

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

Школа: Информационных технологий и робототехники
Направление подготовки: 15.04.06 - Мехатроника и робототехника
Отделение школы (НОЦ): Автоматизации и робототехники

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

Тема работы
Определение положения электрода сварочного робота с использованием технического зрения

УДК 004.415.2:621.791:007.52

Студент

Группа	ФИО	Подпись	Дата
8ЕМ61	Лашкарёв Ермек Эдуардович		

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Профессор	Сырямкин Владимир Иванович	д.т.н.		
15.04.06 - Мехатроника и робототехника	Мальшенко Александр Максимович	д.т.н.		

КОНСУЛЬТАНТЫ:

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Петухов Олег Нико- лаевич	к.э.н.		

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Бородин Юрий Вик- торович	к.т.н.		

ДОПУСТИТЬ К ЗАЩИТЕ:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Леонов Сергей Влади- мирович	к.т.н.		

Томск – 2018 г.

ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ООП

Код результата	Результаты обучения (выпускник должен быть готов)	Требования ФГОС, критериев и/или заинтересованных сторон
Профессиональные		
P1	Применять глубокие естественно-научные, математические знания в области анализа, синтеза и проектирования для решения научных и инженерных задач производства и эксплуатации мехатронных и робототехнических устройств и систем, в том числе их систем управления.	Требования ФГОС (ПК-1, ПК- 3, ОПК-1, ОПК-4, ОК-1, ОК-9). Критерий 5 АИОР (п. 1.1), согласованный с требованиями международных стандартов EUR-ACE и FEANI
P2	Воспринимать, обрабатывать, анализировать и обобщать научно- техническую информацию, передовой отечественный и зарубежный опыт в области теории, проектирования, производства и эксплуатации мехатронных и робототехнических устройств и систем, принимать участие в командах по разработке и эксплуатации таких устройств и систем.	Требования ФГОС (ПК-3, ПК-4, ПК-7, ОПК-1, ОПК-3, ОК-1, ОК-4, ОК-5, ОК-6, ОК-9). Критерий 5 АИОР (пп. 1.1, 1.2), согласованный с требованиями международных стандартов EUR-ACE и FEANI
P3	Применять и интегрировать полученные знания для решения инженерных задач при разработке, производстве и эксплуатации современных мехатронных и робототехнических устройств и систем (в том числе интеллектуальных) с использованием технологий мирового уровня, современных инструментальных и программных средств.	Требования ФГОС (ПК-2, ПК- 3, ПК-4, ПК-5, ПК-15, ПК-18, ОПК-3, ОПК-6, ОК-1, ОК-5, ОК-6, ОК-7). Критерий 5 АИОР (пп. 1.2), согласованный с требованиями международных стандартов EUR-ACE и FEANI
P4	Определять, систематизировать и получать необходимую информацию в области проектирования, производства, исследований и эксплуатации мехатронных и робототехнических модулей, устройств и систем.	Требования ФГОС (ПК-7, ПК-10, ПК-11, ПК-12, ПК-18, ОПК-4, ОПК-6, ОК-1, ОК-4, ОК-6, ОК-8). Критерий 5 АИОР (п. 1.3), согласованный с требованиями международных стандартов EUR-ACE и FEANI
P5	Планировать и проводить аналитические, имитационные и экспериментальные исследования для целей проектирования, производства и эксплуатации мехатронных и робототехнических средств и систем (в том числе интеллектуальных) с использованием передового отечественного и зарубежного опыта, уметь критически оценивать полученные теоретические и экспериментальные данные и делать выводы.	Требования ФГОС (ПК-1, ПК- 2, ПК-3, ПК-4, ПК-5, ПК-6, ПК-13, ПК-17, ПК-18, ОПК-2, ОПК-3, ОК-1, ОК-3, ОК-4, ОК-6, ОК-7, ОК- 8, ОК-9). Критерий 5 АИОР (п. 1.4), согласованный с требованиями международных стандартов EUR-ACE и FEANI
P6	Понимать используемые современные методы, алгоритмы, модели и технические решения в мехатронике и робототехнике и знать области их применения, в том числе в автоматизированных производствах.	Требования ФГОС (ПК-1, ПК- 2, ПК-3, ПК-7, ОПК-1, ОПК-3, ОПК-4, ОК-5, ОК-9, ОК-10). Критерий 5 АИОР (п. 2.1), согласованный с требованиями международных стандартов EUR-ACE и FEANI
Универсальные		

P7	Эффективно работать в профессиональной деятельности индивидуально и в качестве члена команды	Требования ФГОС (ПК-1, ПК-2 ПК-7, ПК-8, ПК-16, ПК17, ОК-1, ОК-2, ОК-4, ОК-6, ОК-9). Критерий 5 АИОР (п. 2.1), согласованный с требованиями международных стандартов EUR-ACE и FEANI
P8	Владеть иностранным языком на уровне, позволяющем работать в интернациональной среде с пониманием культурных, языковых и социальноэкономических различий	Требования ФГОС (ПК-4, ПК-8, ПК-9, ПК-16, ОПК-4, ОК-5). Критерий 5 АИОР (п. 2.2), согласованный с требованиями международных стандартов EUR-ACE и FEANI
P9	Проявлять широкую эрудицию, в том числе знание и понимание современных общественных и политических проблем, демонстрировать понимание вопросов безопасности и охраны здоровья сотрудников, юридических аспектов, ответственности за инженерную деятельность, влияния инженерных решений на социальный контекст и окружающую среду	Требования ФГОС (ПК-5, ПК-8, ПК-15, ПК-16, ПК-18, ОПК-1, ОПК-4, ОПК-5, ОК-3, ОК-4, ОК-5, ОК-6, ОК-8, ОК-9). Критерий 5 АИОР (пп. 1.6, 2.3.), согласованный с требованиями международных стандартов EUR-ACE и FEAN
P10	Следовать кодексу профессиональной этики и ответственности и международным нормам инженерной деятельности	Требования ФГОС (ПК-8, ПК-11, ПК16, ОПК-3, ОПК-6, ОК-4). Критерий 5 АИОР (пп. 2.4, 2.5), согласованный с требованиями международных стандартов EUR-ACE и FEANI
P11	Понимать необходимость и уметь самостоятельно учиться и повышать квалификацию в течение всего периода профессиональной деятельности.	Требования ФГОС (ПК-4, ПК-8, ОПК3, ОПК-4, ОК-5, ОК-6, ОК-7, ОК-8). Критерий 5 АИОР (2.6), согласованный с требованиями международных стандартов EUR-ACE и FEANI.

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

Школа: Информационных технологий и робототехники
Направление подготовки: 15.04.06 - Мехатроника и робототехника
Отделение школы (НОЦ): Автоматизации и робототехники

УТВЕРЖДАЮ:
Руководитель ООП
_____ Малышенко А.М.
(Подпись) (Дата) (Ф.И.О.)

