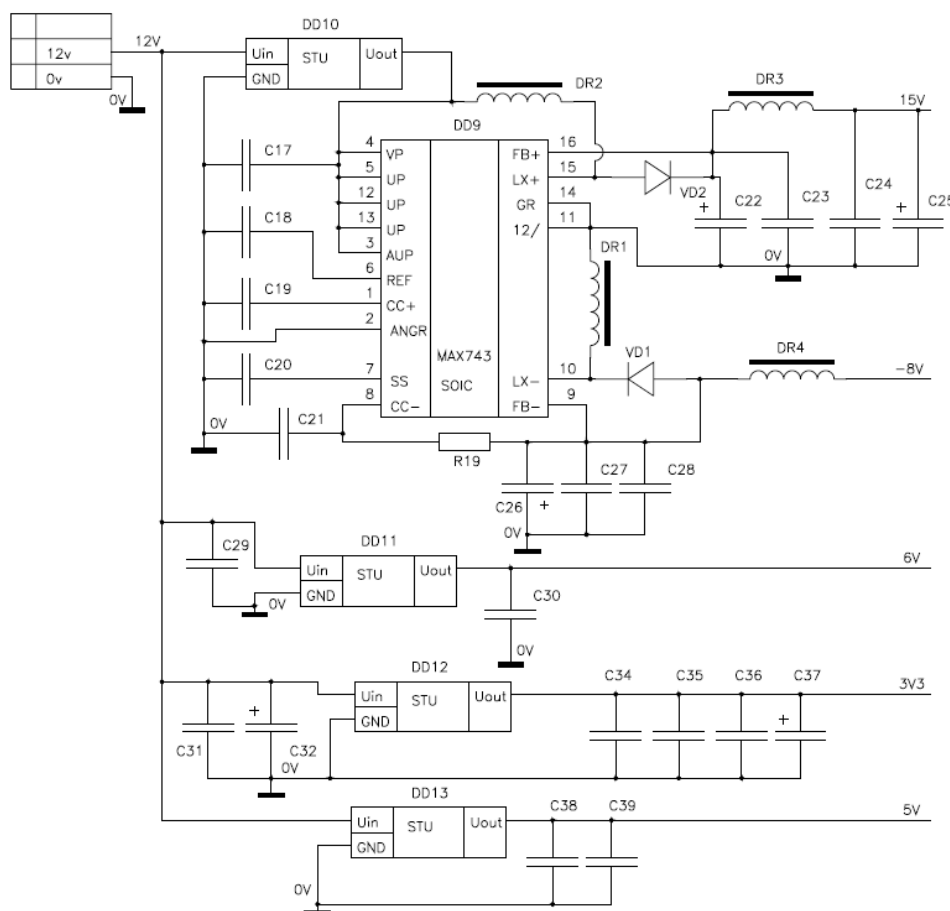


Рисунок 19 – Электрическая схема питания

Изм.	Лист	№ докум.	Подпись	Дата

На этой схеме (см. рисунок 19):

- DD9 – двухканальный импульсный стабилизатор напряжения MAX743;
- DD10, DD11 – стабилизатор напряжения 6В 78L06;
- DD12 – стабилизатор напряжения 3.3 В LP2950ACZ/CZ-3.3;
- DD12 – стабилизатор напряжения 5 В LP2950ACZ/CZ-5;

Проведем анализ условия эксплуатации разрабатываемой печатной платы. Дестабилизирующие факторы в ПП вызывают сложные физико-механические процессы, изменяющие физико-механические (расширение, размягчение, обезгаживание, деформацию, коробление, прогиб, скручивание ПП) и электрофизические свойства материала основания ПП (электропроводность, нагрузочная способность печатных проводников по току, диэлектрические свойства) и вызывающие отказы ЭА [28]. Для данного класса ЭА (наземная) и для класса жесткости 1 при климатических воздействиях допустимые значения воздействующих факторов приведены в таблице 2.

Таблица 2 – Обобщенные значения механических воздействующих факторов для класса наземной ЭА

Воздействующий фактор	Значение
Вибрация: -частота, Гц -ускорение, g	10...70 1...4
Многократные удары: -ускорение, g -длительность, мс	10...15 5...10
Одиночные удары: -ускорение, g -длительность, мс	50...1000 0,5...10
Линейное ускорение, g	2...5
Акустические шумы: -уровень, дБ -частота, Гц	85...125 50...1000

Важную роль при проектировании ПП играет не только учёт механических воздействий, но и климатических. Для группы жёсткости 1 при климатических воздействиях допустимые значения факторов приведены в таблице 3.

Таблица 3 – Допустимые значения климатических факторов

Фактор	Значение
Температура, К	Верхнее значение 328
	Нижнее значение 248
Относительная влажность, %	75 при температуре до 308 К
Перепад температур, К	от 248 до 328
Атмосферное давление	Нормальное

Влияние дестабилизирующих факторов на ПП во время использования и способы предотвращения их влияния на этапах конструирования и производства ПП представлено в таблице 4.

Таблица 4 – Влияние дестабилизирующих факторов на ПП

Воздействующий фактор	Ускоряемые деградационные процессы	Способы предотвращения влияния воздействующих факторов на этапе конструирования и производства ПП
Высокая температура	Расширение, размягчение, обезгаживание, деформация ПП: коробление, прогиб, скручивание	1. Применение нагревостойких материалов. 2. Выбор минимальных размеров ПП. 3. Выбор материалов ПП с близким ТКЛР в продольном и поперечном направлении и с медью.
	Уменьшение электропроводности, нагрузочной способности проводников по току, ухудшение диэлектрических свойств	1. Увеличение ширины и толщины проводников. 2. Применение материалов с низкими диэлектрическими потерями
	Перегрев концевых контактов ПП, увеличение их переходного сопротивления	Выбор гальванического покрытия со стабильными переходными сопротивлениями при нагреве
	Высыхание и растрескивание защитных покрытий	Выбор покрытия, устойчивого к высокой температуре
Низкая температура	Уменьшение электропроводности, нагрузочной способности по току, ухудшение диэлектрических свойств вследствие конденсации влаги, деформация, сжатие, хрупкость; электрохимическая коррозия проводников	1. Увеличение ширины и толщины проводников. 2. Выбор материалов ПП, устойчивых к низким температурам.

Окончание таблицы 4

Воздействующий фактор	Ускоряемые деградационные процессы	Способы предотвращения влияния воздействующих факторов на этапе конструирования и производства ПП
Вибрации	Механические напряжения, вызывающие деформацию или потерю механической прочности ПП; усталостные изменения ПП (разрушение); нарушение электрических контактов	1. Отстройка ПП от резонанса для выхода низшего значения собственной частоты из спектра частот внешних воздействий: а) путём выбора длины, ширины и толщины ПП; б) изменением суммарной массы установленных на ПП ЭРИ; в) выбором материала основания ПП; г) выбором способа закрепления сторон ПП в модулях более высокого конструктивного уровня. 2. Повышение механической прочности и жёсткости ПП: а) приклеиванием ЭРИ к установочным поверхностям ПП; б) покрытие лаком ПП вместе с ЭРИ; в) заливкой компаундами; г) увеличением площади опорных поверхностей; свойствами; д) использованием материалов с высокими демпфирующими е) демпфирующие покрытия; ж) рёбра жёсткости, амортизация и др.
Удары, линейное ускорение	Механические напряжения (разрушение ПП)	Повышение механической прочности и жесткости (см. «Вибрации»)

Также необходимо определить конструкторскую сложность функционального узла. Для этого подсчитаем число выводов микросхем, устанавливаемых на ПП. В таблице 5 приведены необходимые для подсчёта данные.

Таблица 5 – Устанавливаемые элементы

Наименование	Кол-во, шт	Количество выводов, шт.
FT2232HL	1	64
AT93C46-10S	1	8
AD9826KRSZ	1	28
AD811JRZ	1	18
ATmega328	1	32
74HC86	1	14
CXD1267AN	1	20
EL7457CSZ	1	16

Окончание таблицы 5

Наименование	Кол-во, шт	Количество выводов, шт.
MAX743	1	16
78L06	2	3
LP2950ACZ	2	3
ICX459AQ	1	34
Транзистор	3	3
Диод	4	2
Катушка индуктивности	7	2
Конденсатор	21	2
Резистор	25	2
Кварцевый фильтр	1	2
Сумма	75	387

Степень конструкторской сложности функционального узла влияет на тип конструкции ПП и метод изготовления. В зависимости от сложности применяют различные типы конструкции ПП: односторонние, двусторонние, многослойные ПП. Конструкторская сложность является средней, так как общее число выводов устанавливаемых элементов 387 (>300) [27]. Средняя конструкторская сложность ФУ требует применения МПП или ДПП.

Тип элементной базы для разрабатываемой ПП: поверхностно-монтируемые компоненты. Для обеспечения высокого быстродействия необходимо учитывать плотность монтажа, выбрать соответствующий материал основания ПП, увеличить число слоёв МПП, ввести внутренние межслойные переходы для уменьшения длины электрических связей.

В разрабатываемой ПП выводы компонентов штыревые, J-образные, а также применяются безвыводные компоненты. Для стационарной ЭА характерно отсутствие механических воздействий, а также высокой и низкой температуры во время работы, однако, она может подвергаться вибрациям, ударам, а также воздействию высоких и низких температур при транспортировании. К стационарной ЭА обычно не предъявляются высокие ограничения по габаритам и массе.

Основываясь на приведённых выше критериях, выберем третий (3) класс точности. Его основные характеристики приведены в таблицах 6 и 7.

					ЮУрГУ-120401.2017.071. ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		64



Таблица 6 – Номинальные значения параметров для классов точности ПП

Параметр	Класс точности ПП				
	1	2	3	4	5
H, мм					
t, мм	0,75	0,45	0,25	0,15	0,10
s, мм	0,75	0,45	0,25	0,15	0,10
d, мм	0,30	0,20	0,10	0,05	0,025
$\gamma = d / H$	0,40	0,40	0,33	0,25	0,20
$\Delta t$ , мм (без покрытия)	$\pm 0,15$	$\pm 0,10$	$\pm 0,05$	$\pm 0,03$	0; -0,03
$\Delta t$ , мм (с покрытием)	+0,25; -0,20	+0,15 -0,10	$\pm 0,10$	$\pm 0,05$	$\pm 0,03$
T, мм (ОПП и ДПП)	0,15	0,10	0,05	0,03	0,03
T, мм	0,20	0,12	0,10	0,05	0,03

В таблице 6 t – ширина проводника; s – расстояние между проводниками; b – минимально допустимая ширина контактной площадки; d – диаметр отверстия; H – толщина ПП;  $\Delta t$  – предельное отклонение ширины печатного проводника; T – позиционный допуск расположения печатных проводников.

Таблица 7 – Область применения и обоснование класса точности 3.

Область применения	Основные материалы	Вспомогательные материалы
Для ПП с ЭРИ и ПМК, имеющих штыревые и планарные выводы, а также с безвыводными компонентами, при средней и высокой степени насыщенности поверхности ПП ИЭТ.	На основе стеклоткани с гальваностойкой фольгой толщиной не более 35 мкм.	Сухой пленочный фоторезист (СПФ)

Исходя из класса точности (3) и типа конструкции ПП и элементной базы устанавливаемых компонентов, в качестве метода изготовления ПП выбираем метод металлизации сквозных отверстий, который является основным методом изготовления МПП. На рисунке 20 изображена конструкция печатного проводника при изготовлении МПП методом металлизации сквозных отверстий.

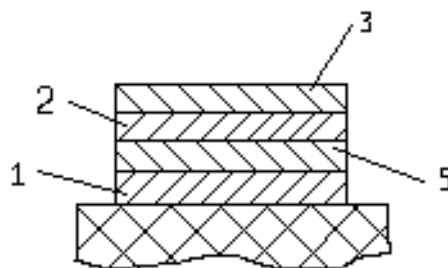


Рисунок 20 – Конструкция печатного проводника при изготовлении МПП методом металлизации сквозных отверстий: 1 – медная фольга (толщина  $h = 5$  мкм); 2 – гальваническая медь (толщина  $h = 25$  мкм); 3 – металлорезист (толщина  $h = 9$  мкм); 5 – химическая медь (толщина  $h = 2$  мкм)

При выборе материала основания ПП необходимо обратить внимание на предполагаемые механические воздействия (вибрации, удары, линейное ускорение и т.п.); класс точности ПП (расстояние между проводниками); реализуемые электрические функции; объект на который устанавливается ЭА; быстродействие; условия эксплуатации, стоимость. Для ЭА стационарной группы 3-го класса точности, 1-го класса жесткости наиболее подходящей является многослойная печатная плата с основанием из теплостойкого армированного стеклотекстолита (СТПА-5-2) [28].

Воспользуемся САПР AltiumDesigner для проектирования печатной платы. Система Altium Designer представляет собой интегрированный пакет программ, предназначенный для проектирования многослойных ПП РЭС [27]. Она адаптирована к операционной среде Windows и использует все настройки и возможности последней.

Altium Designer включает в себя работу с принципиальными схемами, созданием\редактированием библиотек, трассировкой ПП, эмуляцией работы принципиальной схемы, просмотр и редактирование 3D вида компонентов, добавление STEP моделей на ПП, отладка исходного кода по JTAG интерфейсу, аппаратное моделирование проектов на ПЛИС [27].

Трехмерная модель разработанной платы показана на рисунках 21, 22.

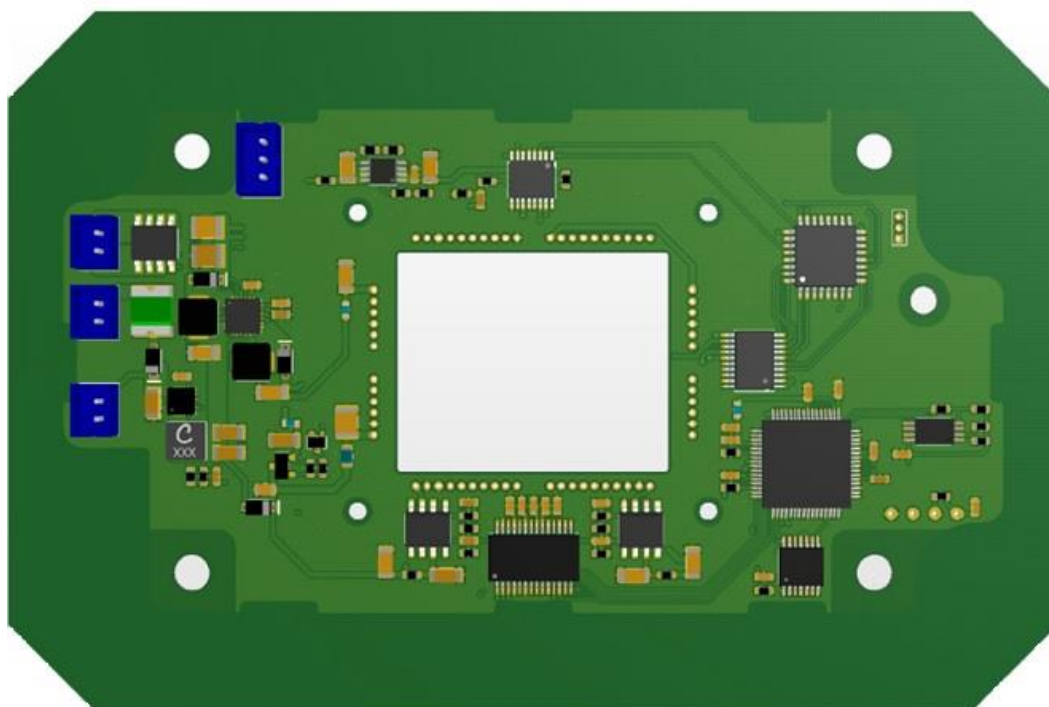


Рисунок 21 – Сенсорная плата

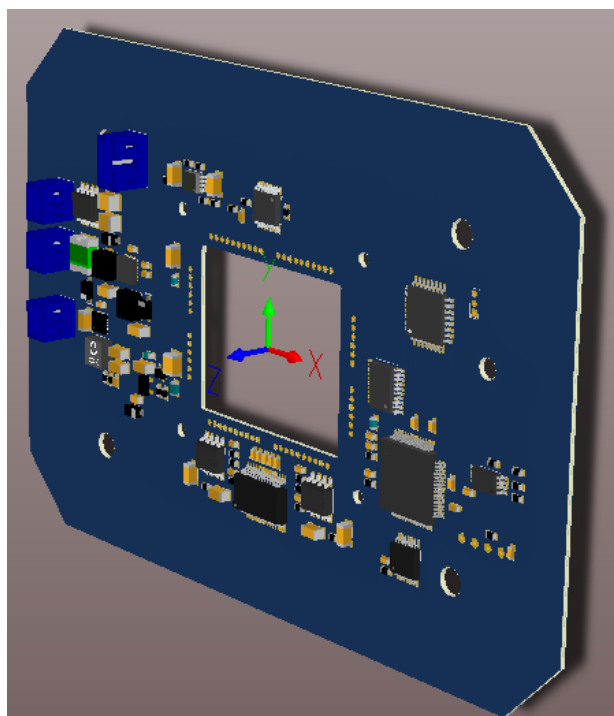


Рисунок 22 – Трехмерная модель платы в рабочем поле Altium Designer  
Электрическая принципиальная схема в приложении E.

					ЮУрГУ-120401.2017.071. ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		67

## 2.5 Плата цифровой обработки изображения

Современные системы технического зрения требуют обработки всё большего количества зрительных данных. В частности, эти потребности вызваны необходимостью получать данные с большей точностью по пространственному разрешению и использованием цветных изображений и данных с нескольких полей зрения [31]. С другой стороны, для СТЗ реального времени способность обрабатывать зрительные данные в рамках отпущенных временных интервалов является принципиальным требованием.

В последнее время вычислительные мощности графических процессоров (GPU) значительно выросли, в связи с чем возможность использования этих мощностей для вычислительных задач общего назначения становится все более заманчивой. Сегодня мы можем наблюдать большую активность в данном направлении [32]. Компьютерное зрение является одной из областей, в которых можно эффективно использовать GPU для широкого круга актуальных задач, включая детектирование лиц и пешеходов, восстановление трёхмерных поверхностей, распознавание объектов для робототехники и визуальной одометрии. Многие алгоритмы такого рода можно эффективно распараллеливать по данным и выполнять на GPU с высокой скоростью.

Область компьютерного зрения достаточно обширна, в ней действует большое число научных и промышленных коллективов, решающих самые разнообразные задачи на различных уровнях. Ранее уже предпринимались попытки использовать видеокарты для решения задач в области компьютерного зрения. Примером могут служить библиотеки CUVID Lib [2], GPU4VISION [3], GPUCV [4], openVidia [5], NPP [6].

NVIDIA Tegra – семейство систем на кристалле (SoC-System-on-a-Chip), разработанное компанией NVIDIA как платформа для производства мобильных интернет-устройств (Mobile Internet Device, MID), таких как смартфоны, смартбуки, коммуникаторы, КПК и др [32]. Кристалл Tegra объединяет в себе ARM-процессор, графический процессор, медиа- и DSP- процессоры,

					ЮУрГУ-120401.2017.071. ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		68

контроллеры памяти и периферийных устройств, имея при этом низкое энергопотребление.

Система на кристалле (однокристальная система) – электронная схема, выполняющая функции целого устройства (например, компьютера) и размещенная на одной интегральной схеме.

NVIDIA Tegra 2 – второе поколение процессора Nvidia Tegra, объединяющего функции целого компьютера в одном чипе, разработанный компанией NVIDIA как платформа для производства мобильных устройств, таких как смартфоны, смартбуки, КПК и т. д. [32].

Процессор включает специализированные ядра, два из которых имеют ARM-архитектуру Cortex-A9 и работают на частоте 1 ГГц, а также восьмиядерный графический процессор, медиа- и DSP- процессоры, контроллеры памяти и периферийных устройств. Компания называет Tegra 2 первым в мире двухъядерным процессором для применения в мобильных устройствах – речь идёт об интернет-планшетах, смартбуках и смартфонах.

Первые в мире мобильные двухъядерные и четырехъядерные CPU обеспечивают более быструю работу и мгновенный отклик. Встроенный ARM Cortex-A9 является первым мобильным процессором в мире с обработкой вне очереди для более эффективных вычислений, обеспечивающим лучшую работу.

GeForce GPU с ультранизким потреблением энергии созданы для эффективной работы с графикой при низком потреблении энергии.

NVIDIA Tegra 3 (кодовое название - NVIDIA Kal-EI) – третье поколение системы на кристалле семейства NVIDIA Tegra, разработанного американской компанией NVIDIA для коммуникаторов, планшетов и смартбуков [32]. Чип включает в себя четырёхъядерный процессор ARM Cortex-A9 MPCore с максимальной частотой 1.6 ГГц, пятое ядро-компаньон с максимальной частотой 500 МГц, двенадцатиядерный видеоускоритель NVIDIA GeForce ULP с поддержкой 3D-Стерео изображения, а также контроллер памяти. Согласно

					ЮУрГУ-120401.2017.071. ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		69

заявлениям NVIDIA, основанным на тестах GLBenchmark 2.0 Egypt, видеочип Tegra 3 в три раза быстрее чипа, интегрированного в Tegra 2.