ЗАДАНИЕ
на выполнение выпускной квалификационной работы

В форме:

Магистерской диссертации (бакалаврской работы, дипломного проекта/работы, магистерской диссертации)

Студенту:

Группа 8ЕМ61	ФИО Лашкарёв Ермек Эдуардович
------------------------	---

Тема работы:

Определение положения электрода сварочного робота с использованием технического зрения
Утверждена приказом директора (дата, номер)

Срок сдачи студентом выполненной работы:	
--	--

ТЕХНИЧЕСКОЕ ЗАДАНИЕ:

<p>Исходные данные к работе <i>(наименование объекта исследования или проектирования; производительность или нагрузка; режим работы (непрерывный, периодический, циклический и т. д.); вид сырья или материал изделия; требования к продукту, изделию или процессу; особые требования к особенностям функционирования (эксплуатации) объекта или изделия в плане безопасности эксплуатации, влияния на окружающую среду, энергозатратам; экономический анализ и т. д.).</i></p>	Интеграция системы технического зрения для определения координат электрода. Осуществить калибровку камер с помощью библиотеки OpenCV. Выбрать подходящий метод технического зрения для нахождения контуров электрода.
<p>Перечень подлежащих исследованию, проектированию и разработке вопросов <i>(аналитический обзор по литературным источникам с целью выяснения достижений мировой науки техники в рассматриваемой области; постановка задачи исследования, проектирования, конструирования; содержание процедуры исследования, проектирования, конструирования; обсуждение результатов выполненной работы; наименование дополнительных разделов, подлежащих разработке; заключение по работе).</i></p>	Анализ основных функциональных аспектов технического зрения. Обзор готовых существующих решений систем технического зрения сварочных роботов. Разработать конструкторское решение для камер. Разработать алгоритм работы системы опре-

	деления контуров и координат электрода. Программно реализовать алгоритм. Провести тестовые испытания для определения его работоспособности.
Перечень графического материала <i>(с точным указанием обязательных чертежей)</i>	
Консультанты по разделам выпускной квалификационной работы <i>(с указанием разделов)</i>	
Раздел	Консультант
«Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»	Петухов Олег Николаевич, доцент ОСГН ШБИП ТПУ
«Социальная ответственность»	Бородин Юрий Викторович, доцент ОКД ИШНКБ
«Иностранный язык»	Горбатова Татьяна Николаевна, старший преподаватель ОИЯ ШБИП
Названия разделов, которые должны быть написаны на русском и иностранном языках:	

Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику	01.02.2018
---	------------

Задание выдал руководитель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Профессор	Сыряжкин В.И.	д.т.н.		01.02.2018

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8ЕМ61	Лашкарёв Ермек Эдуардович		01.02.2018

ЗАДАНИЕ ДЛЯ РАЗДЕЛА «ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И РЕСУРСОСБЕРЕЖЕНИЕ»

Студенту:

Группа	ФИО
8ЕМ61	Лашкарёв Ермек Эдуардович

Школа	ИШИТР	Отделение	Автоматизации и робототехники
Уровень образования	Магистр	Направление/специальность	Мехатроника и робототехника

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

1. Стоимость ресурсов научного исследования (НИ): материально-технических, энергетических, финансовых, информационных и человеческих	Итоговая стоимость затрат на НИ составила 176867,57
2. Нормы и нормативы расходования ресурсов	Тарифные ставки заработной платы рабочих, нормы амортизационных отчислений, нормы времени на выполнение операций и др
3. Используемая система налогообложения, ставки налогов, отчислений, дисконтирования и кредитования	Отчисления во внебюджетные страховые фонды (30 %)

Перечень вопросов, подлежащих исследованию, проектированию и разработке:

--	--

Перечень графического материала (с точным указанием обязательных чертежей):

--	--

Дата выдачи задания для раздела по линейному графику

--	--

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОСГН ШБИП ТПУ	Петухов О.Н.	к.э.н., доцент		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8ЕМ61	Лашкарёв Ермек Эдуардович		

Задание для раздела «СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»

Группа	ФИО		
8ЕМ61	Лашкарёв Ермек Эдуардович		
Школа	ИШИТР	Отделение	Автоматизации и робототехники
Уровень образования	Магистр	Направление/специальность	Мехатроника и робототехника
Исходные данные к разделу «Социальная ответственность»:			
<p>Описание рабочего места (рабочей зоны, технологического процесса, механического оборудования) на предмет возникновения:</p>	<p>Рабочим местом является аудитория № 109 в Научном парке Томского политехнического университета. В аудитории рабочей зоной является место за персональным компьютером, отведённое студенту для выполнения работы. Технологический процесс представляет собой разработку алгоритма для нахождения электрода манипулятора KUKA с помощью технического зрения. Основным оборудованием, на котором производится работа, является персональный компьютер с периферийными устройствами.</p> <p>Вредными факторами производственной среды, которые могут возникнуть на рабочем месте, являются: недостаточная освещённость рабочей зоны, отсутствие или недостаток естественного света, повышенный уровень шума, повышенный уровень электромагнитных излучений, повышенная или пониженная влажность воздуха, повышенная напряжённость электрического поля.</p> <p>Опасными проявлениями факторов производственной среды, которые могут возникнуть на рабочем месте, являются: электрический ток.</p> <p>Чрезвычайной ситуацией, которая может возникнуть на рабочем месте, является возникновение пожара.</p>		
Перечень вопросов, подлежащих исследованию, проектированию и разработке:			
<p>1. Производственная безопасность</p> <p>1.1. Анализ выявленных вредных факторов при разработке и эксплуатации проектируемого решения в следующей последовательности</p> <p>1.2. Анализ выявленных опасных факторов при разработке и эксплуатации проектируемого решения в следующей последовательности:</p>	<p>Анализ выявленных вредных факторов:</p> <ul style="list-style-type: none"> • отсутствие или недостаток естественного света; • уровень шума; • уровень электромагнитных излучений; • напряжённость электрического поля; • микроклимат. <p>Анализ выявленных опасных факторов:</p> <ul style="list-style-type: none"> • электрический ток 		
<p>2. Охрана окружающей среды:</p>	<p>Воздействие объекта на атмосферу, гидросферу не происходит.</p> <p>В работе проведён анализ воздействия на литосферу (образование отходов при поломке ПК, отходов при печати, при утилизации ламп).</p>		

3. Защита в чрезвычайных ситуациях:	В аудиторном помещении возможно ЧС техногенного характера – пожар (возгорание).
--	---

4. Правовые и организационные вопросы обеспечения безопасности:	Рабочее место при выполнении работ в положении сидя должно соответствовать требованиям ГОСТ 12.2.032-78. Требования к организации оборудования рабочих мест с ПК регулируются в СанПиН 2.2.2/2.4.1340-03.
--	---

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

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОКД ИШНКБ	Бородин Ю.В.	к.т.н., доцент		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8ЕМ61	Лашкарёв Ермек Эдуардович		

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

Школа: ИШИТР

Направление подготовки: Мехатроника и робототехника

Уровень образования: магистратура

Отделение школы (НОЦ): автоматизации и робототехники

Период выполнения: весенний семестр 2017/2018 учебного года

Форма представления работы:

Магистерская диссертация

(бакалаврская работа, дипломный проект/работа, магистерская диссертация)

**КАЛЕНДАРНЫЙ РЕЙТИНГ-ПЛАН
выполнения выпускной квалификационной работы**

Срок сдачи студентом выполненной работы:	
--	--

Дата контроля	Название раздела (модуля) / вид работы (исследования)	Максимальный балл раздела (модуля)
15.02.2018 г.	Анализ готовых технических решений сварочных роботов с техническим зрением	15
25.02.2018 г.	Интеграция технического зрения для определения электрода сварочного манипулятора	15
10.03.2018 г.	Сборка конструкции из фанер для камер	20
12.04.2018 г.	Практическая реализация системы технического зрения	20
15.05.2018 г.	Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	10
18.05.2018 г.	Социальная ответственность	10
21.05.2018 г.	Английский язык	10

Составил преподаватель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Профессор	Сырямкин В. И.	д.т.н.		

СОГЛАСОВАНО:

Руководитель ООП	ФИО	Ученая степень, звание	Подпись	Дата
Профессор	Малышенко А.М.	д.т.н.		

Реферат

Выпускная квалификационная работа содержит 116 страниц, 57 рисунок, 13 таблиц, список используемых источников содержит 37 наименования. Ключевые слова: technical vision, computer vision, OpenCV, electrode, техническое зрение, компьютерное зрение, сварочный робот, электрод, программирование, определение координат, контур, манипуляторы.

Объектом исследования является электрод сварочного манипулятора KUKA KR – 16 – 2. Предметом исследования является программная часть системы технического зрения с помощью библиотеки OpenCV.

Цель данной работы – разработка системы для определения координат электрода манипулятора KUKA KR – 16 – 2. В процессе исследования обзревается проблемы доступа к программным решениям технического зрения, которые реализованы крупными компаниями. Представляется обзор существующих решений в области технического зрения сварочных роботов.

Разработана техническая часть в виде конструкции и креплений для электрода и камер. Написан программный код на языке C++ работы системы определения координат и контуров электрода. Проводились определение социальной ответственности, расчет финансовой эффективности проекта.

В лаборатории KUKA были проведены тестовые испытания работы, которые показали работоспособность технического решения.

Область применения разработанной системы: производства, где используют сварочных роботов. Изменив техническую часть, она так же может использоваться для различных целей технического и компьютерного зрения.

Определения, обозначения, сокращения, нормативные ссылки

В данной работе применены следующие термины с соответствующими определениями:

USB – Universal Serial Bus (универсальная последовательная шина).

СТЗ – Системы технического зрения

CCD – Charge Coupled Device (прибор с зарядовой связью)

CMOS – Complementary Metal-Oxide-Semiconductor (комплементарная структура металл-оксид-полупроводник)

TCP - Transmission Control Protocol (протокол управления передачей)

IP 54 – Международный класс защиты, частичная защита от пыли и от брызг

ПК – Персональный Компьютер

HD – High Definition (Высокая чёткость)

OpenCV – Open Source Computer Vision Library (библиотека алгоритмов компьютерного зрения)

RGB – Red, Green, Blue (красный, зелёный, синий)

HSV – Hue, Saturation, Value (тон, насыщенность, значение)

БЖД – Безопасность Жизнедеятельности Человека

ЭВМ – Электронно-вычислительная машина

СОДЕРЖАНИЕ

1. ПОСТАНОВКА ЗАДАЧИ	15
1.1 Обзор вопросов.....	15
1.2. Используемые программные средства	18
1.2.1 Visual Studio Professional 2017	18
1.2.2 Библиотека Open Source Computer Vision	19
2. СВАРОЧНЫЕ РОБОТЫ С ТЕХНИЧЕСКИМ ЗРЕНИЕМ	21
2.1 Примеры сварочных роботов с техническим зрением	21
2.1.1 Сварочный робот компании Fanuc Corporation.....	21
2.1.2 Сварочный робот компании ABB Robotics	25
2.1.2 Сварочный робот компании KUKA Roboter	26
2.2 Техническая подготовка	28
3. АНАЛИЗ БИБЛИОТЕКИ OPENCV	35
3.1 Знакомство с библиотекой OpenCV	35
3.2 Калибровка камеры	40
3.3 Определение контуров с помощью детектора границ Canny	52
3.4 Определение контуров с помощью оператора Лапласа	56
3.5 Определение контуров с помощью детектора границ Собеля	58
4. РЕЗУЛЬТАТЫ ПРАКТИЧЕСКОЙ ЧАСТИ.....	63
4.1 Реализованный поиск контура электрода	63
4.2 Программная часть работы двух камер	65
4.3 Поиск пиксельных координат	68
5.СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ.....	74
5.1 Техногенная безопасность.....	75
5.1.1 Микроклимат	75
5.1.2 Естественная освещенность.....	76
5.1.4 Уровень электромагнитных излучений	78
5.1.5 Электробезопасность.....	79
5.2 Организационные мероприятия обеспечения безопасности	83
5.2.1 Эргономические требования к рабочему месту	83

5.3 Пожарная безопасность.....	84
5.4 Особенности законодательного регулирования проектных решений.....	86
6. ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И РЕСУРСОСБЕРЕЖЕНИЕ	87
6.1 Организация и планирование работ.....	87
6.2 Продолжительность этапов работ	89
6.3 Расчет сметы затрат на выполнение проекта	93
6.3.1 Расчет затрат на материалы и покупные изделия.....	93
6.3.2 Расчет заработной платы	94
6.3.3 Расчет затрат на электроэнергию	96
6.3.4 Расчет затрат на социальный налог	97
6.3.5 Расчет амортизационных расходов.....	97
6.3.6 Расчет прочих (накладных) расходов	97
6.3.7 Расчет общей себестоимости разработки.....	98
Заключение.....	99
Список использованной литературы	100
Приложение А.....	103
Приложение Б	117

ВВЕДЕНИЕ

В настоящее время с развитием технологии в промышленности, производстве, разведки и автоматизации продуктов для повышения качества процесса используются системы машинного зрения. Основными примерами является обработка продукта, где нужно визуальное детектирование на основе изображения. Свою актуальность в данной сфере получили сварочные роботы. За последние годы 80% промышленных роботов, поставляемые в Российскую Федерацию, – это сварочные роботы. Сварочные роботизированные технологические комплексы (РТК) уже успешно используются в крупных предприятиях. В основном применяются при точечной контактной, дуговой и лазерной сварке [1].

Все данные роботы в основном закупаются у крупных компаний, производящие оборудования для промышленной автоматизации. Из-за чего частные компании не имеют полного доступа к программной части оборудования, что является недостатком при интеграции с другими видами программ.

Существуют основные конкурирующие компании в сфере производства промышленных роботов: Fanuc Corporation и KUKA Roboter. Fanuc Corporation очень популярна за счет своих контроллеров R30 – iA. Одной из особенностей данного контроллера является встроенная интегрированная система iRVISION, которая позволяет с помощью системы визуального обнаружения по технологии plug & play распознавать и определять местоположение произвольно расположенных изделий любой формы и размера. KUKA Roboter тоже имеет ряд сварочных манипуляторов с системами технического зрения, но менее популярные чем iRVISION [1].

Все данные системы имеют недостаток интегрирования с другими программными средствами. С помощью лаборатории KUKA, находящийся в научном парке ТПУ постараемся частично решить данную задачу. В данной работе с помощью изображения электрода, полученные с камеры будем определять положения электрода. Основной задачей стоит нахождение координаты пикселя и написание программной части на языке C++ с помощью библиотеки Open CV.

1. ПОСТАНОВКА ЗАДАЧИ

Цель работы: определение положения координат электрода сварочного робота KUKA KR – 16 – 2 с помощью технического зрения. Для достижения необходимо решить следующие задачи:

- ознакомиться с библиотекой Open CV;
- определить пиксельные координаты изображения электрода;
- вычислить мировые координаты электрода.