Инновационная архитектура переменной симметричной многопроцессорной обработки NVIDIA позволяет использовать четыре производительных ядра для задач, требующих максимальной производительности, когда это необходимо, при этом каждое ядро является автономным и автоматически включается / выключается в зависимости от рабочей нагрузки. Единственное ядро с системой экономии ресурсов батареи питания (или ядро-компаньон) справляется с маломощными заданиями, такими как активный режим ожидания, прослушивание музыки и просмотр видео, и является прозрачным для ОС и приложений.

NVIDIA Tegra 4 (кодовое название — NVIDIA Wayne) – четвертое поколение системы на кристалле семейства NVIDIA Tegra, разработанного американской компанией NVIDIA для коммуникаторов, планшетов, смартфонов, игровых консолей и других устройств.

Чип включает в себя четырёхъядерный процессор ARM Cortex-A15 с максимальной частотой 1.9 ГГц, пятое ядро-компаньон с максимальной частотой 500 МГц, 72-ядерный видеоускоритель NVIDIA GeForce ULP с поддержкой 3D-Стерео изображения, а также двухканальный контроллер памяти.

Согласно заявлениям NVIDIA, видеочип Tegra 4 в шесть раз быстрее чипа, интегрированного в Tegra 3, и в двадцать раз производительнее Tegra 2.

Четырёхъядерный процессор ARM-процессор NVIDIA Tegra 4 использует самые мощные в истории процессорные ядра ARM, а также энергосберегающее ядро второго поколения, чтобы обеспечить рекордный уровень производительности и продолжительности работы от батареи. Процессор ARM Cortex-A 15 – это основной двигатель Tegra 4, в то время как Tegra 4i обеспечивает мощностью новый процессор ARM Cortex-A9 r4, который был назван ARM с помощью NVIDIA самым энергоэффективным процессорным ядром в своем классе.

					ЮУрГУ-120401.2017.071. ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		70

NVIDIA Tegra K1 (кодовое название — NVIDIA Logan) – пятое поколение системы на кристалле семейства NVIDIA Tegra, разработанного американской компанией NVIDIA для коммуникаторов, планшетов, смартфонов, игровых консолей и других устройств [32]. Данный чип впервые в истории семейства использует не специальную микроархитектуру графического ускорителя для мобильных устройств, а микроархитектуру Kepler, которая начала применяться в видеокартах класса GeForce 600 и Tesla. Кроме того, компания впервые в семействе Tegra анонсировала переход на архитектуру ARMv8 в специальном ядре NVIDIA Denver, хотя все предыдущие чипы Tegra использовали архитектуру ARMv7. Компания анонсировала, что Tegra K1 будет выпущена в двух вариантах, использующих разные микроархитектуры процессоров: четырёхъядерный процессор Cortex-A15 на частоте 2.3 ГГц с теневым ядром-компаньоном и двухъядерный ARMv8-совместимый процессор на частоте 2.5 ГГц. Обе редакции системы на чипе используют графический ускоритель, выполненный по микроархитектуре Kepler и обладающий 192 CUDA-ядрами.

Четырёхъядерный процессор ARM® Cortex™-A15 с архитектурой NVIDIA 4-Plus-1™ - Tegra K1 использует самый мощный в истории ARM CPU, Cortex A15, а также энергосберегающее ядро третьего поколения, чтобы обеспечить рекордный уровень производительности и продолжительности работы от батареи [32]. Архитектура переменной симметричной многопроцессорной обработки NVIDIA позволяет использовать четыре производительных ядра для задач, требующих максимальной производительности, когда это необходимо, при этом каждое ядро является автономным и автоматически включается / выключается в зависимости от рабочей нагрузки.

На рисунке 23 показана плата Nvidia Jetson TK1.



Рисунок 23 – Nvidia Jetson TK1

Платформа NVIDIA Jetson TK1 предоставляет разработчикам инструменты для создания систем и приложений нового поколения, использующих машинное зрение, обработку изображений и данных в реальном времени и применяемых в робототехнике.

Набор инструментов разработчика Jetson TK1 включает полноценный набор для C/C++ на базе архитектуры NVIDIA CUDA – распространенной платформы и модели программирования параллельных вычислений. Новая платформа с производительностью в 326 GFLOPS (почти втрое выше, чем у аналогичного класса встраиваемых платформ) упрощает разработку приложений по сравнению с процессорами FPGA, ASIC и DSP, широко используемыми в сегодняшних встраиваемых системах [32].

Его 192 полностью программируемых ядра обеспечивают качественную графику и высокую скорость вычислений в мобильном форм-факторе. Пакет Jetson TK1, разработанный с нуля под архитектуру CUDA, включает все необходимые разработчикам инструменты программирования, чтобы быстро создавать и разворачивать вычислительные системы.



Новая платформа поддерживает полный набор инструментов CUDA 6.0, включая отладчики и профилировщики для создания приложений параллельных вычислений. CUDA 6 также переносит на платформу ARM ускоренные библиотеки NVIDIA для алгоритмов быстрого преобразования Фурье, линейной алгебры, разреженных матриц, а также обработки изображений и видео.

Jetson TK1 также поддерживает набор инструментов NVIDIA VisionWorks, который включает широкий спектр алгоритмов машинного зрения и обработки изображения.

## 2.6 Плата питания

Для функционирования плат по пунктам 2.4, 2.5 настоящей работы в системе необходимо предусмотреть двухканальный модуль, обеспечивающий питание стабилизированным напряжением 12 В и 5 В, рассчитанный на работу в сети постоянного тока с напряжением 12, 27 или 48 В и габаритами не превышающий 40x80x20 (ШxВxГ). Для этой задачи целесообразно применить готовое решение, разработанное отечественным производителем электроники ООО «АЛЕКСАНДР ЭЛЕКТРИК источники электропитания».

В перечне каталога продукции данного производителя имеется модуль питания под условным обозначением МДМ7,5-2А0512М, который полностью удовлетворяет вышеперечисленным требованиям.

Данный модуль предназначен для питания стабилизированным напряжением 12 В и 5 В аналоговой и цифровой аппаратуры различного назначения и рассчитан на работу в сети постоянного тока с напряжением 12 В соответствующей ГОСТ В 24425-90, ГОСТ 19705-89, ГОСТ 54073-2010, ГОСТ 23396-78 [33].

Модуль выполнен на современной отечественной элементной базе. Поставляется в соответствии с техническими условиями БКЮС.430609.001 ТУ по КД литеры «О1» и БКЮС.430609.001-01 ТУ по КД литеры «А».

					ЮУрГУ-120401.2017.071. ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		73

Качество входной электроэнергии постоянного тока должно соответствовать требованиям групп А, Г ГОСТ В 24425-90.

В модуле используется принцип высокочастотного преобразования электрической энергии с одновременной стабилизацией выходного напряжения на основе широтно-импульсной модуляции. Подавление высокочастотных помех на входе и на выходе осуществляется встроенными фильтрами.

На приведенной структурной схеме (рисунок 24) показаны основные функциональные узлы, входящие в состав модуля.

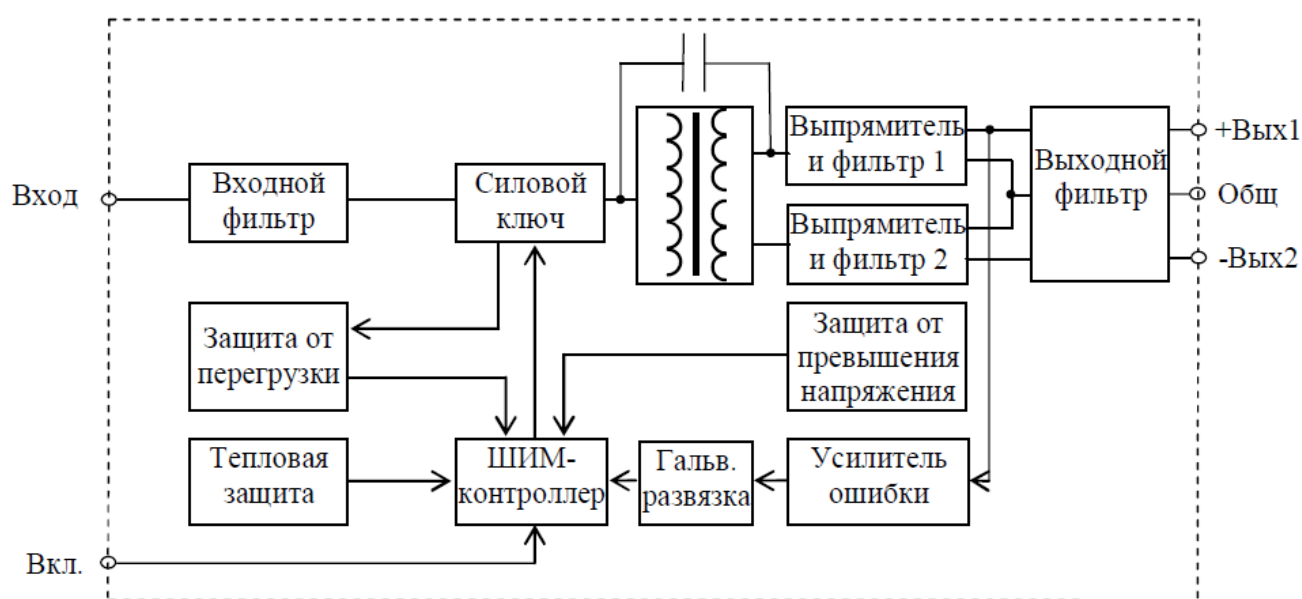


Рисунок 24 – Структурная схема модуля.

Модуль имеет гальваническую развязку между входом и выходом, между входом и корпусом, между выходом и корпусом. Два выходных канала имеют общую среднюю точку.

При разработке модуля использована однотактная обратногоходовая схема преобразователя (Рисунок 25). Гальваническая развязка цепи обратной связи по напряжению выполнена на базе трансформатора ТЗ (см. рисунок 25).

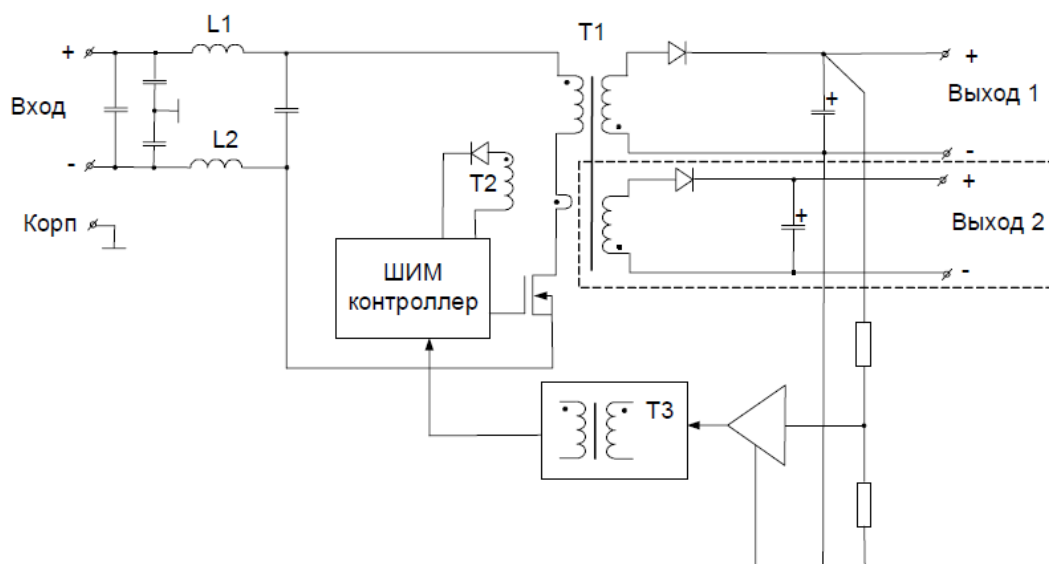


Рисунок 25 – Обратногоходовой преобразователь

Конструктивно модуль выполнен в металлических корпусах с крепежными фланцами или без них. Для установки на радиатор охлаждения или на печатную плату имеются резьбовые втулки и фланцы. Все модули имеют кремнийорганическую полимерную защиту – герметизацию компаундом с теплопроводным наполнителем.

В основе конструкции модуля лежит печатная плата (Рисунок 26) с элементами поверхностного монтажа, размещенная в алюминиевом корпусе и залитая эластичным компаундом.

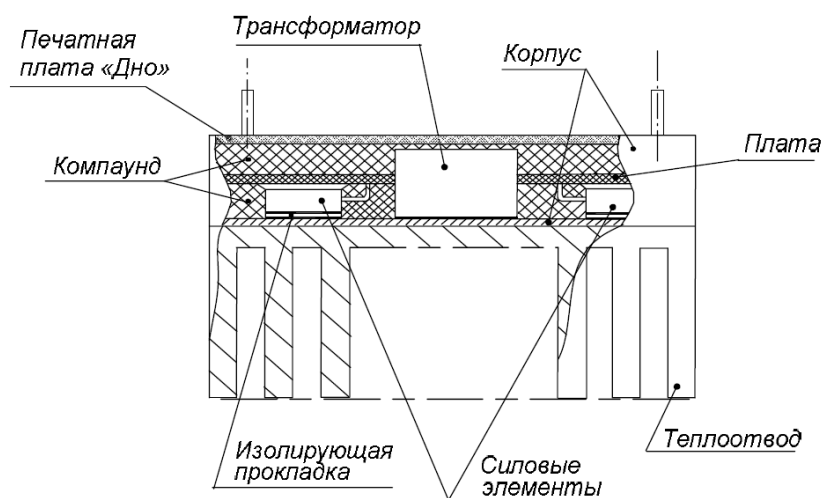


Рисунок 26 – Эскиз модуля в разрезе

Изм.	Лист	№ докум.	Подпись	Дата

Внешний вид, качество покрытия, габаритные, установочные и присоединительные размеры модуля должны соответствовать требованиям ТУ, комплекту конструкторской документации и инструкции по контролю внешнего вида, маркировки и качества покрытия БКЮС.435714.001.И54.

Тепловая энергия модуля выделяется в окружающее пространство излучением и конвекцией, причем основное тепло отдается за счет конвекции.

Теплоотдача путем конвекции с поверхности модуля, корпус которого нагрет до температуры  $\theta_k$  в окружающую среду с температурой  $\theta_{cp}$  определяется законом Ньютона [34].

$$P_{п} = \alpha_k \cdot S \cdot \Delta\theta, \quad (2.42)$$

где  $P_{п}$  – мощность потерь модуля;

$\alpha_k$  – коэффициент конвекции;

$S$  – площадь поверхности;

$\Delta\theta$  – разница между температурами корпуса и среды;

По формуле (2.42) можно определить температуру перегрева  $\Delta\theta$  модуля относительно окружающей среды:

$$\Delta\theta = \frac{P_{п}}{\alpha_k \cdot S}, \quad (2.43)$$

Большинство предприятий, выпускающих универсальные модули, заменяют выражение  $\frac{1}{\alpha_k \cdot S}$  на тепловое сопротивление  $R_{\theta}$ , которое приводят в документации на устройство, при этом формула (2) принимает вид:

$$\Delta\theta = P_{п} \cdot R_{\theta}. \quad (2.44)$$

Мощность потерь  $P_{п}$  определяется по формуле:

					ЮУрГУ-120401.2017.071. ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		76

$$P_{\Pi} = P_{\text{ВЫХ}} \cdot \left(\frac{1}{\eta} - 1\right), \quad (2.45)$$

где  $P_{\text{ВЫХ}}$  – мощность, потребляемая нагрузкой;

$\eta$  – КПД.

Поскольку в ТУ приведен только типовой КПД модулей, при расчётах следует уточнять КПД для каждого типа модуля по графикам, показывающим зависимость КПД от выходной мощности и входного напряжения. Эти графики также приводятся в документации на модули [33]. График для модулей МДМ7,5-1 с выходными напряжениями 5 и 12 В приведен на рисунке 27.

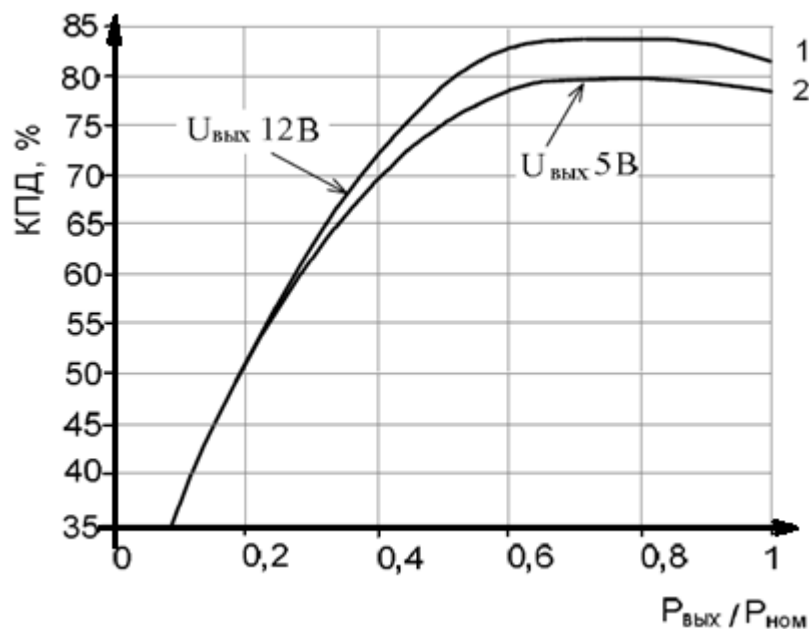


Рисунок 27 – График зависимости КПД модуля МДМ7,5-2А0512М при входном напряжении 27 В

По известным значениям  $R_{\theta}$  и  $P_{\Pi}$  по формуле (2.44) можно рассчитать температуру перегрева  $\Delta\theta$  и определить температуру окружающей среды  $\theta_{\text{ср}}$ , при которой модуль может работать без радиатора:

$$\theta_{\text{ср}} = \theta_{\text{к max}} - \Delta\theta, \quad (2.46)$$

где  $\theta_{к max}$  – максимально допустимая температура корпуса модуля.

Определим максимальную выходную мощность  $P_{вых max}$  модуля, при температуре окружающей среды  $70\text{ }^{\circ}\text{C}$ .

По формуле (2.46) с учетом выражений (2.44) и (2.45) получаем:

$$P_{вых max} = \frac{\theta_{к max} - \theta_{ср}}{R_{\theta} \cdot \left(\frac{1}{\eta} - 1\right)}. \quad (2.47)$$

Из документации на модуль имеем:  $\theta_{к max} = 85^{\circ}\text{C}$ ;  $R_{\theta} = 10.5^{\circ}\text{C}/\text{Вт}$ ;  $\eta = 0.81$ .

$$P_{вых max} = \frac{85 - 70}{10.5 \cdot \left(\frac{1}{0.81} - 1\right)} = 6.1 \text{ [Вт]}.$$

Модуль требует установки на теплоотводы с плотным прилеганием через теплопроводящую пасту (рисунок 28).

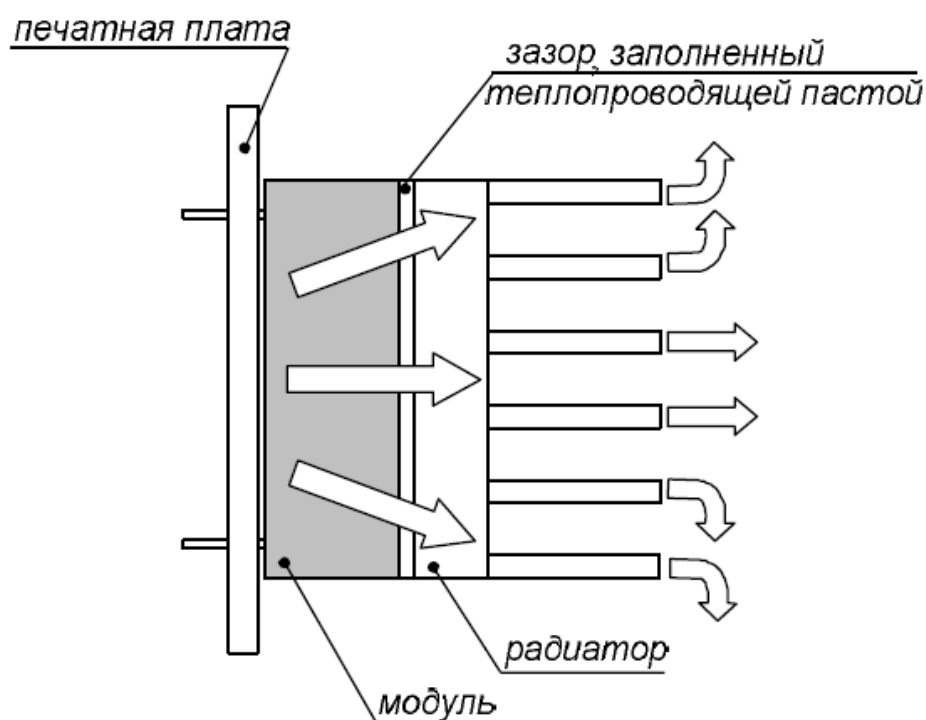


Рисунок 28 – Теплоотвод модуля



соответствии с техническими характеристиками данного типа рабочий отрезок 1, т.е. расстояние от опорной поверхности оправы (плоскости крепления) объектива до фокальной плоскости (плоскости МПИ) равен 19.25 мм, диаметр  $d$  равен 44 мм, а крепление объектива – трехлепестковый байонет. В конструкции корпуса прибора необходимо учесть эти характеристики.