1.1 Обзор вопросов

Сварка с помощью манипуляторов представляет собой полностью автоматизированную систему, который используется за счет специального использования сварочного оборудования. Преимуществами такой сварки является высокое качество готовых изделий и высокая производительность сварочного производства. В каждом современном производстве существует масса важных особенностей. Область сварки не является исключением [2].

Основным преимуществом сварки с помощью манипулятора является высокая точность позиционирования сварочной горелки. Однако бывают случаи, когда сварочный робот не может найти точку для сварки как человек. Из-за этого погрешность правильной сварки повышается, что является не приемлемым. Для решения этой проблеме используют различные методы.

Если достичь данной точности позиционирования невозможно, следует применять методы коррекции сварочной траектории. Примером является использование лазерной системы слежения за стыком шва. Коррекция траекторий позволит сохранить качество сварного изделия. При этом её использование понизит производительность вплоть до 30% [1].

Лидирующие позиции в разработке роботизированных сварочных комплексов принадлежат сегодня компании KUKA Roboter. Под этой маркой выпускаются абсолютные специалисты в области электродуговой сварки в среде

защитного газа – роботы серии HW и роботы вида KUKA KR – 16 – 2. Имея грузоподъёмность до 16 кг и радиус действия до 2016 мм, данные манипуляторы выполняют сварку даже труднодоступных соединений. Благодаря наличию шестой оси с возможностью бесконечного вращения, исключаются временные затраты на возврат в начальное положение, и время обработки детали сокращается [4].

В Научном парке ТПУ стоят несколько манипуляторов KUKA. Среди которых можно увидеть: роботизированную ячейку для сварки KUKA KR – 16 – 2, роботизированную ячейку для фрезерной обработки KUKA KR – 120 – R2700, роботизированную ячейку для паллетирования KUKA KR – 40 – PA и другие учебные роботы. KUKA KR – 16 – 2 изображен на рисунке 1.1. Роботизированная ячейка для фрезерной обработки KUKA KR – 120 – R2700 изображена на рисунке 1.2 [4].



Рисунок 1.1 - Роботизированная ячейка для сварки KUKA KR – 16 – 2



Рисунок 1.2 - Роботизированная ячейка для фрезерной обработки KUKA KR – 120 – R2700

Основной проблемой для сварочного робота KUKA KR – 16 – 2 было в неудобстве работы с электродом при каждом запуске самого робота. В ходе работы нужно было вручную настраивать положение робота, так как электрод уменьшался и тем самым нужно было изменять позиции робота. В ходе решения проблемы было решено использовать техническое зрение, что позволило бы определять пиксельные координаты изображения электрода. После нахождения пиксельных координат нужно было найти зависимость между остальными координатами. На тот момент сотрудниками лаборатории была придумана программа на языке C++, позволяющая напрямую передавать координаты на панель управления роботом. В итоге основной проблемой оставалась найти сами координаты изображения электрода, а также найти зависимость между мировыми координатами, координатами камеры и координатами изображения.

1.2. Используемые программные средства

1.2.1 Visual Studio Professional 2017

Microsoft Visual Studio Professional 2017 – среда разработки, предназначенная для разработки различных компьютерных и не только программ. С помощью данной программы можно создавать также веб-сайты, мобильные приложения и т.д. В программе существуют очень много различных программируемых языков. Популярным из которых является язык программирования C++. Кроме этого можно создавать код и взаимодействовать его с остальными. Также может создавать как собственный код, так и управляемый код [8].

Программа включает в себя исходный код, а также отладчик исходного кода. Здесь не нужно дополнительно устанавливать компилятор, т.к. встроен в саму программу. Visual Studio очень удобен в плане кодинга, потому что содержит для этого все уровни. Содержит строку для вывода ошибок, а также корректор кода. Кроме этого содержит разные редакторы, как для графического интерфейса, т.к. и для веб приложений.

Visual Studio поддерживает 36 различных языков программирования и позволяет редактору кода и отладчику поддерживать (в той или иной степени) практически любой язык программирования, если существует специфическая для языка служба. Встроенные языки включают в себя C, C ++, C ++ / CLI, Visual Basic .NET, C#, F#, JavaScript, TypeScript, XML, XSLT, HTML и CSS. Так же поддерживает другие языки, такие как Python, Ruby, Node.js и Java [8].

Основным преимуществом данной программы является ее доступность, а также возможность интегрирования с библиотекой OpenCV. Кроме того поддерживает язык программирования C++, который тоже интегрирован с OpenCV.

В случае если не получится реализовать программную часть на языке C++ можно будет воспользоваться языком Python, который тоже можно использовать в данной программе. Благодаря библиотеки OpenCV мы получим доступ к техническому зрению, который играет основную роль в нашем проекте.

По технической части и совместимости с персональным компьютером программа занимает небольшую часть памяти, что вполне является доступным в современное время. Удобством данной программы также состоит в том, что все языки программирования находятся вместе и можно использовать их в любой момент. Ведь в других случаях для каждого языка нужно было бы устанавливать все по отдельности.

Рабочее окно Visual Studio Professional 2017 представлена на рисунке 1.3. В данном окне содержится исходный код для работы с двумя камерами.

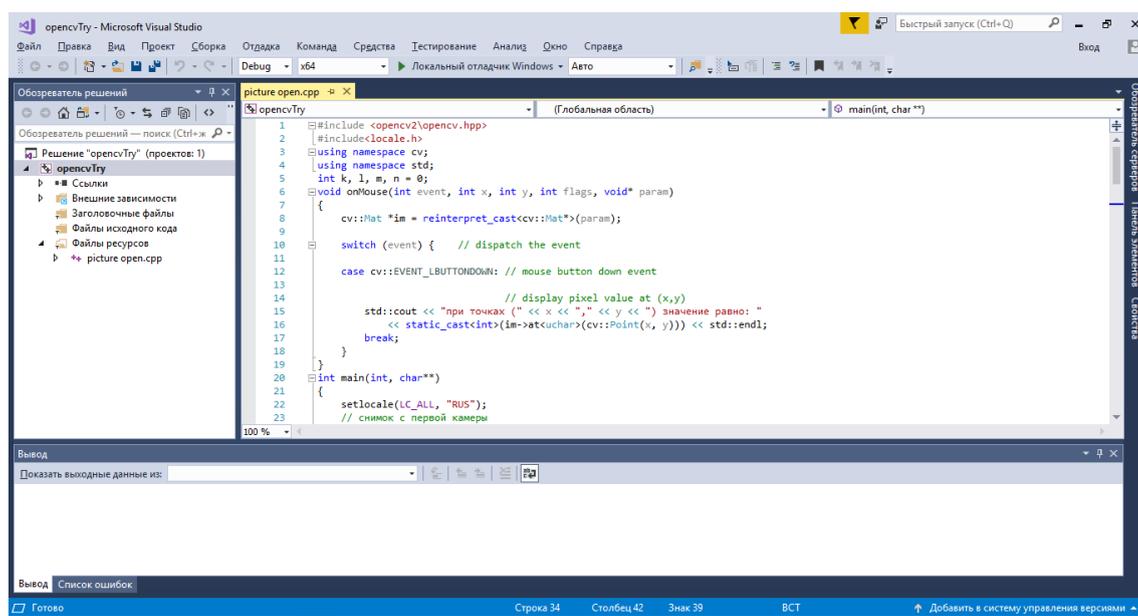


Рисунок 1.3 - Рабочее окно Visual Studio Professional 2017

1.2.2 Библиотека Open Source Computer Vision

OpenCV (Open Source Computer Vision Library) представляет собой библиотеку для технического зрения с открытым исходным кодом, а также библиотеку для машинного обучения. OpenCV была создана как приложение для технического зрения, а также для роста интереса общества к техническому зрению. Данная библиотека является полностью бесплатной и доступна в сети, а также упрощает процесс ознакомления с техническим зрением [14].

Библиотека содержит около 2500 алгоритмов, предназначенный для современного компьютерного зрения и машинного обучения. Данные алгоритмы можно использовать как для работы обычных камер, так и для более профессиональных. Алгоритмы содержат в себе работу с камерами для идентификации объектов, а так же для ряда других работ с техническим зрением. Несколько примером их использования: извлечения 3D-моделей объектов, создания 3D-облаков точек из стереокамер, сшивания изображений вместе для получения высокого разрешения изображение всей сцены, нахождение изображения из базы данных, сглаживание и улучшение изображений, возможность слежения за движениями глаз, распознавания декорации и создания маркеров, чтобы накладывать их на дополненную реальность и т. д. Библиотека широко используется в компаниях, исследовательских группах, учебных заведениях и правительственных органах [14].

Известные компании на весь мир типа Google, Microsoft, Intel и другие используют данную библиотеку. Существует много стартапов, таких как Applied Minds, VideoSurf и Zeitera, которые широко используют библиотеку OpenCV.

Библиотека имеет интерфейсы C++, Python, Java, MATLAB и поддерживает Windows, Linux, Android и Mac OS. OpenCV написан на языке C++, что очень удобно в нашем случае. Официальный сайт, где можно скачать библиотеку представлена на рисунке 1.4.

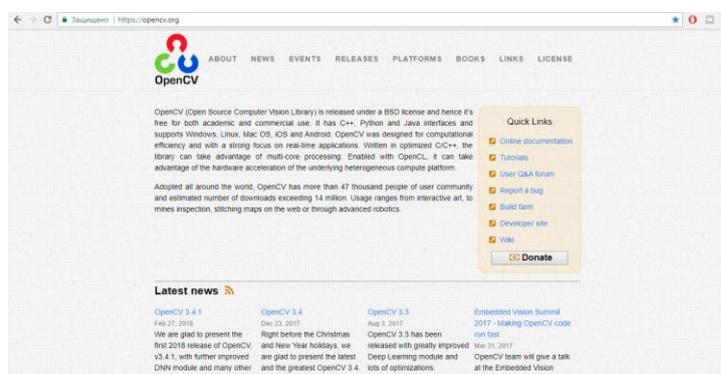


Рисунок 1.4 – Официальный сайт OpenCV

2. СВАРОЧНЫЕ РОБОТЫ С ТЕХНИЧЕСКИМ ЗРЕНИЕМ

2.1 Примеры сварочных роботов с техническим зрением

В настоящее время популярными среди промышленных роботов являются роботизированная сварка. Сварочные роботы позволяют оптимизировать весь процесс сварки. Как говорилось выше это очень удобно и выгодно, но существуют и свои недостатки. Основными являются работа именно при сварке какой-либо детали. У обычного манипулятора точность позиционирования падает при смещении шва, где человек в свою очередь мог бы догадаться, но не робот. В крупных компаниях существуют различного рода решения подобных ситуаций. В дальнейшем будут приведены примеры сварочных роботов с техническим зрением [1].

2.1.1 Сварочный робот компании Fanuc Corporation

Fanuc Corporation один из лидеров производителей оборудования для автоматизации. Компания очень популярная и имеет очень много штабов по всему миру. Основными направлениями является: ЧПУ, промышленной роботы и лазерное оборудование.

Очень популярна компания благодаря серии своих сварочных роботов с техническим зрением iR-Vision. Манипулятор с техническим зрением iR-Vision компании Fanuc Corporation представлен на рисунке 2.1 и на рисунке 2.2 [3].



Рисунок 2.1 – Манипулятор Fanuc Corporation с вертикальной камерой



Рисунок 2.2 – Манипулятор Fanuc Corporation со встроенной камерой

Рассмотрим некоторые функциональные возможности системы iRVision, используемые при роботизированной сварке [5]:

1. VISION SHIFT применяется в контактной и дуговой сварке для компенсации погрешности между позиционированием детали-шаблона, по которой производилось программирование робота, и позиционированием последующих деталей, на которых должна выполняться программа.

Возможности VISION SHIFT:

- функция использует возможности технического зрения для автоматической 3D калибровки программы робота;
- возможно автоматически калибровать рабочую точку инструмента (TCP), что ранее могло быть сделано только вручную;
- на высокоточную неконтактную калибровку программ вместо нескольких часов уходит всего до 30 минут. Таким образом, эта функция значительно сокращает время пуско-наладки;
- поддерживает установку центральной точки инструмента и кончика электрода для управления роботом с большой точностью.
- возможно применять для калибровки групп совместно работающих роботов (Dual-, Triple-, и Quadarm);

- позволяет делать юстировку роботов FANUC после смены двигателей, энкодеров и т.д.

2. FANUC iRVISION Weld Tip служит для визуального контроля и проверки состояния рабочего конца электрода для точечной сварки на предмет повреждений. Это специализированное программное обеспечение может выполнять самые разнообразные проверки, включая измерение износа съёмного конца электрода и определение, производилась ли его замена или заправка.

3. FANUC iRVISION Torch Mate оснащено возможностями программирования на производстве и промышленной камерой. Оно предотвращает возникновение проблем с выравниванием за счёт удерживания центральной точки инструмента на траектории движения инструмента. Система iRVISION обеспечивает стабильное качество сварки, а также увеличивает время безотказной работы за счёт устранения необходимости в упреждающих действиях.

4. 2D Multiview Vision – техническое зрение с использованием нескольких камер для определения точного положения детали. Позволяет точно определять положение деталей больших габаритов.

Контроллер R30-iA в своем составе программного обеспечения имеет приложение Arctool, разработанное для упрощения и стандартизации установки и управления сварочными приложениями. Рассмотрим некоторые функции приложения:

- weaving – сварка с колебаниями. Обеспечивает колебательное движение горелки относительно линии сварки, увеличивающее требуемую толщину шва;

- weld Resume – при остановке работа во время движения по сварочной дорожке и перемещению обратно к дорожке, устанавливается необходимое расстояние, и сварка продолжается.