Также, решено применить модульную структуру конструкции прибора, которая включает в себя:

- сенсорный модуль (плата МПИ);
- процессорный модуль (плата процессора);
- модуль питания (плата питания).

Для бесперебойной работы процессора в конструкции прибора необходимо выполнить систему охлаждения (осевой вентилятор с электродвигателем), однако воздействие потока воздуха и частиц пыли на МПИ приведет к некорректному приему оптического излучения. При проектировании корпуса прибора необходимо предусмотреть воздушную изоляцию сенсорного модуля от процессорного.

Построение трехмерной модели устройства будем проводить в САПР SolidWorks.

Редактор SolidWorks позволяет создавать трехмерные модели отдельных деталей, сборочные единицы, состоящие из нескольких деталей, и чертежи по деталям.

SolidWorks при моделировании может функционировать в трёх режимах: деталь, сборка, чертёж.

При работе в режиме «деталь» сначала создаётся необходимое количество эскизов для основания, а потом добавляются другие элементы.

Редактор SolidWorks позволяет произвольно изменять детали напрямую из сборки. Связь между деталями и сборкой гарантирует их синхронное обновление при изменении каких-либо параметров. Следовательно, чертежи сборки можно создавать на любом этапе проектирования.

					ЮУрГУ-120401.2017.071. ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		80



Для быстрого создания ряда однотипных объектов, могут быть использованы специальные команды «линейный массив» и «круговой массив». В круговом массиве пользователь указывает образец элемента, количество элементов в массиве и характерные размеры. Аналогично происходит создание линейного массива. Удобно то, что созданные таким образом подобные элементы по умолчанию не содержат взаимосвязей, т.е. изменение одного из них не приведёт к автоматическому изменению остальных.

Для преобразования эскизов (т.е. двухмерных объектов) в модели (трёхмерные объекты) существует ряд команд, наиболее часто используемые из которых – «Бобышка» и «Вращение». Команда «Бобышка» вытягивает эскиз перпендикулярно плоскости эскиза, а команда «Вращение» создаёт трёхмерный объект путём вращения эскиза относительно указанной оси симметрии.

В режиме чертежа создаются проекции моделей в привычной для конструктора форме. Режим «чертёж» позволяет создать не менее 6 стандартных видов, сечения в любой плоскости, ступенчатые сечения, выноски; проставить размеры, шероховатости, допуски на перпендикулярность, волнистость; создать и заполнить основную надпись.

Стандартные проекции объекта создаются автоматически при переносе детали или сборки в чертёж. Отображение других проекций можно задавать, используя стандартный диалог. Для создания сечения необходимо штрихпунктирной линией указать секущую плоскость на одной из проекций, а далее, используя команду «сечение», создать его и при необходимости переместить.

Для создания выносок в определённом месте проекции рисуется окружность, границы которой соответствуют границам выноски. Окружность можно перемещать и изменять её размер. Содержание и границы выноски изменяются автоматически.

Для сборки отдельных деталей в сборочную единицу необходимо создать специальный файл сборки (расширение .sldasm). Детали (файлы с расширением .sldpart) переносятся в определённом порядке в сборку и собираются по

					ЮУрГУ-120401.2017.071. ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		81

определённым правилам. Для сборки рекомендуется использовать стандартные команды, располагающие одну деталь относительно другой: «Расстояние», «Соосность», «Параллельность», «Перпендикулярность», «Угол».

SolidWorks позволяет задать материал (или его плотность) для детали или сборки, а затем произвести некоторые статические вычисления (масса, объём, осевые моменты инерции). По сравнению с AutoCAD SolidWorks предоставляет: упрощённое создание 3D-моделей, создание чертежей по моделям, простое по сравнению с AutoCAD создание сборок, создание моделей сварных швов, экспорт файла в двоичные форматы, поддерживаемые большинством CAD/CAE пакетов (IGES, ACIS, Parasolid, STL).

В результате в САПР SolidWorks были созданы трехмерные модели деталей (рисунок 30).

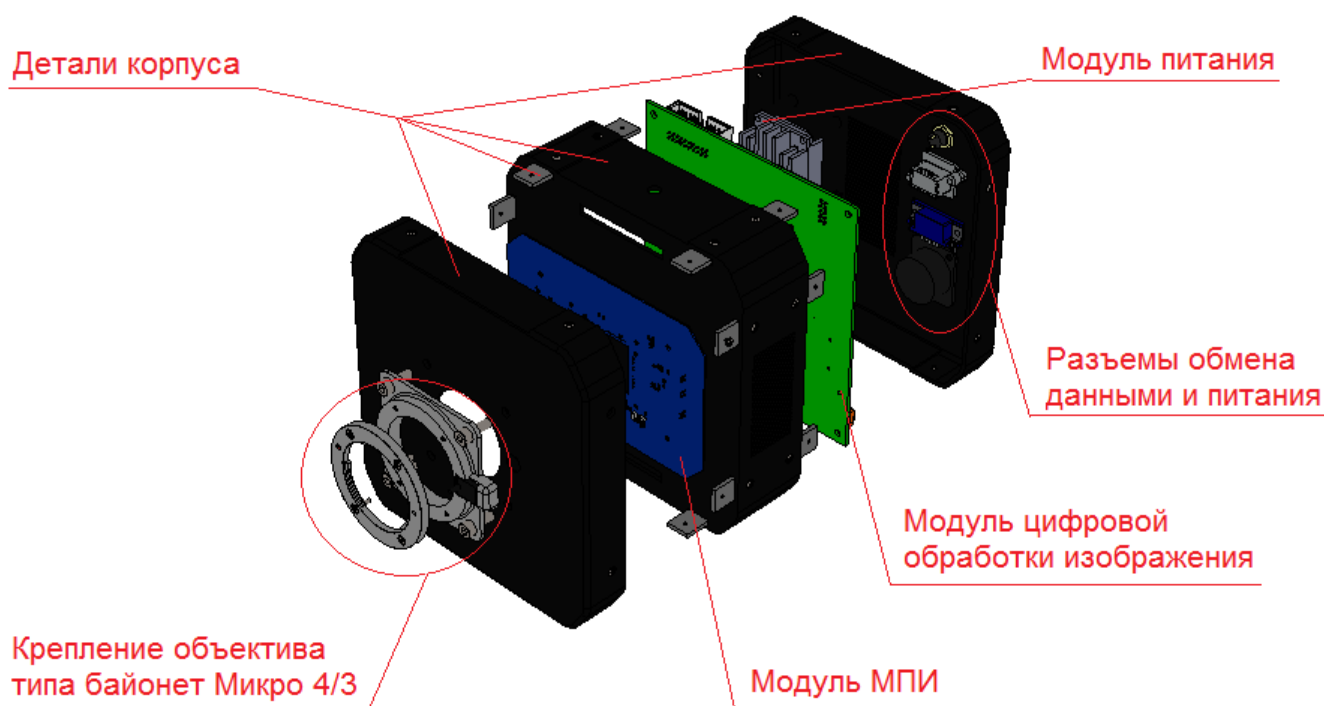
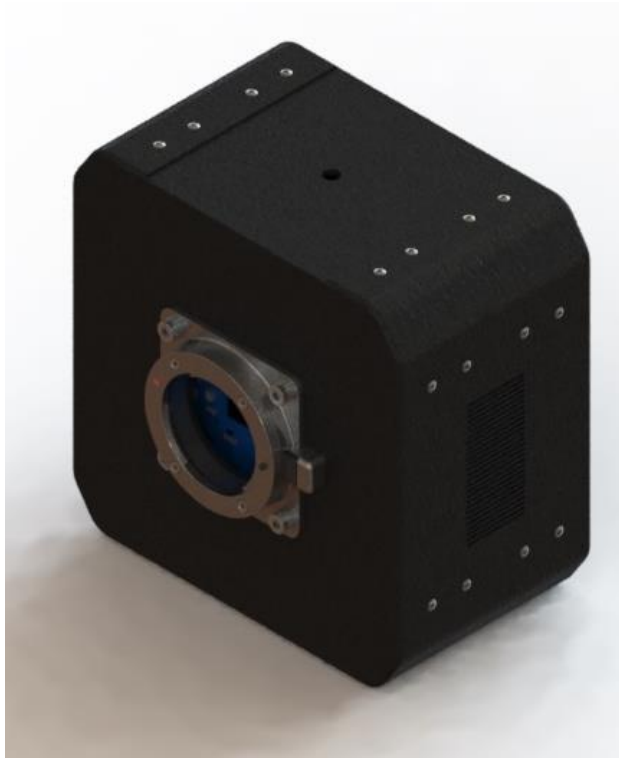


Рисунок 30 – Вид трехмерной модели с разнесенными деталями

А также сборка модели (Рисунок 31).



*а)*

*б)*

Рисунок 31 – Трехмерная модель в SolidWorks:

*а* – вид спереди; *б* – вид сзади

В результате теоретической разработки системы разработаны математическая модель преобразования оптических сигналов, функциональная схема системы, плата приемника излучения, используя современную элементную базу электронных компонентов и современных технологий проектирования печатных плат, определены тип оптической системы объектива и тип крепления объектива исходя из габаритных размеров матричного приемника излучения, разработан корпус, исходя из типа оптической системы и типа крепления объектива.

Рассчитаны пороговые значения потока излучения и освещенности в энергетических и световых величинах для приемника излучения, используемого в разработанной системе. Подобран процессорный модуль, также, подобран модуль питания отечественного производства для применения в разработанной системе.

## 3 РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ОБРАБОТКИ ЗРИТЕЛЬНЫХ ДАННЫХ

В работе представлен метод геометрического парсинга изображений городских сцен. Под геометрическим парсингом подразумевается процесс нахождения геометрических примитивов различного уровня сложности, а также межуровневых отношений между ними, который производится с помощью совместной оптимизации модели. В данном разделе последовательно описывается функция энергии для предложенной модели, далее алгоритм вывода, а также возможное расширение модели.

### 3.1 Описание модели

Повторим нотацию используемых геометрических примитивов. Краевые пиксели обозначаются как  $p = \{p_i\}_{i=1\dots p}$ . Отрезки и прямые обозначаются как соответственно  $s = \{s_i\}_{i=1\dots s}$  и  $l = \{l_i\}_{i=1\dots l}$ . За  $z$  обозначается вертикальная точка схода (зенит), которая соответствует группе вертикальных прямых, за  $h = \{h_i\}_{i=1\dots H}$  обозначаются горизонтальные точки схода, соответствующие горизонтальным группам параллельных прямых. Согласно сделанным предположениям, точки  $h_1, h_2 \dots h_H$  должны лежать близко к одной прямой на плоскости изображения (этот факт в дальнейшем будет называться как ограничение на горизонт).

Функция энергии представленной модели состоит из пяти частей, которые будут рассмотрены последовательно. Первые три части описывают взаимоотношения между различными геометрическими примитивами: между краевыми пикселями и отрезками, между отрезками и прямыми, а также между прямыми и точками схода. Четвёртая часть задаёт ограничение на горизонт. Пятая часть отвечает за реализацию принципа минимальной длины описания.

Первые три части включают в себя индивидуальные члены энергии, описывающие (псевдо)правдоподобие каждого краевого пикселя, каждого отрезка и каждой прямой.

					ЮУрГУ-120401.2017.071. ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		84

Слагаемое, отвечающее за краевые пиксели, записывается как

$$E_{edge}(p|s) = \min \left( \theta_{bg}, \min_{i=1 \dots S} \theta_{dist} \cdot d(p, s_i) + \theta_{grad} \cdot d_{angle}(p, s_i) \right), \quad (3.1)$$

где  $d(p, s_i)$  – евклидово расстояние между краевым пикселем  $p$  и отрезком  $s_i$  на плоскости изображения (минимальное расстояние между краевым пикселем  $p$  и точками отрезка  $s_i$ );

$d_{angle}(p, s_i)$  – угловая разность между направлением градиента, определённого в данном краевом пикселе  $p$ , и направлением нормали сегмента  $s_i$ ;

$\theta_{bg}$  – константа, которая отвечает за правдоподобие фонового шума;

$\theta_{grad}$  и  $\theta_{dist}$  – константы, задающие множество краевых пикселей, определяемых некоторым отрезком.

Так, если значение константы  $\theta_{dist}$  велико, то краевой пиксель может быть отнесён только к отрезку, находящемуся на небольшом евклидовом расстоянии от данного краевого пикселя. Если значение константы  $\theta_{grad}$  велико, тогда краевой пиксель может быть отнесён только к отрезку, для которого направление нормали не сильно отличается от направления градиента в данном пикселе.

Таким образом, слагаемое, отвечающее за данный пиксель  $p$ , принимает маленькое значение, если существует отрезок  $s_i$ , для которого евклидово расстояние до данного пикселя невелико, а также направление нормали не сильно отличается от направления градиента в пикселе. В противном случае, значение данного члена равно  $\theta_{bg}$ , что обозначает принадлежность данного пикселя к фоновому шуму.

Слагаемое, отвечающее за отрезки, записывается как

$$E_{segment}(s|l) = \min \left( \mu_{bg} * length(s), \min_{i=1 \dots l} \mu_{dist} \cdot d_{area}(s, l_i) \right), \quad (3.2)$$

где  $d_{area}(s, l_i)$  – расстояние между отрезком  $s$  и прямой  $l_i$  определяемое как площадь фигуры между прямой и отрезком, делённая на косинус угла между ними (по сути эта формула вычисляет значение интеграла от функции расстояния от отрезка до прямой вдоль длины отрезка);

$\mu_{bg}$  – это константа, обозначающая правдоподобие фонового шума;

$\mu_{dist}$  – константа, задающая множество отрезков, определяемых некоторой прямой.

Таким образом, слагаемое, отвечающее за данный отрезок  $s$ , принимает малое значение, если существует прямая  $l_i$ , для которой расстояние до отрезка невелико. В противном случае, значение данного члена равно  $\mu_{bg} * length(s)$ , что означает, что данный отрезок не может быть отнесён ни к одной из прямых. Такое описание энергии позволяет каждому отрезку вносить вклад в энергию, эквивалентный вкладу набора краевых пикселей, составляющих данный отрезок.

Слагаемое, отвечающее за прямые, записывается как:

$$E_{line}(l|h, z) = \min \left( \eta_{bg}, \min_{i=1 \dots H} \eta_{dist} \cdot \phi(l, h_i)^2, \eta_{dist} \cdot \phi(l, z)^2 \right) \quad (3.3)$$

где  $\phi$  – расстояние на гауссовой сфере между проекцией прямой  $l$  и проекцией, соответствующей точки схода ( $h_i$  или  $z$ );

$\eta_{bg}$  – константа, отвечающая за правдоподобие прямых, которые не являются ни горизонтальными, ни вертикальными;

$\eta_{dist}$  – константа, задающая множество прямых, определяемых некоторой точкой схода.

Таким образом, слагаемое, отвечающее за данную прямую  $l$ , принимает малое значение, если существует точка схода из множества  $h \cup \{z\}$ , для которой расстояние до прямой невелико. В противном случае, значение данного члена равно  $\eta_{bg}$ , что означает, что данная прямая не может быть отнесена ни к одной точке схода, т.е. не является ни горизонтальной, ни вертикальной.

Согласно ограничению на горизонт, все горизонтальные точки схода должны лежать близко к одной прямой на плоскости, которая называется горизонтом. Для задания этого ограничения, используется факт, что прямая, соединяющая зенит и принципиальную точку (обозначим её как  $L(z)$ , где  $z$  – это зенит), перпендикулярна горизонту. Что верно при заданных ограничениях на внутренние параметры камеры. Для выражения этого факта используются следующие парные члены энергии, вычисляемые для каждой пары горизонтальных точек схода:

$$E_{horizon}(u, h|z) = k_{hor} \cdot \tan \varphi (u - h, L(z)), \quad (3.4)$$

где  $u, h$  – две горизонтальные точки схода;

$\varphi$  – абсолютный угол между вектором  $\overline{uh}$  и перпендикуляром к  $L_z$ ;

$k_{hor}$  – константа.

Графическое представление выражения (3.4) изображено на рисунке 32.

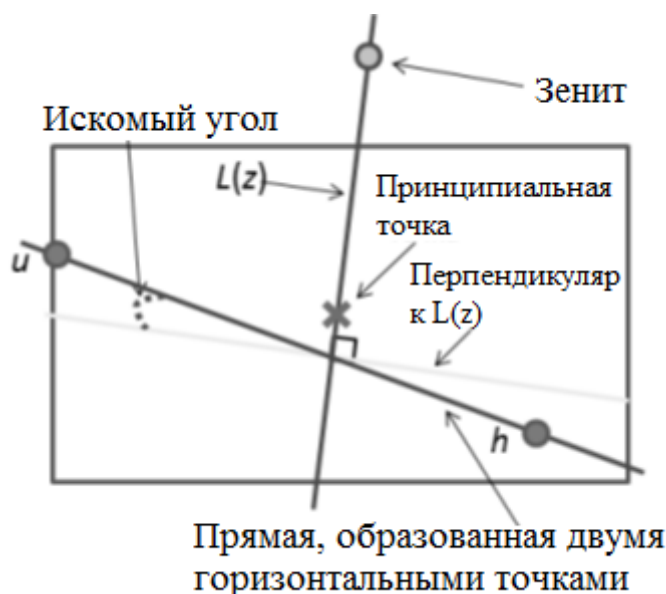


Рисунок 32 – Графическое представление ограничения на горизонт:  $u, h$  – две горизонтальные точки схода

В формуле (3.3) была выбрана функция тангенс, т.к. она позволяет задать большой штраф для больших значений углов (до  $+\infty$ ), означающих строгую перпендикулярность горизонта к прямой  $L_z$ .

Последняя часть энергии реализует принцип минимальной длины описания:

$$E_{prior}(s, l, h) = \lambda_{line}(|l|) + \lambda_{vp}(|h|) + \lambda_{segment} \sum_{i=1 \dots S} length(s_i), \quad (3.5)$$

где  $\lambda_{segment}$ ,  $\lambda_{line}$ ,  $\lambda_{vp}$  – константы, регулирующие вклад данного члена в энергию.

Этот член штрафует количество сегментов  $|s| = S$ , количество прямых  $|l| = L$  и количество горизонтальных точек схода  $|h| = H$ , таким образом, что предпочтение отдаётся более простому описанию изображения, с меньшим количеством элементов. Для отрезков каждое слагаемое, отвечающее за данный отрезок, умножается на его длину. Это позволяет присваивать больший штраф более длинным отрезкам, таким образом балансируя энергию на разных уровнях модели. Длинный отрезок может быть включён в модель, только если достаточное количество краевых пикселей относятся к данному отрезку.

Окончательно, получаем следующую формулу, задающую энергию модели:

$$E_{total}(s, l, h, z|p) = \sum_{i=1 \dots P} E_{edge}(p_i|s) + E_{segment}(s_i|l) + \\ + \sum_{i=1 \dots L} E_{line}(l_i|h, z) + \sum_{1 \leq i \leq j \leq H} E_{horizon}(h_i, h_j|z) + E_{prior}(s, l, h) \quad (3.6)$$

Таким образом, энергия (3.6) включает в себя геометрические примитивы различных уровней произвольного не-Манхэттенского изображения. Геометрический парсинг изображения может быть произведён с помощью минимизации этой энергии 3.6 по вектору отрезков, прямых и точек схода.



### 3.2 Алгоритм вывода для модели

Минимизация энергии (3.6) – очень трудная вычислительная задача, поэтому при её решении сложно обойтись без использования приближений. Один из возможных способов минимизации – использование послойного алгоритма. При таком подходе сначала находится набор отрезков по множеству краевых пикселей, затем – набор прямых по отрезкам, далее выбираются точки схода по найденным прямым, и в самом конце на основании найденных точек схода выбирается зенит и вычисляется горизонт. Существует и другая альтернатива данному подходу, активно представленная в различных работах, заключающаяся в использовании EM-алгоритма, хотя на практике такой подход часто находит локальный минимум энергии, близкий к начальному жадному приближению.

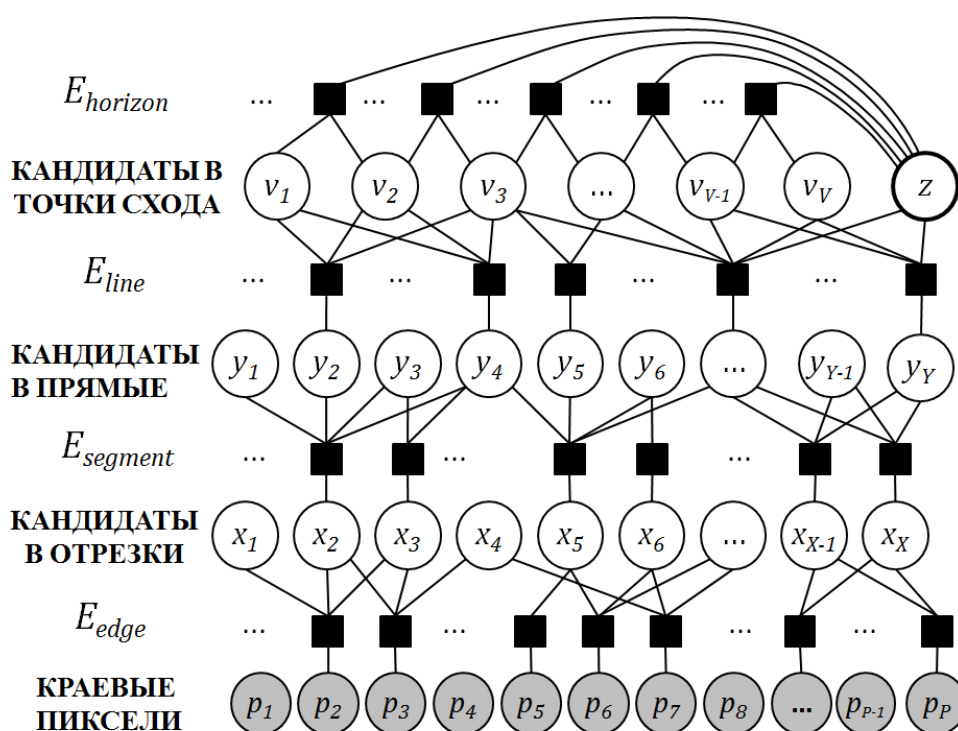


Рисунок 33 – Графическая модель для дискретного приближения энергии (4.6)

Переменные  $x_1 \dots x_S, y_1 \dots y_L, v_1 \dots v_H$  – бинарные, они отвечают за присутствие или отсутствие на изображении кандидатов в отрезки, прямые и горизонтальные точки схода соответственно. Переменная  $z$  обозначает положение зенита и может

принимать одно из фиксированных значений из множества точек, вычисленных заранее как возможное положение зенита. На изображении не показаны унарные клики, отвечающие переменным  $x_1 \dots x_S, y_1 \dots y_L, v_1 \dots v_H$ . Затенённые вершины обозначают краевые пиксели, которые известны как во время обучения модели, так и во время тестирования алгоритма.

В данной работе представлен другой подход, основанный на использовании дискретного приближения первоначальной энергии, которое легче минимизировать. Для этого сначала выполняются три шага стандартного последовательного подхода, но с параметрами, позволяющими получить множества  $X, Y, V$ , содержащие достаточно большое количество отрезков  $\hat{s}_1 \dots \hat{s}_X$ , прямых  $\hat{l}_1 \dots \hat{l}_Y$ , а также точек схода  $\hat{h}_1 \dots \hat{h}_V$  соответственно. На практике можно использовать любые алгоритмы для нахождения отрезков, прямых и точек схода.

Таким образом, первоначальная задача минимизации энергии 3.6 может быть сведена к минимизации энергии с дискретными переменными  $x = \{x_i\}_{i=1\dots X}$ ,  $y = \{y_i\}_{i=1\dots Y}$ ,  $v = \{v_i\}_{i=1\dots V}$ , и  $z$ . Здесь каждая переменная  $x_i$  – бинарная, и она определяет присутствует ли отрезок  $\hat{s}_i$  на изображении ( $x_i = 1$ ) или отсутствует ( $x_i = 0$ ). Аналогично, каждая переменная  $y_i$  – бинарная, и она определяет, присутствует ли прямая  $\hat{l}_i$  на изображении ( $y_i = 1$ ) или нет ( $y_i = 0$ ), и каждая переменная  $v_i$  – бинарная, и она определяет присутствует ли точка схода  $\hat{h}_i$  на изображении ( $v_i = 1$ ) или нет ( $v_i = 0$ ). И наконец, переменная  $z$  принимает значение двумерной точки на плоскости изображения, отвечающей зениту. Множество значений, которое может принимать эта переменная, ограничено конечным множеством вычисленных заранее кандидатов в зениты, выбранных из множества точек схода  $V$ . Для вычислительной эффективности также можно исключить из этого множества точки схода, которые отклоняются от вертикальной оси, проходящей через принципиальную точку, более чем на угол 7.5 градусов. Это можно рассматривать как дополнительные ограничения на положение зенита  $z$  и на энергию.



2) соответствующая бинарная переменная  $y_i$  равна 1 и прямая относится к фоновому шуму, тогда к энергии добавляется значение константы  $\eta_{bg}$ ;

3) соответствующая бинарная переменная  $y_i$  равна 1 и прямая относится к некоторому включённому кандидату в точки схода, тогда к энергии добавляется расстояние до этой точки схода.

Каждая точка схода может быть в двух состояниях:

1) значение бинарной переменной  $v_i$  равно 0, тогда точка схода выключена и никак не влияет на энергию;

2) значение бинарной переменной  $v_i$  равно 1, тогда эта точка схода включена и она участвует в вычислении члена, отвечающего за ограничение на горизонт.

Кроме того, для каждого включённого отрезка, прямой и точки схода к энергии добавляется дополнительный член, отвечающий за принцип минимальной длины описания.

Несмотря на то, что выключенные кандидаты в модели никак не штрафуются, минимум энергии не достигается, когда все кандидаты выключены. Если значение всех переменных  $x, y, v$  равны 0, то в этом случае значение энергии равно

$$\sum_{i=1\dots P} E_{edge}(p_i | \{\hat{s}_j\}_{j:x_j=1}) = \sum_{i=1\dots P} \theta_{bg}. \quad (3.9)$$

Очевидно, при адекватном значении параметров, данное значение не является минимально возможным значением энергии, и при выборе подходящих подмножеств отрезков, прямых и точек схода, оно может быть уменьшено.

Фактор-граф, соответствующий формуле 3.8, показан на рисунке 33. Граф состоит из четырёх уровней вершин, каждый из которых отвечает одному типу геометрических примитивов. Самый нижний уровень соответствует краевым пикселям, которые изначально заданы и обозначаются как  $p$  в энергии. Остальные три уровня соответствуют отрезкам, прямым и точкам схода, каждая вершина этих трёх уровней обозначает дискретную переменную из энергии  $(x, y, v, z)$ .

Также граф включает в себя факторы, которые описывают связи между геометрическими примитивами и соответствуют слагаемым энергии модели. Здесь следует отметить, что каждый геометрический примитив не соединяется со всеми геометрическими примитивами следующего уровня. Так, нет необходимости соединять каждый краевой пиксель со всеми отрезками, поскольку большая часть отрезков находится слишком далеко от данного пикселя, чтобы этот пиксель мог быть отнесён к этим отрезкам. Поэтому каждый краевой пиксель соединяется только с теми отрезками, расстояние до которых  $\theta_{dist} \cdot d(p, s_i) + \theta_{grad} \cdot d_{angle}(p, s_i)$  меньше, чем  $\theta_{bg}$ . Аналогично, отрезки соединяются рёбрами только с ближайшими прямыми, и каждая прямая соединяется с ближайшими точками схода.

Минимизация энергии производится с помощью алгоритма вывода в фактор-графе. Предложенный алгоритм вывода представлен в виде псевдокода. В алгоритме перебираются все возможные значения переменной  $z$ . При фиксированном зените производится минимизация энергии по бинарным переменным  $x, y, v$  с помощью модифицированного алгоритма имитации отжига со случайным порядком обхода вершин. В начале устанавливается значение температуры  $k_{hor}$ . Затем в цикле выбираются случайно по две точки и проверяется, что произойдёт при переключении соответствующих переменных, Если значение новой энергии отличается от текущей не более, чем на значение текущей температуры, то изменения принимаются. Аналогичная процедура проходит для прямых и для отрезков, однако в отличие от точек, одновременно переключается только одна переменная. Это связано прежде всего с особенностями энергии: в ней содержатся парные члены с точками схода и только унарные члены для отрезков и прямых. Из-за этого наблюдается лучшая сходимость метода при парном переключении точек. После каждой итерации алгоритма имитации отжига значение температуры уменьшается. Весь процесс повторяется, пока значение температуры не станет слишком мало. В проведённых

экспериментах обычно требовалось порядка 300 шагов для выполнения вывода. Алгоритм вывода для геометрического парсинга представлен в приложении А.

### 3.3 Технические детали реализации

Представленный алгоритм требует выполнения предобработки для нахождения первоначального приближения множеств геометрических примитивов. В проведённых экспериментах использовались следующие алгоритмы для нахождения кандидатов в отрезки, прямые и точки схода.

Для нахождения отрезков использовался алгоритм [35]. Для того, чтобы получить избыточное количество отрезков, алгоритм применялся к пирамиде изображений с тремя различными масштабами. В сумме обычно находилось около 2000 отрезков, полученных на трёх изображениях пирамиды.

Для нахождения прямых использовалась вероятностная модель преобразования Хафа, предложенная в [36]. Данный алгоритм возвращает набор прямых, причём каждой прямой сопоставляется некоторое значение уверенности. Для экспериментов выбирались 500 прямых с наибольшим значением уверенности.

Кандидаты в точки схода находились с помощью J-linkage процедуры, описанной в [37]. Этот метод является рандомизированным, поэтому при разных запусках и разных начальных приближениях, алгоритм находит различные точки схода. Для получения множества кандидатов в точки схода, данный алгоритм запускался 3 раза, при этом обычно находилось от 50 до 100 точек. При этом после запуска вывода в предложенной модели, выбиралось от 2-х до 5-ти точек схода.

Предложенный метод был реализован на языке C++ (в виде MEX-функций), а также на Матлабе. Основная вычислительно ёмкая часть алгоритма была реализована на C++, в силу высокой производительности языка. Графическая визуализация производилась на языке Матлаб из-за простоты реализации.

Для представления фактор-графа была разработана система классов. Диаграмма классов представлена на рисунке 34.

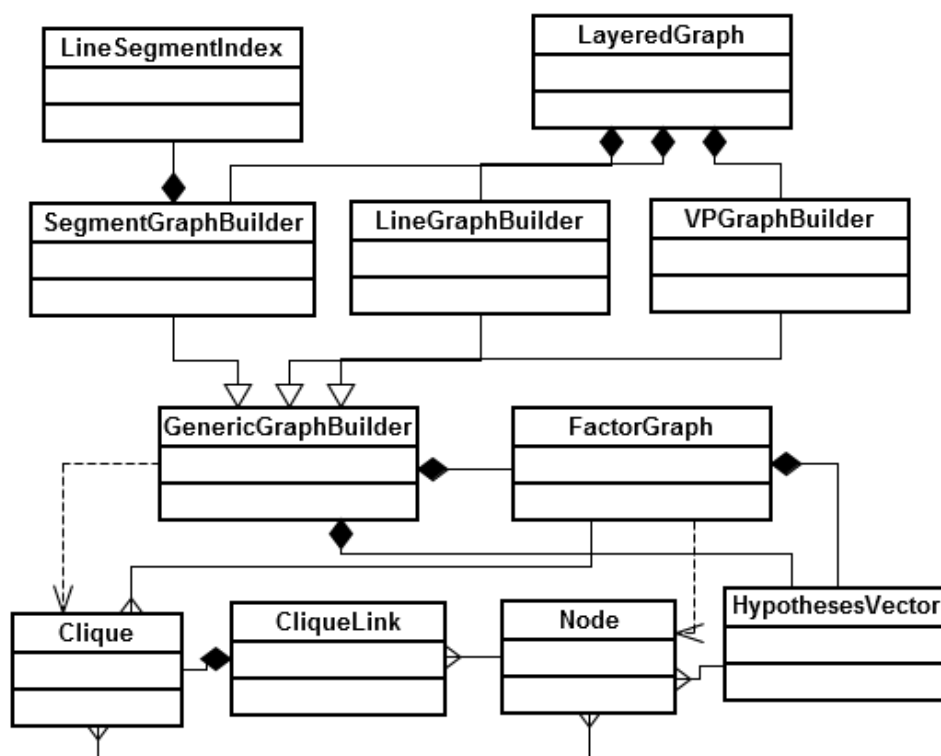


Рисунок 34 – Диаграмма классов

Основной класс, хранящий фактор-граф – LayeredGraph, состоящий из трёх классов-уровней SegmentGraphBuilder, LineGraphBuilder и VPGraphBuilder. Классы, соответствующие уровням графа, имеют общий базовый класс GenericGraphBuilder, включающий в себя класс FactorGraph. В этом классе хранится вся информация о вершинах (Node) и факторах (Clique) для данного уровня. Класс CliqueLink представляет связи между факторами и вершинами, а класс HypothesesVector включает информацию обо всех вершинах. Класс LineSegmentIndex используется для предварительного индексирования отрезков, которое позволяет значительно ускорить метод.

Параметры представленной модели настраивались на обучающей выборке изображений. В процессе настройки параметров результат работы алгоритма

оценивался с помощью анализа полученных геометрических примитивов и взаимоотношений между ними. Полностью автоматическая настройка параметров довольно затруднительна: с помощью имеющейся разметки данных в базах изображений и введённой метрики, можно настроить параметры для точного нахождения горизонта на обучающей выборке, но хорошая оценка положения горизонта ещё не означает хорошую работу алгоритма в целом, т.к. результатом работы алгоритма также являются отрезки, прямые и точки схода.

Как уже говорилось, моделирование (графическая визуализация) проводилась в пакете прикладных программ MatLab R2016b, с использованием Computer Vision System Toolbox, а также Image Acquisition Toolbox.

Детектирование особых точек на изображении средствами языка MatLab осуществляется следующими строками кода:

```
ref_pts = detectSURFFeatures(ref_img_gray);  
[ref_features,ref_validPts] = extractFeatures(ref_img_gray,ref_pts);
```

Визуализация первых 25 особых точек можно осуществить следующим методом:

```
figure;  
subplot(5,5,3); title('First 25 Features');  
for i=1:25  
    scale = ref_pts(i).Scale;  
    image = imcrop(ref_img,[ref_pts(i).Location-10*scale 20*scale 20*scale]);  
    subplot(5,5,i);  
    imshow(image);  
    hold on;  
    rectangle('Position',[5*scale 5*scale 10*scale 10*scale],'Curvature',1,'EdgeColor','g');  
end
```

Сопоставление особых точек, найденных на эталонном изображении (ref\_features) и исследуемом (I\_features) с последующей визуализацией реализовано следующими строками кода:



```

index_pairs = matchFeatures(ref_features, I_features);
ref_matched_pts = ref_validPts(index_pairs(:,1)).Location;
I_matched_pts = I_validPts(index_pairs(:,2)).Location;
figure, showMatchedFeatures(pic, ref_img, I_matched_pts, ref_matched_pts, 'montage');

```

Также, была реализована геометрическая трансформация эталонного изображения, посредством алгоритма RANSAC:

```

gte = vision.GeometricTransformEstimator;
gte.Method = 'Random Sample Consensus (RANSAC)';
[tform_matrix, inlierIdx] = step(gte, ref_matched_pts, I_matched_pts);
ref_inlier_pts = ref_matched_pts(inlierIdx,:);
I_inlier_pts = I_matched_pts(inlierIdx,:);

```

Полный текст программы приведен в приложении Б. Тексты функций, разработанных для применения в программе представлены в приложении В.

В данной главе показано, что задача оценки геометрии городской сцены может быть решена с помощью одновременного нахождения отрезков, прямых, их группировки в параллельные семейства, точек схода и зенита с горизонтом.

Разработано программное обеспечение реализующее предложенный метод.

Предложенный метод был реализован на языке C++ (в виде MEX-функций), а также на Матлабе. Основная вычислительно ёмкая часть алгоритма была реализована на C++, в силу высокой производительности языка. Графическая визуализация проводилась на языке Матлаб из-за простоты реализации.

#### 4 АНАЛИЗ ЭКСПЕРИМЕНТАЛЬНЫХ ДАННЫХ

Моделирование разработанных алгоритмов с последующей визуализацией проводилось в пакете прикладных программ MatLab R2016b. Результаты показаны на рисунках 35.

В качестве входных данных для моделирования использовались фотографии, сделанные на фотоаппарат Olympus OMD EM10.

В качестве тестовых изображений были использованы различные фотографии городской среды г. Челябинска. Для демонстрации результатов работы программы объектом-целью выбрана дверь центрального входа главного учебного корпуса ЮУрГУ. Такой массивный пространственный объект как фасад ЮУрГУ демонстрирует наглядный пример кажущихся сокращений размеров, изменений очертаний формы и светотеневых отношений, наблюдаемых в натуре.

На рисунке х. показан объект-цель (а) и анализируемое изображение (б).



а)



б)

Рисунок 35 - Входные данные:

а – изображение объекта-цели; б – анализируемое изображение

Как уже приводилось ранее алгоритм программы подразумевает определение особых точек на изображении посредством вычисления гессиана функции яркости изображения и сопоставление этих точек между анализируемым и эталонным изображением.

На рисунке 36 а, б показаны вычисленные особые точки на изображении объекта-цели и на анализируемом изображении.

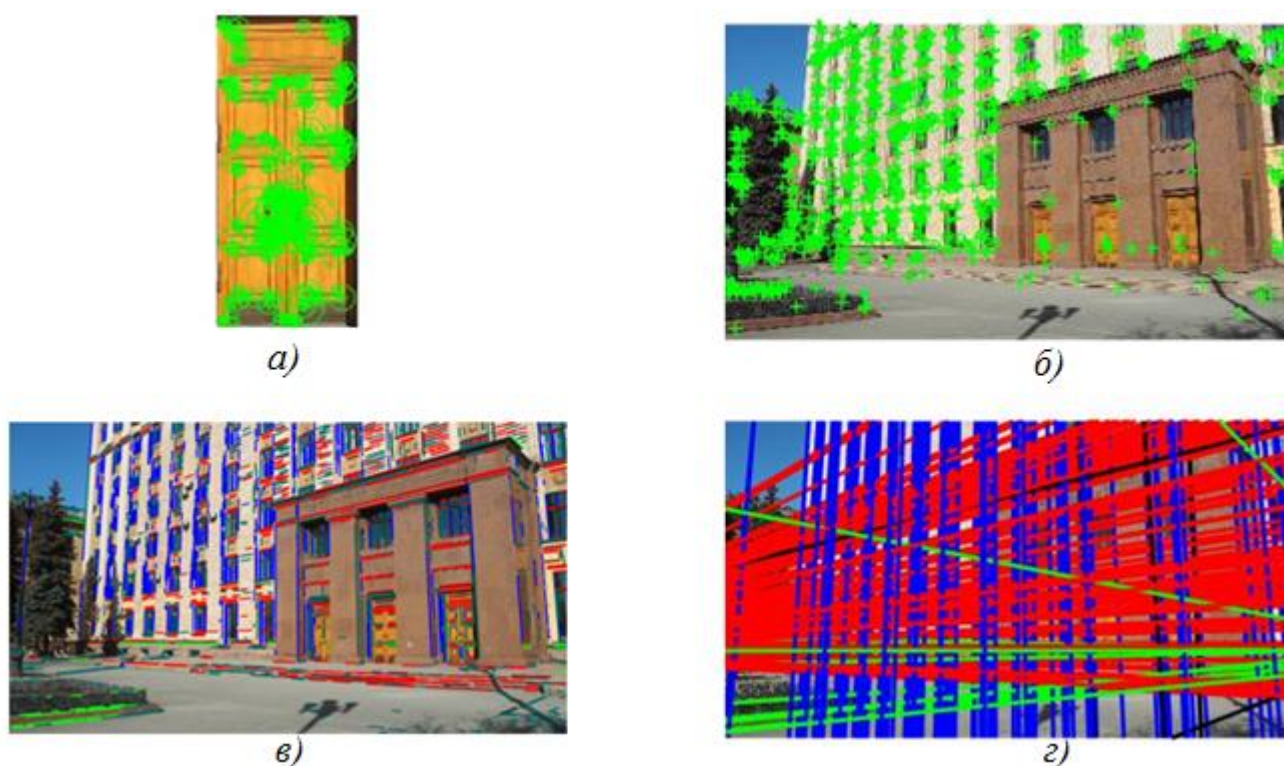


Рисунок 36 - Обработка изображения:

*а* – особые точки на изображении объекта-цели; *б* – особые точки на анализируемом изображении; *в* – группы параллельных отрезков; *г* – группы параллельных прямых

Определение горизонта и зенита на изображении требует определения точек схода. В свою очередь, для определения точек схода необходимо найти группы параллельных отрезков и группы параллельных прямых.

На рисунке 36 в, г показаны найденные группы параллельных отрезков и прямых.