- защита электрода – компенсирует значение положения кончика электрода проволокой после долгого времени сварки или замены кончика электрода. В этих случаях, сварочные операции не могут выполняться корректно. Защита

электрода помогает предотвратить сбой центральной точки инструмента (TCP), в противном случае её необходимо перезаписать;

- automatic adjustment of the tool center point – функция автоматического восстановления горелки обеспечивает автоматическую регулировку центральной точки инструмента (TCP). Автоматическое восстановление электрода автоматически компенсирует изгиб сварочного электрода и износ наконечника, уменьшая количество дефектов и увеличивая производительность;

- torch guard collision detection – параметры для запястья очень чувствительны к столкновениям электрода. Функция останавливает робот до того момента, как электрод, столкнувшись с препятствием, сломается;

- arc sensor TAST – (в момент движения по шву) метод контроля шва, используемый для контроля положения поверхности детали, что очень важно для роботизированной дуговой сварки. Этот метод используется в качестве датчика контроля сварочного тока в случае изменения длины дуги. Робот будет компенсировать изменение положения сварочной горелки;

- контроль положения горелки TAST позволяет контролировать положение электрода за счёт изменения сварочного тока при амплитудном движении. Программа определяет центр шва, сравнивая левую и правую половины по току. Также можно контролировать высоту положения электрода. Положение электрода задано (согласно значению тока) и сохраняется по середине амплитуды;

- touch sensing состоит из: а) движения центральной точки инструмента (TCP) к объекту, используя контроль движения робота, скорость и направление; б) использование входящих сигналов для определения начала контакта робота с объектом; в) сохранение положений объекта в позиционные регистры; г) использование информации о сохраненных положениях роботом. Touch Sensing определяет положение отклонения сварочных швов в момент сварки с помощью Arc Sensor TAST. Touch sensing позволяет роботу изменять направление дорожки автоматически при движении в случае, если поверхность детали неточна, либо деталь занимает некорректное положение.

- опция Automatic Voltage Control (AVC) компенсирует изменение формы детали и отклонение во время сварочного процесса TIG (сварка вольфрамовым электродом в инертном газе) либо других процессов при постоянном токе, например, при выполнении плазменной резки. AVC компенсирует различие между оптимальными и реальными сварочными параметрами во время движения по сварочной дорожке. Функция AVC демонстрирует напряжение при дуговой сварке TIG процесса. Она позволяет изменять высоту горелки над деталью для сохранения постоянного сварочного напряжения [3].

2.1.2 Сварочный робот компании ABB Robotics

ABB Robotics – это компания, производящая промышленное оборудование. В том числе и роботизированные комплексы. Хотя и компания была основана в 1988 году, но уже является одним из лидеров в мире. У компании много различных манипуляторов: для сборки и разборки, палеттирование, точечной и дуговой сварки, лакировки и покраски, а также других. Одним из роботов является манипулятор для электродуговой сварки IRB 1520ID. Новый манипулятор IRB 1520ID, высокопроизводительный робот со встроенным сварочным оборудованием, разработанный специально для электродуговой сварки. Все кабели и подающие каналы проложены внутри манипулятора, что упрощает программирование и обеспечивает оптимальную защиту для всех элементов. IRB 1520ID представлен на рисунке 2.3 [11].



Рисунок 2.3 – Манипулятор для электродуговой сварки
компании ABB Robotics

В компании ведутся исследования технического зрения. Одним из примеров является робот Roberta. Роботы по имени Roberta, были специально разработаны для маленьких и средних предприятий, которые хотят добиться эффективной промышленной автоматизации. Целью разработчиков было создание гибких, легких и эргономичных роботов, которые можно без труда перемещать по цеху [11].

Основные Характеристики Roberta:

- легкий вес;
- 6 степеней свободы без точки сингулярности;
- большая грузоподъемность.

В роботах Roberta представлены несколько интегрированных концепции безопасности. Roberta может работать с определенным захватом, который является абсолютно безопасным для человека. Специальные датчики, еще один аспект безопасности, который снижает риск каких-либо повреждений. Робот также оснащен техническим зрением. То есть он может видеть объект, с которым работает. Серия Roberta доступна в 3 различных моделях с грузоподъемностью 4.0 кг, 8.0 кг и 12,0 кг. Различия между моделями – в максимальной досягаемости и грузоподъемности.

2.1.2 Сварочный робот компании KUKA Roboter

Промышленные роботы Kuka – одни из популярных роботов сварщиков в мире. Они с успехом заменяют ручной труд на многих европейских предприятиях. Для сварочных работ у компании есть как модели высокой грузоподъемности, а также сварочные роботы малой грузоподъемности. Перечислим некоторые модели сварочных роботов Kuka [10]:

- модель KR 5 arc рекомендуют, когда нужны относительно простые роботы для сварки(дуговой), паяльных работ, окраски и прочих манипуляций. Её полезная нагрузка равна 5 кг, дополнительная – 12, а максимальный радиус действия измеряется 1412 мм. Модель изображена на рисунке 2.4;



Рисунок 2.4 – Модель KR 5 arc

- роботизированный комплекс для сварки в среде защитных газов позволяет работать даже с очень сложными деталями крупных размеров, также и на большой глубине деталей. Для подачи газа используют различные комплекты шлангов. Это модель KR 5-2 arc HW с теми же параметрами веса и рабочего радиуса, что и у предыдущей. Такие роботы могут крепиться и на полу, и на потолке, эффективно выполняя пайку и сварку, в том числе и электродугую. Сюда же относится KR 16 L8 arc HW [10];
- KR 16 L8 arc HW входит в серию L вместе с KR 16 L6-2. Удлиненный манипулятор позволяет увеличить рабочую зону до 2016 мм и 1911 мм соответственно, но грузоподъемность снижается (8 и 6 кг);
- целый комплекс работ может выполнять многофункциональный промышленный робот, представленный моделями KR 6-2 и KR 16-2 (в каталоге Кука обозначаются как KR 6-3 и KR 16-3), которые могут настраиваться под десятки различных операций. Первая модель обладает полезной нагрузкой 6 кг и дополнительной – 10, с максимальным радиусом в 1611 мм. Вторая модель имеет такие же пределы дополнительной нагрузки, но полезная составляет 16 кг, а рабочая зона – 1610 мм. Модель KR 16-2 представлен на рисунке 1;

– для скоростных работ, где требуется усиленная мощность, создан робот KR 16-3 S, у которого время такта сокращено на 18%. Он обладает грузоподъемностью 16 кг и досягаемостью 1611 мм [7].

Также есть и другие модели с различными возможностями. Для машин малой грузоподъемности масса груза может быть от пяти до 16 кг, а максимальная загрузка в пределах 36-48 кг. Все данные роботы могут быть оснащены камерами для выполнения работ, связанные с техническим зрением. Компания KUKA Roboter имеет ряд решений в данной области, но эти решения не находятся в открытом доступе.

2.2 Техническая подготовка

В ходе решения задачи были проанализированы методы для решения поставленной задачи. Для начала были изучены возможные решения в статьях, связанной с техническим зрением, а точнее с системами для определения объектов и их координат. Но как таковых решений не было.

Для определения координат электрода нужно было место для них и для манипулятора. В связи с этим была придумана конструкция, которая позволила бы делать снимки электрода манипулятора. Г образная конструкция представлена на рисунке 2.5.

Данная конструкция была смоделирована в программе Autodesk Inventor. Как вы видите нижняя часть конструкции размером 31,8 см на 31,8 см. Размеры для листов фанер взяты под А4: 21,8 см и 31,8 соответственно. Для каждого листа взят запас прочности: в 0,8 см для одного, 2,1 для второго. Это связано с тем, что при креплении листы фанеры пересекаются друг с другом.

После того, как модель была готова, нужно было ее изготовить. В этом нам помогла роботизированная ячейка для фрезерной обработки KUKA KR 120 R2700, которая вырезала модель конструкций. В ходе работы были соблюдены все размеры, а также безопасность персонала. Весь персонал находился снаружи

кабинки для манипулятора. Фрезерный манипулятор представлен на рисунке 2.6.