Также, алгоритм подразумевает отсечение неверно найденных особых точек на изображениях, что наглядно показано на рисунке 37 а, б.



а)



б)

Рисунок 37 – Обнаружение цели:

а – сопоставление «особых» точек; б – удаление неверно найденных «особых» точек

Конечный результат работы программы представлен на рисунке х. На изображении найден горизонт, зенит, а также искомый объект-цель.

Определение местоположения объекта-цели проводилось с учетом геометрической трансформации (поворот, масштабирование объекта относительно плоскости светочувствительного элемента фотоаппарата, МОП-матрицы). Определение расстояние до объекта проводилось по масштабу изображения объекта-цели, с учетом фокусного расстояния оптической системы (объектива Olympus M.Zuiko Digital 14-42 mm). Положение определялось

относительно главной оптической оси фотоаппарата, т.е. геометрического центра плоскости изображения.



Рисунок 38 – Результат работы программы

Моделирование алгоритма проводилось в операционной среде Windows 10 с предустановленным пакетом прикладных программ Matlab R2016b и средой разработки Microsoft Visual Studio 2010 на ноутбуке Asus K50IN, используя два различных процессора, исполняющих машинные инструкции. В первом случае был задействован центральный процессор Intel Core 2 Duo, обладающий двумя вычислительными ядрами, одно из которых было задействовано для выполнения кода программы. Во втором же случае было задействовано 14 вычислительных ядер графического процессора Nvidia G102M с применением технологии параллельного программирования Nvidia Cuda.

Также было зафиксировано время выполнения различных частей алгоритма, которое представлено в таблице 8.

					ЮУрГУ-120401.2017.071. ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		101

Таблица 8 – Временная оценка выполнения алгоритма

Задействованный процессор Массив данных	Центральный процессор		Графический процессор	
	3000x2000	1600x1200	3000x2000	1600x1200
Определение кандидатов в отрезки	1.15	1.02	0.41	0.33
Определение кандидатов в прямые	9.81	5.42	3.23	1.86
Определение кандидатов в точки схода	3.12	1.01	1.06	0.35
Определение особых точек	3.56	2.38	1.16	0.79
Определение соответствия с шаблоном	1.89	1.54	0.67	0.57
Суммарное время	19.53	11.37	6.53	3.9

Как видно из полученных данных, вычисление на графическом процессоре сократило время выполнения программы в три раза, откуда вытекает следующий вывод: компьютерное зрение является одной из областей, в которых можно эффективно использовать вычисления на графическом процессоре для широкого круга актуальных задач, включая не только анализ городской среды и распознавание объектов для робототехники, но и такие области как детектирование лиц и пешеходов, восстановление трёхмерных поверхностей. Многие алгоритмы такого рода можно эффективно распараллеливать по данным и выполнять на графическом процессоре с более высокой скоростью.

## ЗАКЛЮЧЕНИЕ

В результате выполнения выпускной квалификационной работы была проведена разработка оптической системы локальной навигации, которая включала в себя:

- 1) проведение обзора современных методов цифровой обработки изображения и систем навигации, работающих на их основе;
- 2) составление математической модели преобразования оптических сигналов;
- 3) расчет пороговых значений потока излучения и освещенности;
- 4) разработку функциональной схемы системы;
- 5) разработку печатной платы МПИ;
- 6) анализ современных высокопроизводительных цифровых устройств обработки сигналов, выбор и обоснование для применения в разрабатываемой системе;
- 7) разработку корпуса системы;
- 8) разработку программы обработки изображения.

Обзор современных технических отечественных решений в области оптических систем навигации показал, что задачи навигации с применением систем технического зрения являются актуальными на сегодняшний день.

Составление математической модели преобразования оптических сигналов и результаты расчета пороговых значений потока излучения и освещенности показали, что за счет увеличения времени накопления заряда, можно добиться значительного уменьшения пороговой освещенности; меньшее значение квантовой эффективности, размера пикселя и коэффициента использования излучения источника следует меньшим значениям пороговой освещенности и потока излучения.

Были разработаны плата приемника излучения и корпус системы, используя современную элементную базу электронных компонентов и современных технологий проектирования печатных плат.

					ЮУрГУ-120401.2017.071. ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		103

Также, для решения задачи анализа геометрии городских сцен и поиска геометрических примитивов были представлены различные алгоритмы. В данной работе показано, что задача оценки геометрии городской сцены может быть решена с помощью одновременного нахождения отрезков, прямых, их группировки в параллельные семейства, точек схода и зенита с горизонтом.

Возможны различные продолжения представленной работы. В настоящее время в модели никак не учитывается информация о семантике объектов в сцене. Например, информация о положении неба или дороги может быть также полезной информацией для оценки геометрии сцены.

					ЮУрГУ-120401.2017.071. ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		104



## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Ковалёв, В.А. Метод эластичных экспоненциальных деформаций для совмещения цифровых изображений / В.А.Ковалёв // Сб. науч. Тр. «Цифровая обработка изображений». – Минск. – ИТК НАН Беларуси. – 1999. – С.147-156.
2. Мельниченко, А.С. Автоматическая аннотация изображений на основе выделения низкоуровневых текстурных признаков / А.С.Мельниченко // Известия ЮФУ. Технические науки. Тематический выпуск «Актуальные проблемы математического моделирования». – №8. – 2009. – С.193.
3. Липатов А.Н., Ляш А.Н., Макаров В.С., Антоненко, С.А., Захаркин Г.В. Оптический датчик координат для системы автоматической посадки // Сб. тр. науч.-техн. конф. «Современные проблемы определения ориентации и навигации космических аппаратов». М., 2008. – С. 146-156.
4. Ваулин Ю.В., Костенко В.В., Павин А.М. Особенности навигационного и алгоритмического обеспечения телеуправляемого необитаемого подводного аппарата// Подводные исследования и робототехника. 2013. № 2(16). – С. 4-16.
5. Артюхов М.Ю., Кропотов А.Н., Макашов А.А., Сахарова Е.И. Опыт создания системы локальной видеонавигации для подводных аппаратов II Сб. тр. науч.-техн. конф. «Техническое зрение в системах управления мобильными объектами 2010». М., 2012. – С. 230-242.
6. Barnard, S. Interpreting perspective images / S. Barnard // Artificial Intelligence. – 1983. – Vol. 4. – P. 435-462.
7. Hassanien, A.E. Image morphing with scattered data point based on snakes and thin plate spline transformation / A.E.Hassanien, M.Nakajima // Machine Graphics & Vision. – 1997. – Vol.6. – No. 3. – P.341–451.
8. Форсайт Д. Компьютерное зрение. Современный подход / Д. Форсайт, Ж. Понс – М.: Вильямс, 2004 г. – 928 с.
9. Collins R., Weiss R. Vanishing point calculation as a statistical inference on the unit sphere // ICCV. 1990. – P. 400-403.

					ЮУрГУ-120401.2017.071. ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		105

10. Шапиро Л. Компьютерное зрение/ Л. Шапиро, Дж. Стокман. – М.: Бином. Лаборатория знаний, 2006 г. – 752 с.
11. Boulanger, K. A tool for 3d navigation inside a single image with automatic camera calibration // EG UK Theory and Practice of Computer Graphics / Boulanger K., Bouatouch K., Pattanaik S. Atip. 2006. – P.38-45.
12. Pitas, I. Digital Image Processing Algorithms / I.Pitas // Prentice Hall. – New York. – 1993. – P.361.
13. Denis P., Elder J. H., Estrada F. J. Efficient edge-based methods for estimating manhattan frames in urban imagery // ECCV. 2008. – P. 197-210.
14. Kosecka J., Zhang W. Video compass // ECCV. 2002. 4. P. 476-490.
15. Umbaugh, S.E. Computer Vision and Image Processing / S.E.Umbaugh // Prentice Hall PTR. – Upper Saddle River. – 1999. – P.504.
16. Schindler G., Dellaert F. Atlanta world: An expectation maximization framework for simultaneous low-level edge grouping and camera calibration in complex man-made environments // CVPR. 2004. – P. 203-209.
17. Kim Z. Geometry of vanishing points and its application to external calibration and realtime pose estimation: Tech. rep.: Institute of Transportation Studies, Research Reports. 2006. – P. 23-29.
18. Aguilera D. G., Lahoz J. G., Codes J. F. A new method for vanishing points detection in 3d reconstruction from a single view // Proc. of ISPRS Commission V. 2005. – P. 254-269.
19. Tardif J.-P. Non-iterative approach for fast and accurate vanishing point detection // ICCV. 2009. – P. 1250-1257.
20. Форсайт Д. Компьютерное зрение. Современный подход / Д. Форсайт, Ж. Понс – М.: Вильямс, 2004 г. – 928 с.
21. Харитоновна, Е.Н. Математическая модель выходного сигнала и геометрического шума матричных фотоприёмных устройств, учитывающая нелинейность характеристики чувствительности пикселей / Е.Н.Харитоновна // Вестник поморского университета: Ест. науки. – 2010. – 1. – С.117-122.

22. Методы автоматического обнаружения и сопровождения объектов. Обработка изображений и управление / Б. А. Алпатов, П. В. Бабаян, О. Е. Балашов, А. И. Степашкин – М.: Радиотехника, 2008 г. – 176 с.
23. Розенфельд А., Дейвис Л.С. Сегментация и модели изображений. ТИИЭР. 1979. Т.67, №5. С. 71-82.
24. Лазарев, Л.П. Оптико-электронные приборы наведения / Л.П. Лазарев – М.: Машиностроение, 1989. 512 с.
25. Криксунов, Л.З. Справочник по основам инфракрасной техники / Л.З. Криксунов. – М: Сов. Радио, 1978. – 396 с.
26. Орлов, П.Е. Методы и устройства контроля, диагностики и обеспечения электромагнитной совместимости электрических соединений аппаратуры космического применения: дис. к-та тех. наук: 05.12.07 / П.Е. Орлов – Томск, 2013. – 192 с.
27. Чермошенцев, Ф. Автоматизация проектирования печатных плат цифровых электронных средств с учетом электромагнитной совместимости: дис. д-ра тех. наук: 05.13.12, 05.13.05 / Чермошенцев С.Ф. – Казань, 2004. – 474 с
28. Князев, А.Д. Проектирование печатных плат для цифровой быстродействующей аппаратуры. / А.Д. Князев, Л.Н. Кечиев, Б.В. Петров. – М.: ООО Группа ИДТ, – 2007. – 616 с.
29. Иванов, Ю.А. Технологии компьютерного зрения в системах автоведения / Ю.А.Иванов // Автоматика, связь, информатика. – 2011. – №6. – С. 46-48.
30. Соколов, С.М. Интеллектуальные алгоритмы обработки изображений для решения задачи распознавания в реальном времени бортовыми системами / С.М.Соколов, А.А.Богуславский // Тезисы докладов научно-технической конференции-семинара. Техническое зрение в системах управления мобильными объектами. – М.: КДУ. – 2011. – С.86-88.
31. Nvidia. Intro to Parallel Programming. [Электронный ресурс]. Nvidia. – 2013. – Режим доступа: [http://www.nvidia.com/object/cuda\\_home\\_new.html](http://www.nvidia.com/object/cuda_home_new.html)

32. И. Плоткин, С. Маняшин, О. Комаров. АЛЕКСАНДЕР ЭЛЕКТРИК источники электропитания производитель унифицированных вторичных источников для вооружения и военной техники. Электронные Компоненты №4 2005.
33. И. Твердов. Охлаждение универсальных модулей питания. Электронные компоненты №8 2008.
34. R.G. von Gioi, J.Jakubowicz, J.-M. Morel, G. Randall LSD: A fast line segment detector with a false detection control // IEEE Transactions on Pattern Analysis and Machine Intelligence. 2010. 32. – P. 722-732.
35. Barinova O., Lempitsky V., Kohli P. On detection of multiple object instances using hough transforms // CVPR. 2010. – P. 2233-2240.
36. Tardif J.-P. Non-iterative approach for fast and accurate vanishing point detection // ICCV. 2009. P. 1250-1257.
37. Страуструп, Б. Язык программирования С++. / Б. Страуструп. М: Бином, 2011. – 1104 с.
38. Мейерс, С. Эффективное использование С++. 55 верных способов улучшить структуру и код ваших программ. / С. Мейерс. – СПб: ДМК Пресс, 2013. – 143.

## ПРИЛОЖЕНИЯ

### ПРИЛОЖЕНИЕ А

Алгоритм вывода для геометрического парсинга

```
1  минЭнергия ← ∞
2  для  $z \leftarrow$  кандидатыВЗенит [1] до кандидатыВЗенит[ $M$ ]
3      температура ← начТемпература
4      выключитьВсеТочкиСхода( $V$ )
5      включитьВсеПрямые( $Y$ )
6      включитьВсеОтрезки( $X$ )
7      включитьЗенит( $z$ )
8      пока температура >  $eps$ 
9          повторить  $StepsV$  раз
10             точка1 ←  $random(|V|)$ 
11             точка2 ←  $random(|V|)$ 
12             стоимПереключения ← посчитатьСтоимПереключения
(точка1, точка2)
13             если стоимПереключения < температура
14                 тогда переключитьТочки(точка1, точка2)
15             повторить  $StepsV$  раз
16                 прямая ←  $random(|Y|)$ 
17                 стоимПереключения ← посчитатьСтоимПереключения
(прямая)
18                 если стоимПереключения < температура
19                     тогда переключитьПрямую(прямая)
20             повторить  $StepsV$  раз
21                 отрезок ←  $random(|X|)$ 
```

22                                   стоимПереключения←посчитатьСтоимПереключения  
 (отрезок)

23                                   **если** стоимПереключения < температура

24   **тогда** переключитьОтрезок(отрезок)

25                                   температура ← температура \* коэфф

26                                   текЭнергия ← посчитатьТекЭнергию()

27                                   **если** текЭнергия < минЭнергия

28   **тогда** сохранитьКонфигурацию()

29                                   минЭнергия ← текЭнергия

## ПРИЛОЖЕНИЕ Б

### Текст программы обработки изображения

```
basetype = 'Other';

addpath ('../CPP/mexFunctions');
addpath ('../CPP/mxJLinkage');
addpath ('../');

pic_folder = '../PICS/';
out_pic_folder = '../PICS/';

im_idx = 14;
im_prefix = sprintf('%d', im_idx);

DRAW_GT_HORIZON = false;

LOAD_SEGMENT_CANDIDATES = false;
SAVE_SEGMENT_CANDIDATES = true;
LOAD_LINES_CANDIDATES = false;
SAVE_LINES_CANDIDATES = true;
LOAD_VP_CANDIDATES = false;
SAVE_VP_CANDIDATES = true;

display_all = true;

close all;

rgb_palette = [255, 0, 0;...
    0, 255, 0; ...
    0, 0, 255; ...
    128, 0, 128; ...
    128, 128, 0; ...
    0, 128, 128; ...
    255, 128, 0; ...
    255, 0, 128; ...
    128, 255, 0; ...
    128, 0, 255; ...
    0, 128, 255; ...
    0, 255, 128; ...
    0, 0, 0;...
    0, 100, 100];

%%%%%%%%%%%%%% параметры детектирования линий %%%%%%%%%%%%%%%

maxLines = 500; % максимальное кол-во линий
LineBGBias = 1.5e-4;
LineHypPenalty = 3.5e-3;
EdgeToLineDistCoef = 1e-4;
AngleBetweenGradientAndNormalCoef = 3e-4;

%%%%%%%%%%%%%% параметр Тардифа %%%%%%%%%%%%%%%
TardifsCount = 3;

%%%%%%%%%%%%%% загрузка изображения %%%%%%%%%%%%%%%
```

					ЮУрГУ-120401.2017.071. ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		111

```

pic_name = [pic_folder im_prefix '.jpg'];
pic = imread(pic_name);
picsize = min([size(pic, 1), size(pic, 2)]);
scale_koeff = 768/picsize;
pic = imresize(pic, 768/picsize);

imcanny = edge(rgb2gray(pic), 'canny'); % детектор границ Канни
edgeCount = sum(imcanny(:));

width = size(pic, 2);
height = size(pic, 1);
if (strcmp( basetype, 'York'))
    if (width > height)
        pp_imcoords = [307, 251];
    else
        pp_imcoords = [251, 307];
    end;
    pp_imcoords = pp_imcoords .* 768/picsize;
else
    pp_imcoords = [0.5*width, 0.5*height];
end

edgweightCoef = 2.0;

%%%%%%%%%%%%%% определение линий %%%%%%%%%%%%%%%
grayim = rgb2gray(pic);
lines_file = [pic_folder im_prefix '.txt'];

if (LOAD_LINES_CANDIDATES)

    vsLines = load(lines_file);

    disp('Line candidates loaded');
else

    disp('Finding line candidates');

    tStart=tic;
    [vsLines, ARGS.imE] = mxGetLines(grayim, ... % оттенки серого
        LineBGBias, ...
        LineHypPenalty, ...
        EdgelToLineDistCoef, ...
        AngleBetweenGradientAndNormalCoef, ...
        maxLines);

    tElapsed=toc(tStart);
    disp(['Elapsed time (finding line candidates): ' num2str(tElapsed) ' sec']);

    if (SAVE_LINES_CANDIDATES)
        save(lines_file, 'vsLines', '-ascii');
    end;

end;
if (display_all)

```



```

%%%%%%%%%%%%%% визуализация линий %%%%%%%%%%%%%%%
sfigure (1);
imshow(pic);
hold on;
for i = 1:size(vsLines, 1)

    a = cos(vsLines(i, 1));
    b = sin(vsLines(i, 1));
    x0 = a*vsLines(i, 2) + 1;
    y0 = b*vsLines(i, 2) + 1;

    x1 = x0 + 5000*(-b);
    y1 = y0 + 5000*(a);
    x2 = x0 - 5000*(-b);
    y2 = y0 - 5000*(a);

    theta = vsLines(i, 1);
    A = cos(theta);
    B = sin(theta);
    C = -sin(theta)*y0 - cos(theta)*x0;

    plot([x1 x2], [y1 y2], 'Color', 'black', 'LineWidth', 2);
end;

hold off;

%%%%%%%%%%%%%% сохранение изображения с линиями %%%%%%%%%%%%%%%
filename= sprintf( '%s\%s_lines.png', out_pic_folder, im_prefix );
print('-noui', '-r0', '-dpng', filename);

end

%%%%%%%%%%%%%% определение отрезков %%%%%%%%%%%%%%%

line_segments_file = [pic_folder im_prefix '_segments.txt'];

if (LOAD_SEGMENT_CANDIDATES)

    line_segments = load(line_segments_file);
    disp('Line segments loaded');

else

    disp('Finding line segments candidates');

    tStart=tic;
    [line_segments] = calc_candidates_segments_lsd(pic, display_all);
    tElapsed=toc(tStart);
    disp(['Elapsed time(line segments): ' num2str(tElapsed) ' sec']);

    if (SAVE_SEGMENT_CANDIDATES)
        save(line_segments_file, 'line_segments', '-ascii');
    end;
end;

```

%%%%%%%%%%%% визуализация отрезков %%%%%%%%%%

```

if (display_all)
    sfigure (1);
    imshow(pic);
    hold on;
    Nedge = size (line_segments, 1);
    colourmp = hsv(Nedge); % HSV colour map with Nedge entries
    colourmp = colourmp(randperm(Nedge,:)); % Random permutation
    for I = 1:Nedge
        plot(line_segments(I,1:2), line_segments(I,3:4), 'Color', colourmp(I,:), 'LineWidth', 1 );
    end
    hold off;

```

%%%%%%%%%%%% сохранение изображения с отрезками %%%%%%%%%%

```

filename= sprintf( '%s\%s_line_segments.png', out_pic_folder, im_prefix );
print('-noui', '-r0', '-dpng', filename);

```

end

%%%%%%%%%%%% вычисление точек схода %%%%%%%%%%

```

vp_coords_name = [pic_folder im_prefix '_vp.txt'];
nPtsHor_name = [pic_folder im_prefix '_nPtsHor.txt'];

```

```

infinite_zenith = [width / 2, height * 15];

```

```

if (LOAD_VP_CANDIDATES)

```

```

    vp_imcoords = load(vp_coords_name);
    nPtsHor = load(nPtsHor_name);
    disp('Vanishing points candidates loaded');

```

else

```

    disp('Finding vanishing points candidates');
    tStart=tic;
    [vp_imcoords, nPtsHor] = calc_candidates_tard(pic, TardifsCount, display_all);
    tElapsed=toc(tStart);

```

```

    disp(['Elapsed time(candidate vanishing points finding): ' num2str(tElapsed) ' sec']);

```

```

    if (SAVE_VP_CANDIDATES)

```

```

        save(vp_coords_name, 'vp_imcoords', '-ascii');
        save(nPtsHor_name, 'nPtsHor', '-ascii');

```

end;

end;