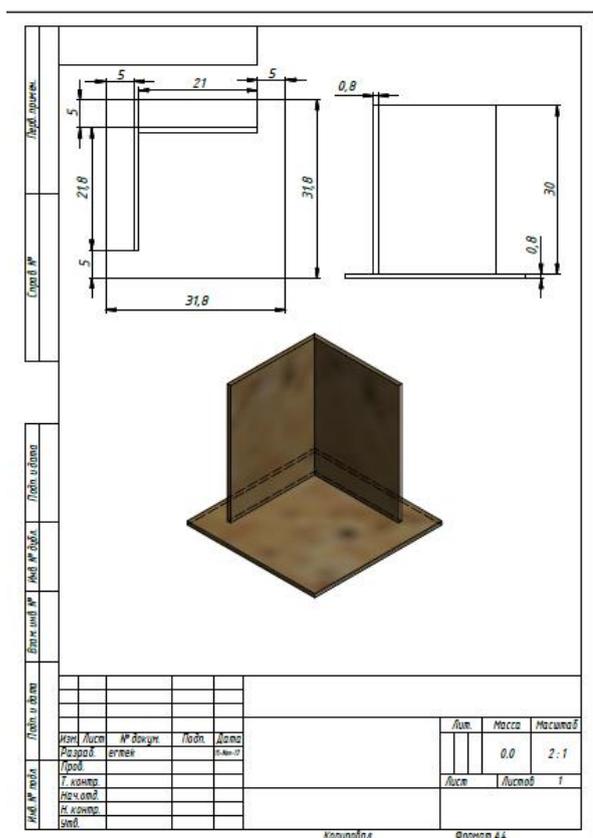


Рисунок 2.5 – Г образная конструкция фанер, собранная в Autodesk Inventor



Рисунок 2.6 – Роботизированная ячейка для фрезерной обработки KUKA KR 120 R2700

В техническую часть так же входят 2 веб - камеры Microsoft «LifeCam Cinema» 6CH-00002. Данные камеры очень чувствительны к свету и имеют изгибающую часть для крепления. Технические характеристики веб - камеры Microsoft «LifeCam Cinema» 6CH-00002 представлены в таблице 1.1. Камера изображена на рисунке 2.7 [12].

Таблица 1.1 - Технические характеристики камеры Microsoft «LifeCam Cinema»

Поддерживаемые разрешения видео	1280x720
Максимальная частота кадров	30 Гц
Тип подключения	USB 2.0
Размещение	Крепление на ЖК-монитор
Тип матрицы	CMOS
Число мегапикселей матрицы	0,9
Поддерживаемые разрешения фотоснимков	1280x960
Фокусировка	Автоматическая фокусировка
Коннекторы и интерфейсы	USB Тип А штекер (USB 2.0)
Материал корпуса	Пластик
Размеры (ШxВxГ), мм	46x40x56
Вес, г	95



Рисунок 2.7 – Веб - камеры Microsoft «LifeCam Cinema» 6CH-00002

После работы с фрезерным манипулятором из кусков вырезанных фанер была собрана Г конструкция. Для крепления использовались маленькие шурупы, так как на поверхности фанер не должно было быть неровностей. Кроме этого небольшие шурупы позволяли без проблем поставить конструкцию на стол манипулятора, потому что их не было видно, и конструкция была ровной.

После того как куски фанер были собраны вместе, оставалось закрепить их на столе. В лаборатории использовались крепления, которые были уже заранее обдуманы для столов манипулятора. Данные крепления очень устойчивы и надежны. Вид полученной конструкции изображена на рисунке 2.8. На рисунке видны те самые крепления желтого цвета позади фанер. Также вид конструкции на фоне сварочного манипулятора KUKA KR – 16 – 2 показана на рисунке 2.9.

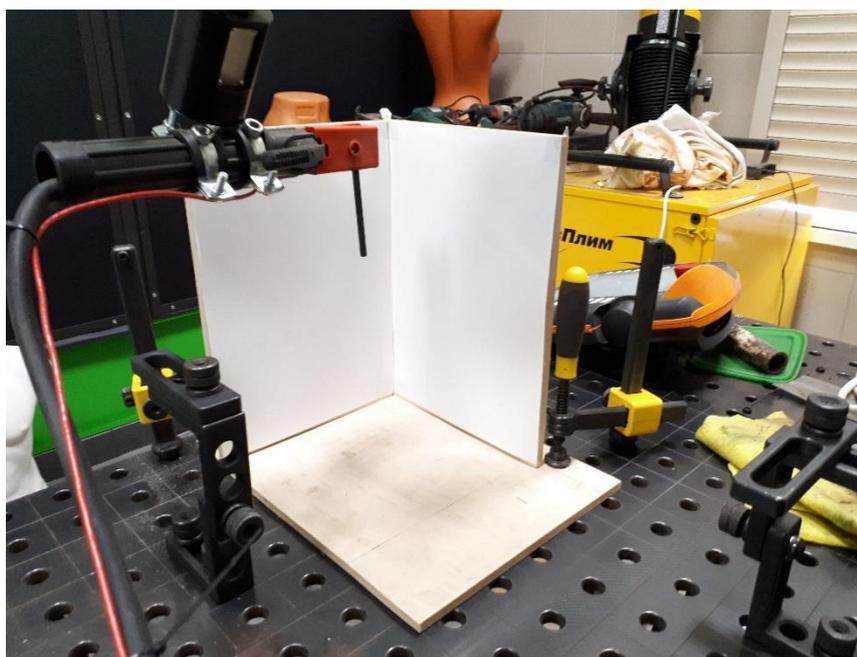


Рисунок 2.8 – Собранная конструкция из фанеры

Кроме этого напротив каждой из фанер закреплен Г образный металлический крепеж. Они нужны для устойчивости камер Microsoft «LifeCam Cinema». Расположение камер изображены на рисунке 2.10. Сами камеры очень чувствительны к внешним воздействиям. На рисунке 2.11 представлена конструкция с 2 камерами.