%%%%%%%%%%%%%% геометрический парсинг %%%%%%%%%%%%%%%

```
vpStr = [pic_folder im_prefix '_res_vp.txt'];
vpAssStr = [pic_folder im_prefix '_res_vpass.txt'];
```

```
if (strcmp (basetype, 'York'))
```

```
    params = [(0.00040843 )... %LineBGBias
              0.0045...      %LineHypPenalty
              1.2*0.000452... %EdgeToSegmentDistCoef
              (0.017842 / 1.55)... %VPsBGBias
              0.023...      %VPsHypPenalty
              (0.5561)...     %LineToVPDistCoef
              4.646554...    %VPsPairwisePenalty
              (0.000327 /800)... %SegmentToLineDistCoef
              0.000682...    %SegmentBGBias
              (0.00007 )...   %SegmentHypPenalty
              0.000452 * 1.2]; %AngleBetweenGradientAndSegmentCoef
```

```
else
```

```
    params = [(0.00040843 )... %LineBGBias
              0.0035...      %LineHypPenalty
              0.000542...    %EdgeToSegmentDistCoef
              0.009584...    %VPsBGBias
              0.035...      %VPsHypPenalty
              0.645...      %LineToVPDistCoef
              4.646554...    %VPsPairwisePenalty
              (0.000327 /800)... %SegmentToLineDistCoef
              0.000682...    %SegmentBGBias
              (0.00007 /30)... %SegmentHypPenalty
              0.000542]; %AngleBetweenGradientAndSegmentCoef
```

```
end
```

```
manhattanWorld = -1;
```

```
tStart=tic;
```

```
[vp, vpAss, vpSegmAss]= mxParsing(params, pp_imcoords, grayim, vsLines, vp_imcoords, nPtsHor, line_segments,
manhattanWorld);
tElapsed=toc(tStart);
```

```
time_parsing = tElapsed;
```

```
disp(['Elapsed time (geometric parsing): ' num2str(tElapsed) ' sec']);
```

```
if (display_all)
```

```
    close all;
    vp = vp.*(1/scale_koeff);
    hor = vp(1:end-1, :);
    zen_current_method = vp(end, :);
    weights = zeros(size(hor, 1), 1);
    for i=1:size(hor, 1)
        weights(i) = sum (vpAss == i);
    end;
```

```

pic_big = pic;
pic = imresize(pic, 1/scale_koeff);
width = size(pic, 2);
height = size(pic, 1);

sfigure;
imshow(pic_big);

if (strcmp (basetype, 'York'))
    if (width > height)
        pp_imcoords = [307, 251];
    else
        pp_imcoords = [251, 307];
    end;
else
    pp_imcoords = [0.5*width, 0.5*height];
end

hold on;

%%%%%%%%%%%% визуализация групп линий по цвету %%%%%%%%%%%%%

vpAss(vpAss == 0) = 13;
for i = 1:min([length(vpAss), length(vsLines)])

    a = cos(vsLines(i, 1));
    b = sin(vsLines(i, 1));
    x0 = a*vsLines(i, 2);
    y0 = b*vsLines(i, 2);

    x1 = x0 + 5000*(-b) + 1;
    y1 = y0 + 5000*(a) + 1;
    x2 = x0 - 5000*(-b) + 1;
    y2 = y0 - 5000*(a) + 1;

    if (vpAss(i) ~= -1)
        idxColor = min(vpAss(i), size(rgb_palette, 1));
        plot([x1 x2], [y1 y2], 'Color', rgb_palette(idxColor, :) / 255.0, 'LineWidth', 2);
    end

end;

filename= sprintf( '%s\\%s_result_lines.png', out_pic_folder, im_prefix );
print('-noui', '-r0', '-dpng', filename);

%%%%%%%%%%%% визуализация групп отрезков по цвету %%%%%%%%%%%%%
sfigure;
hold off;
clear axis;
imshow(pic_big);
hold on;

```

```

for i = 1:length(vpSegmAss)

    if (vpSegmAss(i) ~= -1)

        if (vpSegmAss(i) ~= 0)

            a = cos(vsLines(vpSegmAss(i), 1));
            b = sin(vsLines(vpSegmAss(i), 1));
            x0 = a*vsLines(vpSegmAss(i), 2);
            y0 = b*vsLines(vpSegmAss(i), 2);

            x1 = x0 + 5000*(-b) + 1;
            y1 = y0 + 5000*(a) + 1;
            x2 = x0 - 5000*(-b) + 1;
            y2 = y0 - 5000*(a) + 1;

            idxColor = min(vpAss(vpSegmAss(i)), size(rgb_palette, 1));
        else
            idxColor = 14;
        end;

        plot(line_segments(i,1:2)+ 1, line_segments(i,3:4)+1, 'Color', rgb_palette(idxColor, :) / 255.0 , 'LineWidth', 2);
    end

end;

filename= sprintf( '%s\\%s_result_segm.png', out_pic_folder, im_prefix );
print('-noui', '-r0', '-dpng', filename);

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% распознавание %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

disp('Start SURF');
%% Обработка эталонного изображения
ref_img = imread('etalon.jpg');
% ref_img = imresize(ref_img, [2000 3000]);
ref_img_gray = rgb2gray(ref_img);
ref_pts = detectSURFFeatures(ref_img_gray);
[ref_features, ref_validPts] = extractFeatures(ref_img_gray, ref_pts);

figure; imshow(ref_img);
hold on; plot(ref_pts.selectStrongest(1000));
%% визуализация 25 точек
figure;
subplot(5,5,3); title('First 25 Features');
for i=1:25
    scale = ref_pts(i).Scale;
    image = imcrop(ref_img,[ref_pts(i).Location-10*scale 20*scale 20*scale]);
    subplot(5,5,i);
    imshow(image);
    hold on;
    rectangle('Position',[5*scale 5*scale 10*scale 10*scale], 'Curvature',1, 'EdgeColor','g');
end

```

Изм.	Лист	№ докум.	Подпись	Дата

```

st_img = imread(pic_name);
st_img_gray = rgb2gray(st_img);
I_pts = detectSURFFeatures(st_img_gray);
[I_features, I_validPts] = extractFeatures(st_img_gray, I_pts);
figure;imshow(pic);
hold on; plot(I_pts.selectStrongest(1000));

%% сопоставление изображений со всеми особыми точками
index_pairs = matchFeatures(ref_features, I_features);

ref_matched_pts = ref_validPts(index_pairs(:,1)).Location;
I_matched_pts = I_validPts(index_pairs(:,2)).Location;

figure, showMatchedFeatures(pic, ref_img, I_matched_pts, ref_matched_pts, 'montage');
title('Showing all matches');

%% Геометрическая трансформация
gte = vision.GeometricTransformEstimator;
gte.Method = 'Random Sample Consensus (RANSAC)';

[tform_matrix, inlierIdx] = step(gte, ref_matched_pts, I_matched_pts);

ref_inlier_pts = ref_matched_pts(inlierIdx,:);
I_inlier_pts = I_matched_pts(inlierIdx,:);

% Построение линий особых точек эталона
figure;showMatchedFeatures(pic, ref_img, I_inlier_pts, ref_inlier_pts, 'montage');
title('Showing match only with Inliers');

tform = maketform('affine',double(tform_matrix));
[width, height,~] = size(ref_img);
corners = [0,0;height,0;height,width;0,width];
new_corners = tformfwd(tform, corners(:,1),corners(:,2));

%%%%%%%%%%%%% визуализация горизонта %%%%%%%%%%%%%%

sfigure;
imshow (pic);

patch(new_corners(:,1),new_corners(:,2),[0 1 0],'FaceAlpha',0.5);

p1 = [pp_imcoords(1) pp_imcoords(2) 1];
p2 = [zen_current_method(1) zen_current_method(2) 1];

l1 = cross (p1, p2);
l1 = l1(:)/norm (l1(1:2));

hold on;
hor_current_method = drawHorizont_orthogonal(hor, zen_current_method, weights, size (pic, 2), size (pic, 1), 'Magenta',
pp_imcoords);

if (DRAW_GT_HORIZON)

truehorfile = sprintf ('%s\\%03dhor.mat', pic_folder, im_idx );

```

					ЮУрГУ-120401.2017.071. ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		118

```

st = load (truehorfile);
l = st.horizon;
x = 1:width;
l = l / l(2);
y = -l(1)*x - l(3);
plot( x, y, 'Color', 'Yellow', 'LineWidth', 3);
end

```

```

p_inters = cross (l1, hor_current_method);

```

```

p_inters (:) = p_inters (:)/p_inters(3);
plot( [p_inters(1) zen_current_method(1)], [p_inters(2) zen_current_method(2)], '-.', 'Color', 'Cyan', 'LineWidth', 3 );

```

```

x = pp_imcoords(1);
y = pp_imcoords(2);
r = 5;
hold on;
line ([x-r x+r], [y-r y + r], 'Color', 'Red', 'LineWidth', 3);
line ([x-r x+r], [y+r y-r], 'Color', 'Red', 'LineWidth', 3);

```

```

%%%%%%%%%%%%% сохранение%%%%%%%%%%%%%
filename= sprintf( '%s\\%s_result_horizon.png', out_pic_folder, im_prefix );
print('-noui', '-r0', '-dpng', filename);

```

```

vp = vp / (768/picsize);
save(vpStr, 'vp', '-ascii');
save(vpAssStr, 'vpAss', '-ascii');

```

## ПРИЛОЖЕНИЕ В

### Текст функций обработки изображения

#### angle\_between\_two\_zeniths.m

```
function ang = angle_between_two_zeniths(zen1_x, zen1_y, zen2_x, zen2_y, pp_x, pp_y)
```

```
v1 = [zen1_x - pp_x, abs(zen1_y - pp_y)];  
v2 = [zen2_x - pp_x, abs(zen2_y - pp_y)];
```

```
v1_length = sqrt(sum(v1.*v1));  
v2_length = sqrt(sum(v2.*v2));
```

```
prod = sum(v1.*v2);
```

```
ang = acos( prod / (v1_length * v2_length) );
```

#### deleteEqualPoints.m

```
function points_new = deleteEqualPoints (points, eps)
```

```
npoints = size (points, 1);
```

```
X = points(:, 1);  
Y = points(:, 2);
```

```
points3d = [2*X ./ (1+X.*X+Y.*Y), 2*Y ./ (1+X.*X+Y.*Y), (-1+X.*X+Y.*Y) ./ (1+X.*X+Y.*Y)];  
points2d = points;  
points = points3d;
```

```
unique (1:npoints) = true;
```

```
for i = 1: npoints
```

```
    if (unique (i))
```

```
        for j = i+1 : npoints
```

```
            if (unique(j)) && abs (points (i, 1) - points (j, 1)) < eps ...
```

```
                && (abs (points (i, 2) - points (j, 2)) < eps...
```

```
                && (abs (points (i, 3) - points (j, 3)) < eps) )
```

```
                unique (j) = false;
```

```
            end
```

```
        end
```

```
    end
```

```
end
```

```
points_new = points2d(unique, 1:2);
```

#### distanceBetweenLines.m

```
function d = distanceBetweenLines( l, hor, width )
```

```
%calculates distances between horizon lines
```

```
dist = zeros (1, 4);
```

```
x = 1;
```

```
p1 = [x; (-l(1)*x - l(3))/ l(2); 1];
```

					ЮУрГУ-120401.2017.071. ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		120



```
x = width;
p2 = [x; (-l(1)*x - l(3))/ l(2); 1];
```

```
dist(1) = abs(dot(hor, p1));
dist(2) = abs(dot(hor, p2));
```

```
x = 1;
p1 = [x; (-hor(1)*x - hor(3))/ hor(2); 1];
x = width;
p2 = [x; (-hor(1)*x - hor(3))/ hor(2); 1];
```

```
dist(3) = abs(dot(l, p1));
dist(4) = abs(dot(l, p2));
```

```
d = max (dist(:));
end
```

### drawHorizont\_orthogonal.m

```
function hor = drawHorizont_orthogonal(points_hor, points_ver, weights_hor, width, height, hor_color, pp_imcoords)
```

```
% default horizon
angle_horizon = 0;
```

```
horA = tan(angle_horizon);
horC = height/2 - horA*width/2;
```

```
hor = [horA; -1; horC];
hor = hor / norm(hor(1:2));
```

```
hold on;
```

```
if (~isempty(points_ver) && ~isempty(points_hor))
```

```
    points_hor_x = points_hor(:, 1);
    finite_pts = find(abs(points_hor_x - width/2) < width * 7);
```

```
    if (size(points_hor, 1) > length(finite_pts) && (size(points_hor, 1) ~= 0))
```

```
        points2(1:length (finite_pts), 1:2) = points_hor(finite_pts, 1:2);
        points_hor = points2;
        weights_hor = weights_hor(finite_pts);
```

```
    end;
```

```
if (size (points_hor, 1) >= 1)
```

```
    projections = zeros(size(points_hor, 1), 2);
```

					ЮУрГУ-120401.2017.071. ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		121

```

Ax = points_ver(1, 1); % x-coordinate of zenith
Ay = points_ver(1, 2); % y-coordinate of zenith

Bx = pp_imcoords(1); % x-coordinate of principal point
By = pp_imcoords(2); % y-coordinate of principal point

for iHorPoint = 1:size(points_hor, 1)

    Cx = points_hor(iHorPoint, 1); % x-coordinate of current vanishing point
    Cy = points_hor(iHorPoint, 2); % y-coordinate of current vanishing point

    LC = sqrt( (Bx-Ax)*(Bx-Ax) + (By-Ay)*(By-Ay) );
    rC = ( (Ay-Cy)*(Ay-By)-(Ax-Cx)*(Bx-Ax) ) / (LC*LC);

    PCx = Ax + rC*(Bx-Ax); % x-coordinate of projecton of vanishing point on AB
    PCy = Ay + rC*(By-Ay); % y-coordinate of projecton of vanishing point on AB

    projections(iHorPoint, 1) = PCx;
    projections(iHorPoint, 2) = PCy;

end;

mean_projection_x = sum(projections(:, 1).*weights_hor) / sum(weights_hor);
mean_projection_y = sum(projections(:, 2).*weights_hor) / sum(weights_hor);

x = mean_projection_x;
y = mean_projection_y;

angle_zenith_pp = atan2(Ay-By, Ax-Bx);
angle_horizon = angle_zenith_pp + pi/2;

horA = tan(angle_horizon);
horC = mean_projection_y - horA*mean_projection_x;

hor = [horA; -1; horC];
hor = hor / norm(hor(1:2));

end
end;

x = 1;
p1 = [x; (-hor(1)*x - hor(3)) / hor(2); 1];
x = width;
p2 = [x; (-hor(1)*x - hor(3)) / hor(2); 1];

hold on;
plot([p1(1) p2(1)], [p1(2) p2(2)], 'Color', hor_color, 'LineWidth', 7);

calc_candidates_segments_lsd.m

function [line_segments] = calc_candidates_segments_lsd(pic, display)

```

```
out_segм = [];  
pic_init = pic;  
  
%find line segments on 3 scales of the input image  
for iter = 1:3  
  
    pic = imresize(pic_init, 1/iter);  
  
    BW_im = rgb2gray(pic);  
    [out_segм_2] = mxLSD(uint8(BW_im));  
  
    %eliminate the difference between opencv and matlab coordinate systems  
    out_segм_2(:, :) = (out_segм_2(:, :) + 1) * iter - 1;  
    eq = zeros (size(out_segм, 1), 1);  
  
    if (display)  
        sfigure (1);  
        imshow (pic_init);  
        hold on;  
        Nedge = size (out_segм_2, 1);  
        colourmp = hsv(Nedge*5); % HSV colour map with Nedge entries  
        colourmp = colourmp(randperm(Nedge*5,:)); % Random permutation  
    end  
  
    if (size (out_segм) ~= 0)  
        for I = 1:size (out_segм_2, 1)  
            eq (:) = equal_segments (out_segм(:, :), out_segм_2(I, :));  
            if sum (eq(:) == 0) % if this segment isn't already presented  
                out_segм = [out_segм; out_segм_2(I, :)];  
                if (display)  
                    plot([out_segм_2(I,1) + 1 out_segм_2(I,3) + 1], [out_segм_2(I,2) + 1, out_segм_2(I,4) + 1], 'Color',  
                        colourmp(I,:), 'LineWidth', 1 );  
                end  
            end  
        end  
    else  
        line_segments = out_segм_2;  
        Nedge = size (line_segments, 1);  
        if (display)  
            for I = 1:Nedge  
                plot([line_segments(I,1) + 1 line_segments(I,3) + 1], [line_segments(I,2) + 1, line_segments(I,4) + 1], 'Color',  
                    colourmp(I,:), 'LineWidth', 1 );  
            end  
        end  
        out_segм = out_segм_2;  
    end  
end  
  
line_segments = zeros(size(out_segм, 1), 6);  
for iLine = 1:size(out_segм, 1)  
    x1 = out_segм (iLine, 1);  
    y1 = out_segм (iLine, 2);  
    x2 = out_segм (iLine, 3);  
    y2 = out_segм (iLine, 4);  
    if (abs(y2-y1) < 1e-6)  
        theta = pi / 2.0;
```

```

else
    theta = atan ((x2-x1)/(y1-y2));
end
r = sqrt((x1-x2)^2 + (y1-y2)^2);
line_segments(iLine, :) = [x1 x2 y1 y2 theta r];
end;

end

function [equal] = equal_segments (segm1, segm2)
    equal = true;
    for i = 1:4
        equal = equal && (abs (segm1 (i) - segm2(i)) < 1 );
    end

end

```

### calc\_candidates\_tard.m

```

function [vp_imcoords, nPtsHor] = calc_candidates_tard(im, TardifsCount, display)

displayPoints = false;

maxDistBetweenDistinctVP = 10.0;

addpath ('../Tardif');
addpath(genpath('../CPP/JLinkage'));
addpath '../CPP/lineSegDetect/'

ARGS = FACADE_ARGS_default();
const = FACADE_const();
ARGS.mKinv = [];

%arguments for Vanishing point detection
ARGS.plot = 0;
ARGS.savePlot = false;

ARGS.manhattanVP = true;
ARGS.manhattanVP = false;
ARGS.REF_remNonManhantan = true;
ARGS.ALL_calibrated = false;
ARGS.ERROR = const.ERROR_DIST;
%-----

im = rgb2gray(im);

ARGS.imgS = max(size(im));

width = size(im, 2);
height = size(im, 1);

[vsEdges, ARGS.imE] = FACADE_getEdgelets2(im, ARGS);

```

```

ARGS.JL_ALGO=2;

tic;

[firstVP,vClass] = FACADE_getVP_JLinkage(vsEdges, im,ARGS);

vsVP = firstVP;
for iTard = 1:TardifsCount-1
    [tmpVP, vClass] = FACADE_getVP_JLinkage(vsEdges, im,ARGS);
    vsVP = [vsVP tmpVP];
    if (length(vsVP) > 150)
        break;
    end;
    vbOutliers = [vClass == 0];

    if (display)
        hold on;

        sfigure(1);
        imshow (im);
        classClr = FACADE_plotSegmentation(im, vsEdges, vClass, [], vbOutliers, 0);
    end
    if (displayPoints)
        for i = 1:size(tmpVP, 2)
            hold on;
            x = tmpVP(i).VP(1)/tmpVP(i).VP(3);
            y = tmpVP(i).VP(2)/tmpVP(i).VP(3);
            x = x * ARGS.imgS;
            y = - y * ARGS.imgS;
            plot ([x size(im, 2)/2], [y size(im, 1)/2], 'Color', classClr(i, :), 'LineWidth', 5);
        end
    end
end;

toc;

points = zeros (size(vsVP, 2), 2);
point = zeros (1, 3);

for i = 1:size(vsVP, 2)
    point(1:3) = vsVP(i).VP;
    points(i, 1) = point(1) / point(3);
    points(i, 2) = point(2) / point(3);
end

points = deleteEqualPoints (points, maxDistBetweenDistinctVP/ARGS.imgS);

points_imcoords = zeros (size(points, 2), 2);

for i = 1:size(points, 1)

```

```

points_imcoords (i, 1) = points (i, 1) * ARGS.imgS;
points_imcoords (i, 2) = - points (i, 2) * ARGS.imgS;
end

[hor_points_imcoords, zen_points_imcoords] = splitPointsThrAngleAndY (points_imcoords, width, height);

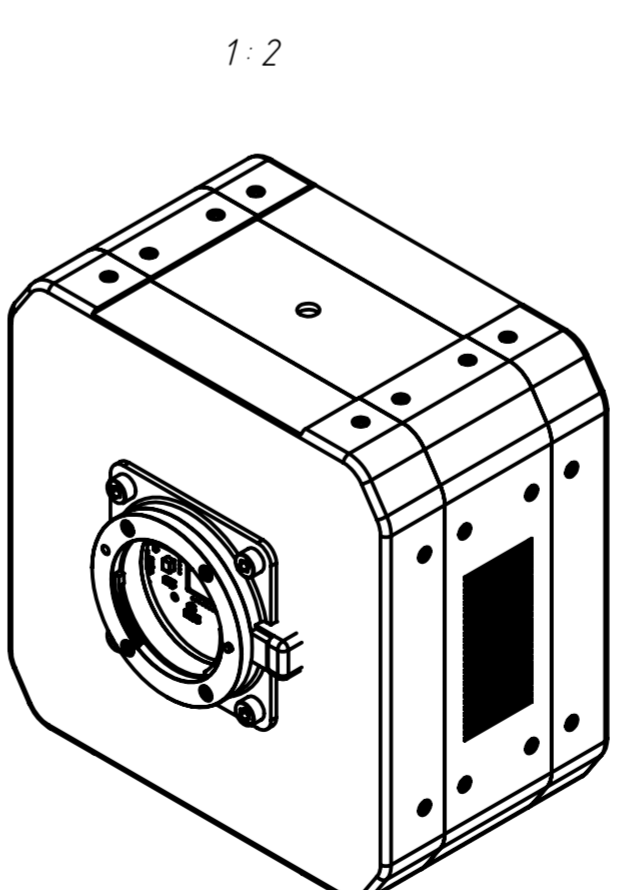
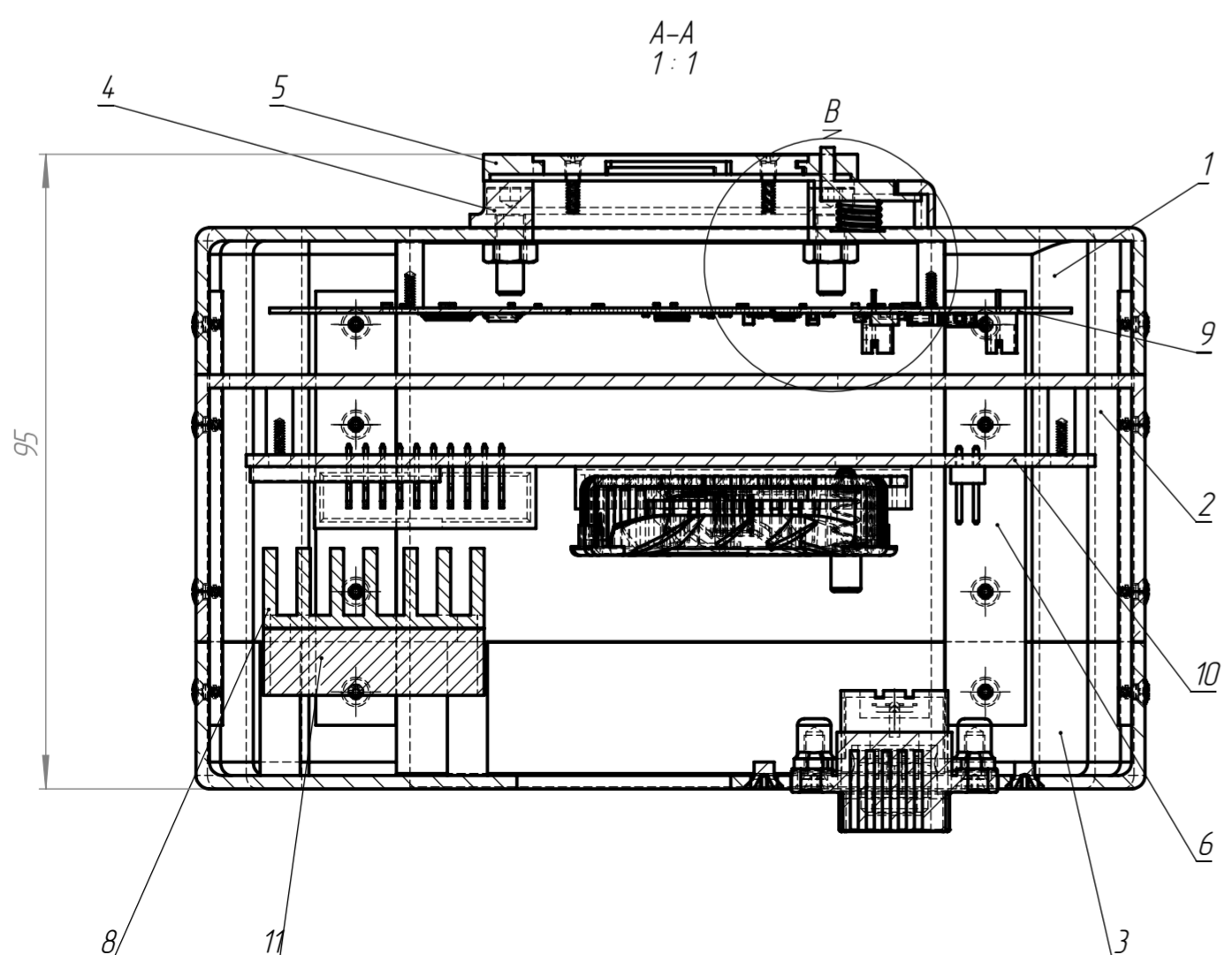
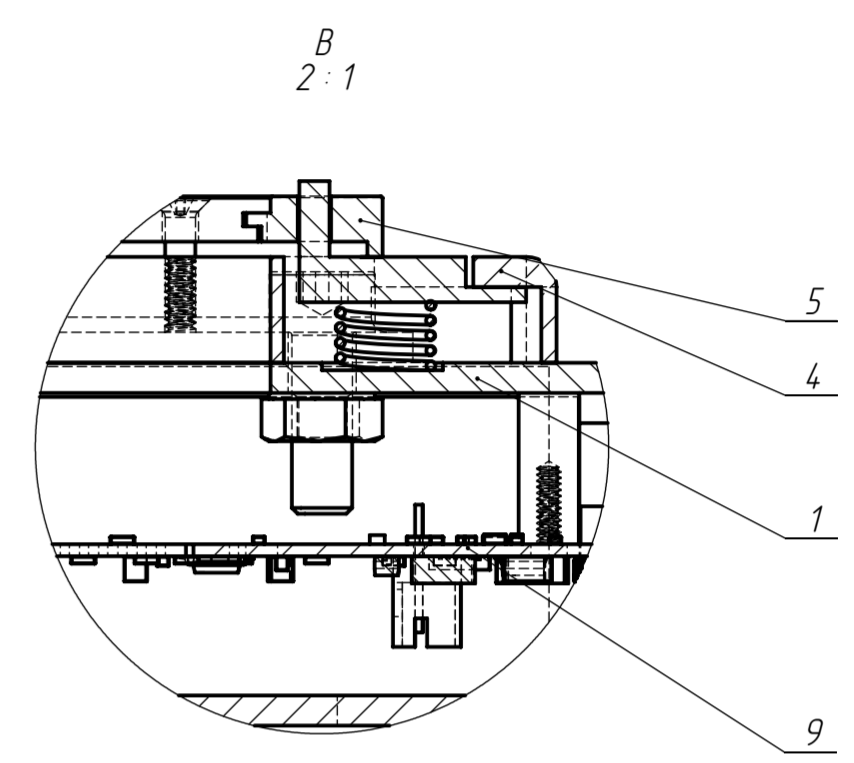
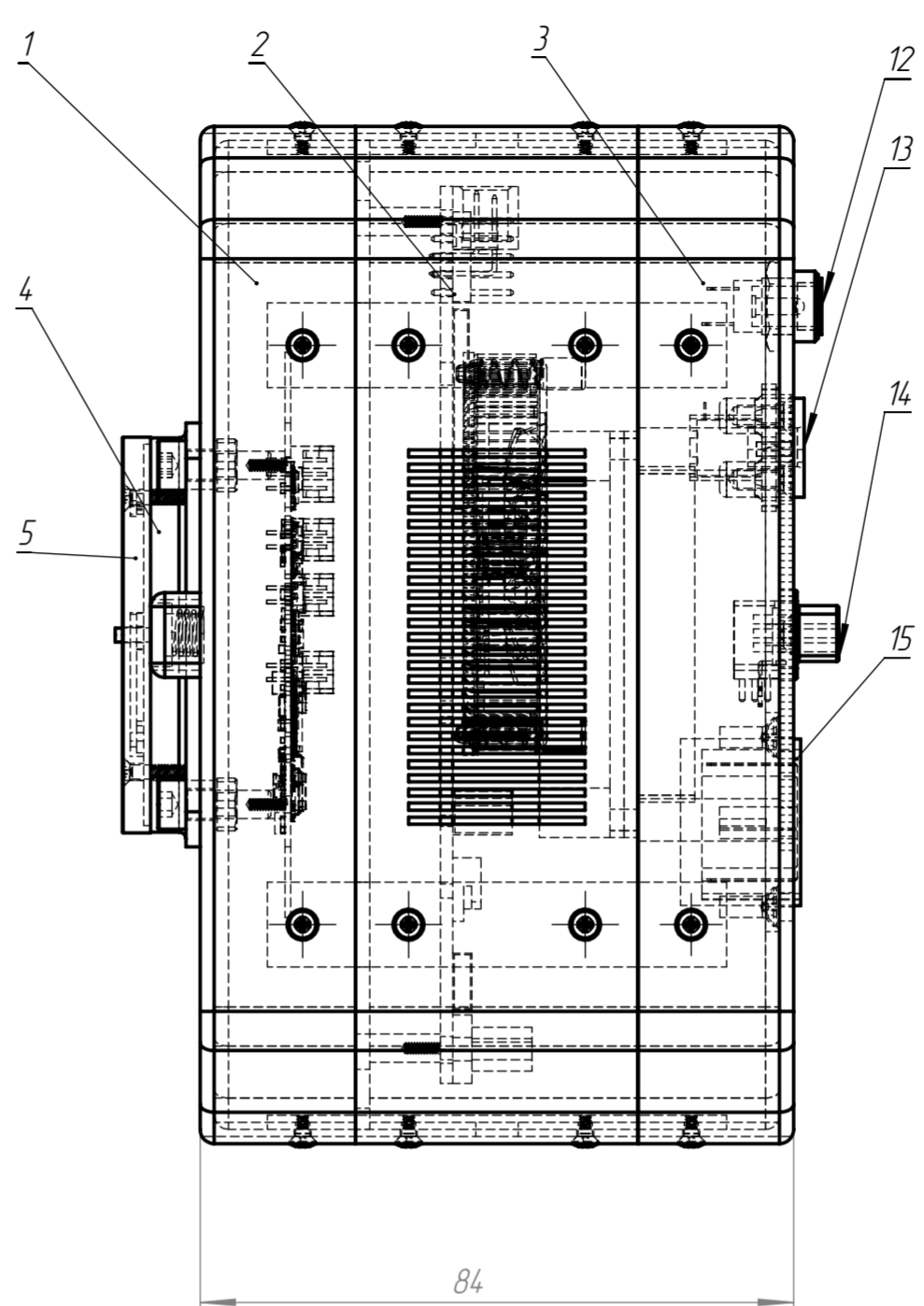
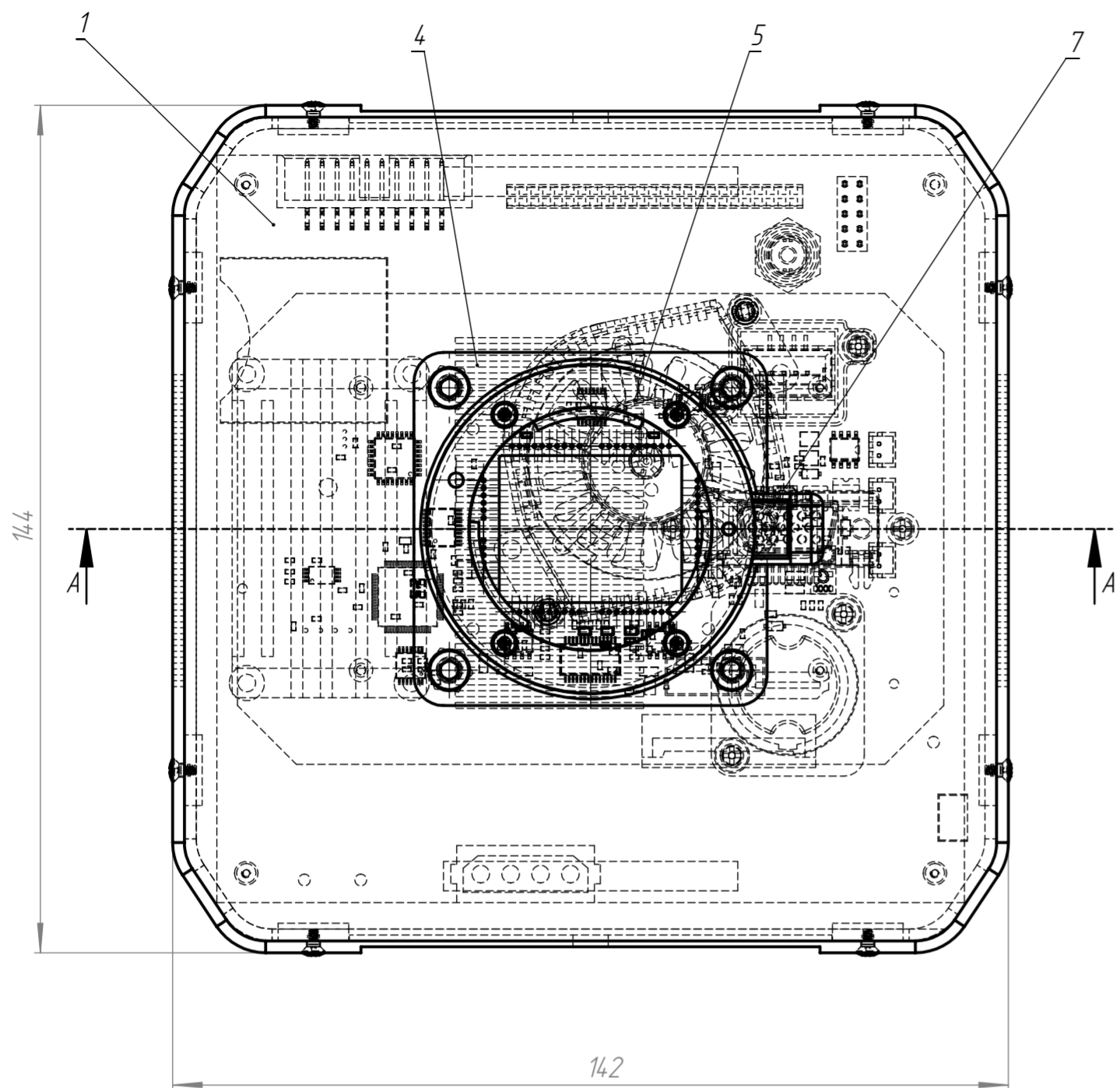
vp_imcoords = zeros (size(hor_points_imcoords, 1) + size(zen_points_imcoords, 1), 2);
vp_imcoords (1:size(hor_points_imcoords, 1), :) = hor_points_imcoords (1:size(hor_points_imcoords, 1), :);
vp_imcoords (size(hor_points_imcoords, 1)+1:size(hor_points_imcoords, 1)+size(zen_points_imcoords, 1), :) =
zen_points_imcoords (1:size(zen_points_imcoords, 1), :);

nPtsHor = size(hor_points_imcoords, 1);

```

					ЮУрГУ-120401.2017.071. ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		126

Первичное применение		Формат	Зона	Позиция	Обозначение	Наименование	Кол.	Примечание		
Справочный №						<u>Документация</u>				
		A2			12.04.01.2017.01.00 СБ	Сборочный чертеж				
							<u>Детали</u>			
				1			Корпус 1			
				2			Корпус 2			
				3			Корпус 3			
				4			Рабочий отрезок			
				5			Крепление объектива			
				6			Направляющая			
				7			Кнопка			
				8			Теплоотвод			
				9			Сенсорный модуль			
				10			Процессорный модуль			
				11			Модуль питания			
				12			Разъем питания			
		13			Разъем USB					
		14			Разъем RS232					
		15			Разъем HDMI					
Подпись и дата					12.04.01.2017.071.01.00					
		Изм.	Лист	№ документа					Подпись	Дата
Инв. № подл.	Разраб.	Грачев Е.А.			Оптическая система			Лит.	Лист	Листов
	Проб.	Кацай Д.А.								1
	Н.контр.	Лысова А.А.								
	Утв.									
Взам. инв. №										
Инв. № дубл.										
Подпись и дата										
ЮУрГУ Кафедра ИНИТ										



Перв. примен.  
Справ. №  
Подп. и дата  
Инв. № дфл.  
Взам. инв. №  
Подп. и дата  
Инв. № подл.

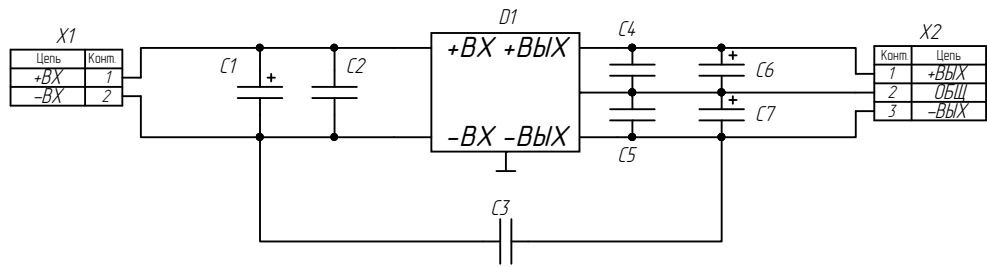
					12.04.01.2017.071.01.00 СБ			
Изм.	Лист	№ докум.	Подп.	Дата	Оптическая система Сборочный чертёж	Лит.	Масса	Масштаб
Разраб.	Грачев Е.А.							1:1
Проб.	Кацай Д.А.				Лист	Листов 1		
Т. контр.								
Н. контр.	Льсова А.А.							ЮрГУ Кафедра ИНИТ
Утв.								Копировал Формат А2



12.04.01.2017.071.01.11

Перв. примен.

Справ. №



Подп. и дата

Инд. № дубл.

Взам. инд. №

Подп. и дата

Инд. № подл.

12.04.01.2017.071.01.1133

Модуль питания

СХЕМА ЭЛЕКТРИЧЕСКАЯ  
ПРИНЦИПИАЛЬНАЯ

Лит.	Масштаб	Масса
Лист		Листов 1

Изм.	Лист	№ докум.	Подп.	Дата
Разраб.		Грачев Е.А.		
Пров.		Кацай Д.А.		
Т.контр.				
И.контр.		Лысова А.А.		
Утв.				

ЮУрГУ Кафедра ИНИТ

Перв. примен.	Поз. обозначение	Наименование	Кол.	Примечание
		<b>Конденсаторы</b>		
	C1	Конденсатор танталовый выводной 47 мкФ ± 10% 20В K53-18	1	
Справ. №	C2...C5	GRM188R71E105KA12D Конденсатор керамический 1 мкФ ± 10% 25В X7R 0603 Murata	4	
	C6,C7	Конденсатор танталовый выводной 47 мкФ ± 10% 20В K53-18	2	
	D1	Модуль питания МДМ7,5 7,5Вт	1	
	X1	Вилка JST-PH-2.0-2 2 контакта, угловая, шаг 2 мм	1	
	X2	Вилка JST-PH-2.0-3 3 контакта, угловая, шаг 2 мм	1	
Подп. и дата				
Инв. № дубл.				
Взам. инв. №				
Подп. и дата				
Инв. № подл.	12.04.01.2017.071.01.11ПЭЗ			
	Изм.	Лист	№ докум.	Подп.
Инв. № подл.	Разраб.	Грачев Е.А.		
	Пров.	Кацай Д.А.		
	Т.контр.			
	Н.контр.	Лысова А.А.		
	Чтв.			
<b>Модуль питания</b>			Лит.	Лист
<b>Перечень элементов</b>				Листов
				1
			ЮУрГУ Кафедра ИНИТ	

12.04.01.2017.071.01.09

Перепроверен

Справ. №

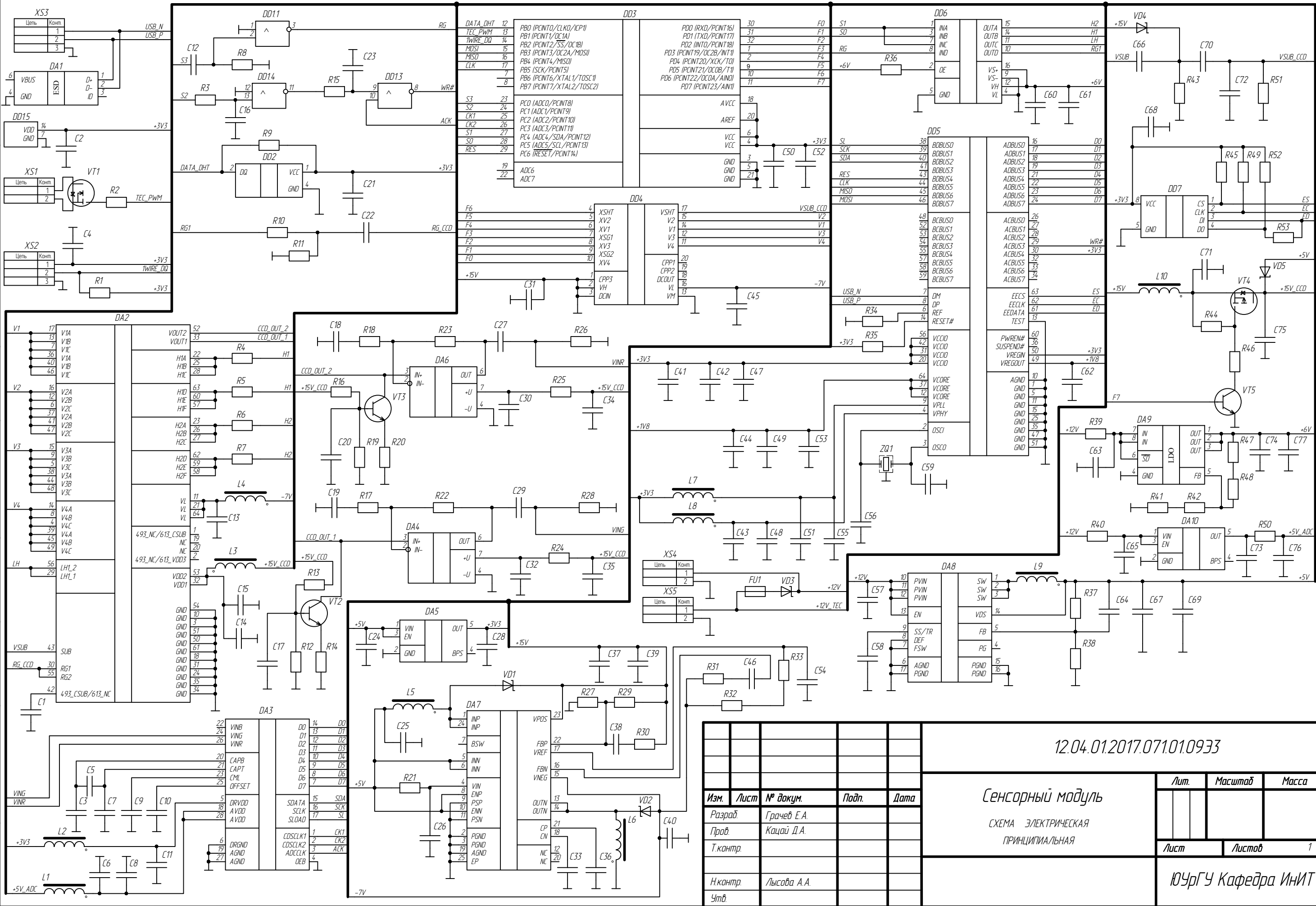
Подп. и дата

Инд. № дубл.

Взам. инв. №

Подп. и дата

Инд. № подл.



12.04.01.2017.071.01.0933

Сенсорный модуль

СХЕМА ЭЛЕКТРИЧЕСКАЯ  
ПРИНЦИПАЛЬНАЯ

Изм.	Лист	№ докум.	Подп.	Дата
Разраб		Грачев Е.А.		
Пров		Кацай Д.А.		
Т.контр.				
Н.контр.		Лысова А.А.		
Умб.				

Лист	Масштаб	Масса
Лист	Листов	1

ЮрГУ Кафедра ИНИТ

Поз. обозначение	Наименование	Кол.	Примечание						
<b>Конденсаторы</b>									
C1	GRM21BR71H105K Конденсатор керамический 1 мкФ ± 10% 50В X7R 0805 Murata	1							
C2...C4	GRM188R71C104KA01D Конденсатор керамический 100 нФ ± 10% 25В X7R 0603 Murata	3							
C5	GRM188R60J106ME47D Конденсатор керамический 10 мкФ ± 20% 6,3В X5R 0603 Murata	1							
C6,C7	GRM188R71C104KA01D Конденсатор керамический 100 нФ ± 10% 25В X7R 0603 Murata	2							
C8	GRM188F51A475ZE20D Конденсатор керамический 4,7 мкФ ± 10% 10В Y5V 0603 Murata	1							
C9	GRM188R71C104KA01D Конденсатор керамический 100 нФ ± 10% 25В X7R 0603 Murata	1							
C10	GRM188R71E105KA12D Конденсатор керамический 1 мкФ ± 10% 25В X7R 0603 Murata	1							
C11	GRM188R71C104KA01D Конденсатор керамический 100 нФ ± 10% 25В X7R 0603 Murata	1							
C12	GRM1885C1H300J Конденсатор керамический 30 нФ ± 5% 50В NPO 0603 Murata	1							
C13	GRM31CR61C226ME15L Конденсатор керамический 22 мкФ ± 20% 16В X5R 1206 Murata	1							
C14,C15	GRM32DR71E106K Конденсатор керамический 10 мкФ ± 10% 25В X7R 1210 Murata	2							
C16	GRM1885C1H300J Конденсатор керамический 30 нФ ± 5% 50В NPO 0603 Murata	1							
12.04.01.2017.071.01.09ПЭЗ									
<b>Сенсорный модуль</b>									
Перечень элементов									
Изм.		Лист		N докум.		Подп.		Дата	
Разраб.		Грачев Е.А.							
Пров.		Кацай Д.А.							
Т.контр.									
Н.контр.		Лысова А.А.							
Чтв.									
Лит.		Лист		Листов					
		1		9					
ЮУрГУ Кафедра ИНИТ									

Поз. обозначение	Наименование	Кол.	Примечание
C17...C20	GRM188R71E105KA12D Конденсатор керамический	4	
	1 мкФ ± 10% 25В X7R 0603 Murata		
C21,C22	GRM188R71C104KA01D Конденсатор керамический	2	
	100 нФ ± 10% 25В X7R 0603 Murata		
C23	GRM1885C1H680J Конденсатор керамический	1	
	68 нФ ± 5% 50В NPO 0603 Murata		
C24	GRM188F51A475ZE20D Конденсатор керамический	1	
	4,7 мкФ ± 10% 10В Y5V 0603 Murata		
C25	GRM31CR61C226ME15L Конденсатор керамический	1	
	22 мкФ ± 20% 16В X5R 1206 Murata		
C26,C27	GRM188R71C104KA01D Конденсатор керамический	2	
	100 нФ ± 10% 25В X7R 0603 Murata		
C28	GRM188F51A475ZE20D Конденсатор керамический	1	
	4,7 мкФ ± 10% 10В Y5V 0603 Murata		
C29	GRM188R71C104KA01D Конденсатор керамический	1	
	100 нФ ± 10% 25В X7R 0603 Murata		
C30	GRM188F51A475ZE20D Конденсатор керамический	1	
	4,7 мкФ ± 10% 10В Y5V 0603 Murata		
C31	GRM21BR71H105K Конденсатор керамический	1	
	1 мкФ ± 10% 50В X7R 0805 Murata		
C32	GRM31CR71H475KA12L Конденсатор керамический	1	
	4,7 мкФ ± 10% 50В X7R 1206 Murata		
C33	GRM188F51A475ZE20D Конденсатор керамический	1	
	4,7 мкФ ± 10% 10В Y5V 0603 Murata		
C34,C35	GRM31CR71H475KA12L Конденсатор керамический	2	
	4,7 мкФ ± 10% 50В X7R 1206 Murata		
C36	GRM188R61H103KA01D Конденсатор керамический	1	
	10 нФ ± 10% 50В X5R 0603 Murata		
C37	GRM32DR71E106K Конденсатор керамический	1	
	10 мкФ ± 10% 25В X7R 1210 Murata		
C38	GRM1885C1H100J Конденсатор керамический	1	
	10 нФ ± 5% 50В NPO 0603 Murata		

Инв. № подл.	Подп. и дата
	Инв. № дубл.
Взам. инв. №	Подп. и дата
	Инв. № подл.

Изм.	Лист	№ докум.	Подп.	Дата
------	------	----------	-------	------

12.04.01.2017.071.01.09П33

Поз. обозначение	Наименование	Кол.	Примечание
C39	GRM32DR71E106K Конденсатор керамический	1	
	10 мкФ ± 10% 25В X7R 1210 Murata		
C40	GRM31CR61C226ME15L Конденсатор керамический	1	
	22 мкФ ± 20% 16В X5R 1206 Murata		
C41,C42	GRM188R71C104KA01D Конденсатор керамический	2	
	100 нФ ± 10% 25В X7R 0603 Murata		
C43	GRM188F51A475ZE20D Конденсатор керамический	1	
	4,7 мкФ ± 10% 10В Y5V 0603 Murata		
C44,C45	GRM188R71C104KA01D Конденсатор керамический	2	
	100 нФ ± 10% 25В X7R 0603 Murata		
C46	GRM1885C1H100J Конденсатор керамический	1	
	10 нФ ± 5% 50В NPO 0603 Murata		
C47...C50	GRM188R71C104KA01D Конденсатор керамический	4	
	100 нФ ± 10% 25В X7R 0603 Murata		
C51,C52	GRM188F51A475ZE20D Конденсатор керамический	2	
	4,7 мкФ ± 10% 10В Y5V 0603 Murata		
C53...C55	GRM188R71C104KA01D Конденсатор керамический	3	
	100 нФ ± 10% 25В X7R 0603 Murata		
C56	GRM1885C1H220J Конденсатор керамический	1	
	22 нФ ± 5% 50В NPO 0603 Murata		
C57	GRM31CR71H475KA12L Конденсатор керамический	1	
	4,7 мкФ ± 10% 50В X7R 1206 Murata		
C58	GRM1885C1H222JA01D Конденсатор керамический	1	
	2,2нФ ± 5% 50В COG 0603 Murata		
C59	GRM1885C1H220J Конденсатор керамический	1	
	22 нФ ± 5% 50В NPO 0603 Murata		
C60,C61	GRM188R71C104KA01D Конденсатор керамический	2	
	100 нФ ± 10% 25В X7R 0603 Murata		
C62	GRM188F51A475ZE20D Конденсатор керамический	1	
	4,7 мкФ ± 10% 10В Y5V 0603 Murata		
C63	GRM31CR71H475KA12L Конденсатор керамический	1	
	4,7 мкФ ± 10% 50В X7R 1206 Murata		

Подп. и дата  
Инв. № дудл.  
Взам. инв. №  
Подп. и дата  
Инв. № подл.

Изм.	Лист	№ докум.	Подп.	Дата
------	------	----------	-------	------

12.04.01.2017.071.01.09П33

Лист  
3

Поз. обозначение	Наименование	Кол.	Примечание
C64	GRM1885C1H220J Конденсатор керамический	1	
	22 нФ ± 5% 50В NPO 0603 Murata		
C65	GRM31CR71H475KA12L Конденсатор керамический	1	
	4,7 мкФ ± 10% 50В X7R 1206 Murata		
C66	GRM21BR71H105K Конденсатор керамический	1	
	1 мкФ ± 10% 50В X7R 0805 Murata		
C67	GRM31CR61C226ME15L Конденсатор керамический	1	
	22 мкФ ± 20% 16В X5R 1206 Murata		
C68	GRM188R71C104KA01D Конденсатор керамический	1	
	100 нФ ± 10% 25В X7R 0603 Murata		
C69	GRM31CR61C226ME15L Конденсатор керамический	1	
	22 мкФ ± 20% 16В X5R 1206 Murata		
C70	GRM21BR71H105K Конденсатор керамический	1	
	1 мкФ ± 10% 50В X7R 0805 Murata		
C71	GRM31CR71H475KA12L Конденсатор керамический	1	
	4,7 мкФ ± 10% 50В X7R 1206 Murata		
C72	GRM1885C2A102JA01D Конденсатор керамический	1	
	1нФ ± 5% 100В COG 0603 Murata		
C73	GRM188F51A475ZE20D Конденсатор керамический	1	
	4,7 мкФ ± 10% 10В Y5V 0603 Murata		
C74	GRM31CR61C226ME15L Конденсатор керамический	1	
	22 мкФ ± 20% 16В X5R 1206 Murata		
C75	GRM21BR71H105K Конденсатор керамический	1	
	1 мкФ ± 10% 50В X7R 0805 Murata		
C76	GRM31CR61C226ME15L Конденсатор керамический	1	
	22 мкФ ± 20% 16В X5R 1206 Murata		
C77	GRM188R71C104KA01D Конденсатор керамический	1	
	100 нФ ± 10% 25В X7R 0603 Murata		

Инв. № подл.	Подп. и дата
	Инв. № дудл.
Взам. инв. №	Подп. и дата
	Подп. и дата
Инв. № подл.	Подп. и дата
	Подп. и дата

Изм.	Лист	№ докум.	Подп.	Дата
------	------	----------	-------	------

12.04.01.2017.071.01.09П33

Поз. обозначение	Наименование	Кол.	Примечание
	<b>Микросхемы</b>		
DA1	Микросхема TPD4S012DRYR 4-CHANNEL ESD SOLUTION FOR USB-HS/USB OTG/USB CHARGER INTERFACE, Texas Instruments	1	
DA2	Микросхема ICX493AQ CCD Image Sensor, SONY	1	
DA3	Микросхема AD9826KRS Сигнальный процессор 5,25В, Analog Devices	1	
DA4	Микросхема AD811JRZ Высокоэффективный, широкополосный видеоусилитель с отрицательной обратной связью, Analog Devices	1	
DA5	Микросхема LP2985-33DBVR Малощумящий стабилизатор напряжения с отключением, Texas Instruments	1	
DA6	Микросхема AD811JRZ широкополосный видеоусилитель с отрицательной обратной связью, Analog Devices	1	
DA7	Микросхема TPS65130RGE DC to DC Converter and Switching Regulator Chip Texas Instruments	1	
DA8	Микросхема TLV62130RGT 3-17V 3A Step-Down Converter, Texas Instruments	1	
DA9	Микросхема ADP3336ARMZ Стабилизатор с малым падением напряжения, регулируемый, Analog Devices	1	
DA10	Микросхема LP2985-50DBVR Регулятор напряжения 5В, 150мА Texas Instruments	1	
DD1	Микросхема 74HC86PWR Логический элемент исключающее ИЛИ, семейство HC, 2 входа, 4 элемента, 2В до 6В, Texas Instruments	1	
DD2	Микросхема DHT22 Датчик температуры и влажности, Waveshare Electronics	1	

Подп. и дата  
Инв. № дудл.  
Взам. инв. №  
Подп. и дата  
Инв. № подл.

Изм.	Лист	№ докум.	Подп.	Дата
------	------	----------	-------	------

12.04.01.2017.071.01.09П33



Поз. обозначение	Наименование	Кол.	Примечание
DD3	Микросхема ATMEGA328P-AU	1	
	Микроконтроллер 8-Бит, 20МГц, 32КБ Atmel		
DD4	Микросхема CXD1267AN CCD Vertical ClockDriver	1	
	SONY		
DD5	Микросхема FT2232H Преобразователь	1	
	USB-UART/FIFO Future Technology Devices		
DD6	Микросхема EL7457CUZ	1	
	ИС драйвера МОП-транзистора, 4,5В-18В Intersil		
DD7	Микросхема 93C46	1	
	Энергонезависимая память, SPI, 1КБит, 2МГц Atmel		
FU1	Предохранитель 2016L030 Resettable POLYFUSE 20A	1	
	Littelfuse		
	<i>Катушки индуктивности</i>		
L1,L2	Индуктивность BLM18PG121SN1D	2	
	Дроссель подавления ЭМП, 0603 Murata		
L3	Индуктивность BLM21PG221SN10	1	
	Дроссель подавления ЭМП, 0603 Murata		
L4	Индуктивность BLM18PG121SN1D	1	
	Дроссель подавления ЭМП, 0603 Murata		
L5,L6	Индуктивность SRP4020TA	2	
	Силовой индуктор, 10 мкГ ± 10%, 1.8 А BOURNS		
L7,L8	Индуктивность BLM18PG121SN1D	2	
	Дроссель подавления ЭМП, 0603 Murata		
L9	Индуктивность XFL4020-152	1	
	Силовой индуктор 1,5 мкГ ± 20%, 4,4 А, 59 МГц		
	COILCRAFT		
L10	Индуктивность BLM18PG121SN1D	1	
	Дроссель подавления ЭМП, 0603 Murata		

Инв. № подл.	Подп. и дата
	Инв. № дудл.
Взам. инв. №	
Подп. и дата	
Инв. № подл.	

Изм.	Лист	№ докум.	Подп.	Дата
------	------	----------	-------	------

12.04.01.2017.071.01.09П33

Поз. обозначение	Наименование	Кол.	Примечание
	<b>Резисторы</b>		
R1	Резистор 4,7 кОм ± 5% 0,063Вт 0603	1	
R2	Резистор 22 Ом ± 5% 0,063Вт 0603	1	
R3	Резистор 560 Ом ± 5% 0,063Вт 0603	1	
R4...R7	Резистор 3 Ом ± 5% 0,063Вт 0603	4	
R8	Резистор 1 кОм ± 5% 0,063Вт 0603	1	
R9	Резистор 4,7 кОм ± 5% 0,063Вт 0603	1	
R10	Резистор 51 Ом ± 5% 0,063Вт 0603	1	
R11	Резистор 200 Ом ± 5% 0,063Вт 0603	1	
R12	Резистор 51 кОм ± 5% 0,063Вт 0603	1	
R13	Резистор 100 кОм ± 5% 0,063Вт 0603	1	
R14	Резистор 10 кОм ± 5% 0,063Вт 0603	1	
R15	Резистор 330 Ом ± 5% 0,063Вт 0603	1	
R16	Резистор 100 кОм ± 5% 0,063Вт 0603	1	
R17,R18	Резистор 1 кОм ± 5% 0,063Вт 0603	2	
R19	Резистор 51 кОм ± 5% 0,063Вт 0603	1	
R20	Резистор 10 кОм ± 5% 0,063Вт 0603	1	
R21	Резистор 300 Ом ± 5% 0,063Вт 0603	1	
R22,R23	Резистор 1,5 кОм ± 5% 0,063Вт 0603	2	
R24,R25	Резистор 5,1 Ом ± 5% 0,063Вт 0603	2	
R26	Резистор 1 МОм ± 5% 0,063Вт 0603	1	
R27	Резистор 68 кОм ± 5% 0,063Вт 0603	1	
R28	Резистор 1 МОм ± 5% 0,063Вт 0603	1	
R29	Резистор 750 кОм ± 5% 0,063Вт 0603	1	
R30	Резистор 0 Ом ± 5% 0,063Вт 0603	1	
R31	Резистор 10 кОм ± 5% 0,063Вт 0603	1	
R32	Резистор 1 МОм ± 5% 0,063Вт 0603	1	
R33	Резистор 160 кОм ± 5% 0,063Вт 0603	1	
R34	Резистор 12 кОм ± 5% 0,063Вт 0603	1	
R35,R36	Резистор 10 кОм ± 5% 0,063Вт 0603	2	
R37	Резистор 56 кОм ± 5% 0,063Вт 0603	1	
R38	Резистор 10 кОм ± 5% 0,063Вт 0603	1	
Инв. № подл.			
Взам. инв. №			
Инв. № дубл.			
Подп. и дата			
Подп. и дата			
Инв. № подл.			
Изм.	Лист	№ докум.	Подп.
			Дата

12.04.01.2017.071.01.09П33

Лист

7

Поз. обозначение	Наименование	Кол.	Примечание
R39,R40	Резистор 5,1 Ом ± 5% 0,063Вт 0603	2	
R41	Резистор 200 Ом ± 5% 0,063Вт 0603	1	
R42	Резистор 62 кОм ± 5% 0,063Вт 0603	1	
R43	Резистор 100 кОм ± 5% 0,063Вт 0603	1	
R44,R45	Резистор 10 кОм ± 5% 0,063Вт 0603	2	
R46	Резистор 22 Ом ± 5% 0,063Вт 0603	1	
R47	Резистор 15 кОм ± 5% 0,063Вт 0603	1	
R48	Резистор 240 кОм ± 5% 0,063Вт 0603	1	
R49	Резистор 10 кОм ± 5% 0,063Вт 0603	1	
R50	Резистор 5,1 Ом ± 5% 0,063Вт 0603	1	
R51	Резистор 1 МОм ± 5% 0,063Вт 0603	1	
R52	Резистор 10 кОм ± 5% 0,063Вт 0603	1	
R53	Резистор 2,2 кОм ± 5% 0,063Вт 0603	1	
	<b>Диоды</b>		
VD1...VD5	Диод RB160M-60TR-ND	5	
	выпрямительный одиночный, 60 В , 1 А		
	Rohm Semiconductor		
	<b>Транзисторы</b>		
VT1	Транзистор IRF7809AVPBF Nкан 30В 13,3А	1	
	International Rectifier		
VT2,VT3	Транзистор BC850BW биполярный	2	
	РЧ, NPN, 45 В, 100 МГц, 200 мВт, 100 мА, 450 hFE		
	Neperia		
VT4	Транзистор IRLML5103 P-канал 30В 0.76А	1	
	International Rectifier		
VT5	Транзистор MUN5214 T1G	1	
	биполярный цифровой, 50 В, 100 мА, 10 кОм		
	ON Semiconductor		

Подп. и дата  
 Инв. № дудл.  
 Взам. инв. №  
 Подп. и дата  
 Инв. № подл.

Изм.	Лист	№ докум.	Подп.	Дата
------	------	----------	-------	------

12.04.01.2017.071.01.09П33

Лист  
8