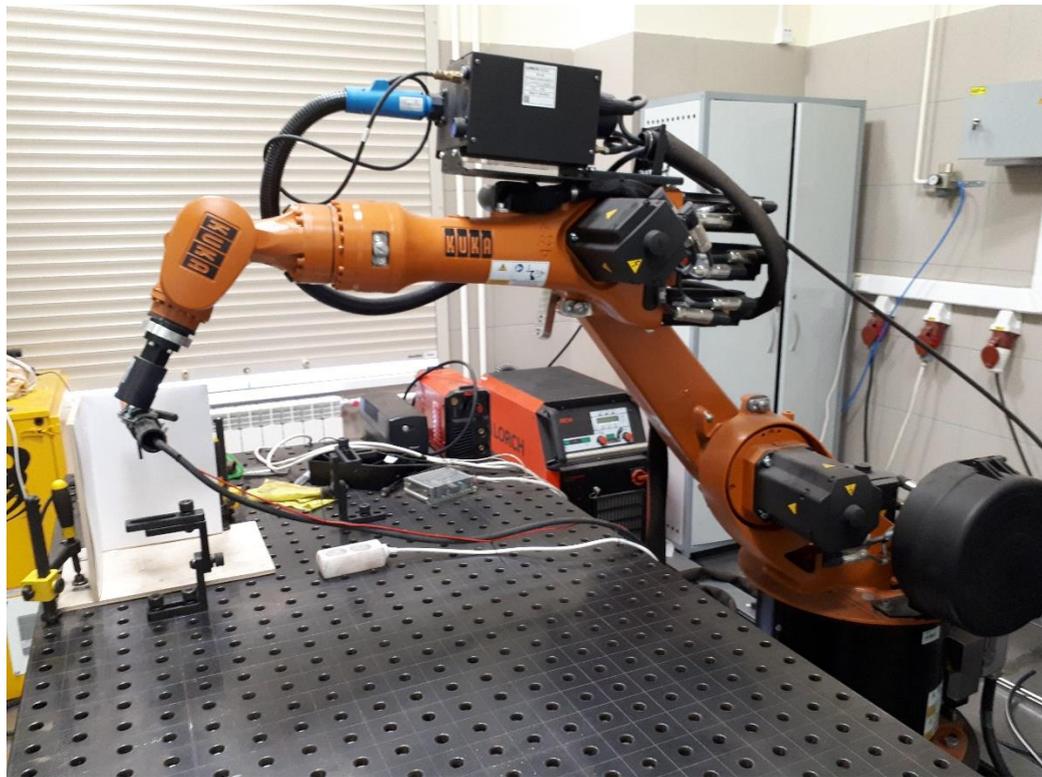


Рисунок 2.9 – Вид конструкции на фоне сварочного манипулятора KUKA KR – 16 – 2

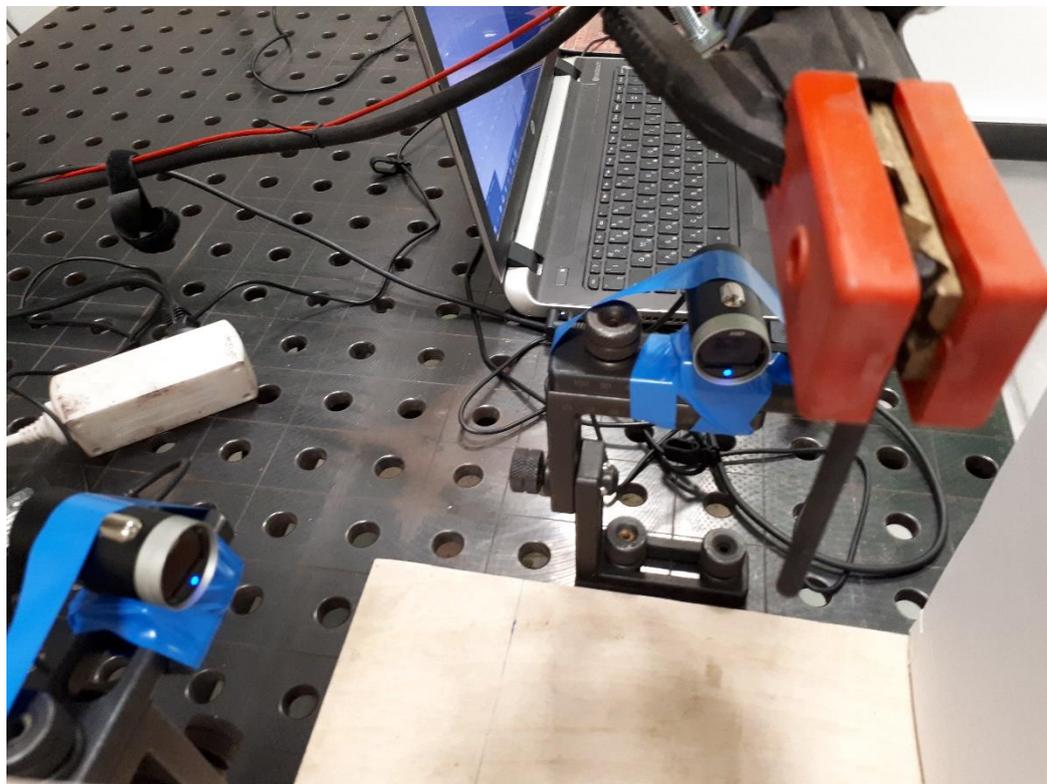


Рисунок 2.10 – Камеры Microsoft, установленные на креплениях

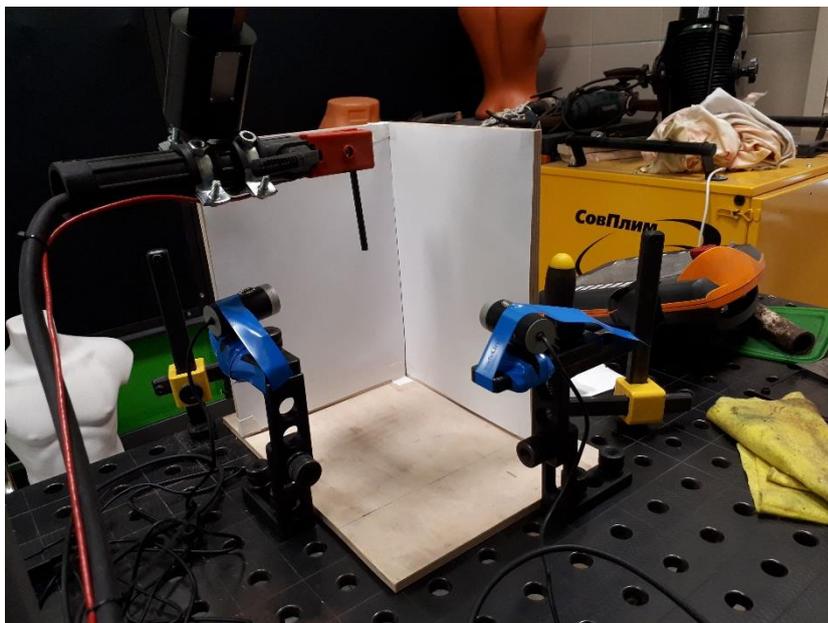


Рисунок 2.11 – Конструкция из фанеры с камерами

Так же на рисунке 2.12 виден электрод сварочного манипулятора. Сам электрод устанавливается вручную, задавая движение манипулятора с помощью пульта управления. Кроме этого на рисунке видны листки А4, закреплённые у каждой из сторон фанер. Как говорилось ранее они нужны для более стабильной работы камер, так как в самой кабинке освещение меняется в течение дня. Из-за этого появляются тени, блики на самом изображении, которые очень тяжело решить. Одним из решений конечно является постоянное освещение в виде прожектора, но к сожалению, такая техника затратная. Листы А4 белого цвета, что позволяет упрощать поиск объектов, в нашем случае электрода.



Рисунок 2.12 – Электрод сварочного манипулятора

Для более наглядного вида манипулятора на фоне собранной конструкции с закрепленными камерами и рабочей зоной представлен рисунок 2.13. В ходе работы начальное положение электрода было задано вручную. После шла работа с настройкой камер и их калибровка. В ходе работы манипулятор был выключен для безопасной работы. Как видно на рисунке 2.13 камеры подключены к ноутбуку. В самом ноутбуке производится наладка программной части на языке C++. Данные, полученные с камер сохраняются в корневой папке. После каждого снимка старое изображение удаляется и заменяется новым снимком электрода. Параллельно можно изменять положения камер для удобства снимков. В случае если не получается настроить по размеру изображения, полученные с камеры, их можно обрезать с помощью встроенных команд библиотеки OpenCV.

Работа программной части начинается после всех настроек камер, а также после установки камер в неподвижное положение. В случае невыполнения настроек программная часть будет иметь погрешности при нахождении контуров и координат электрода.

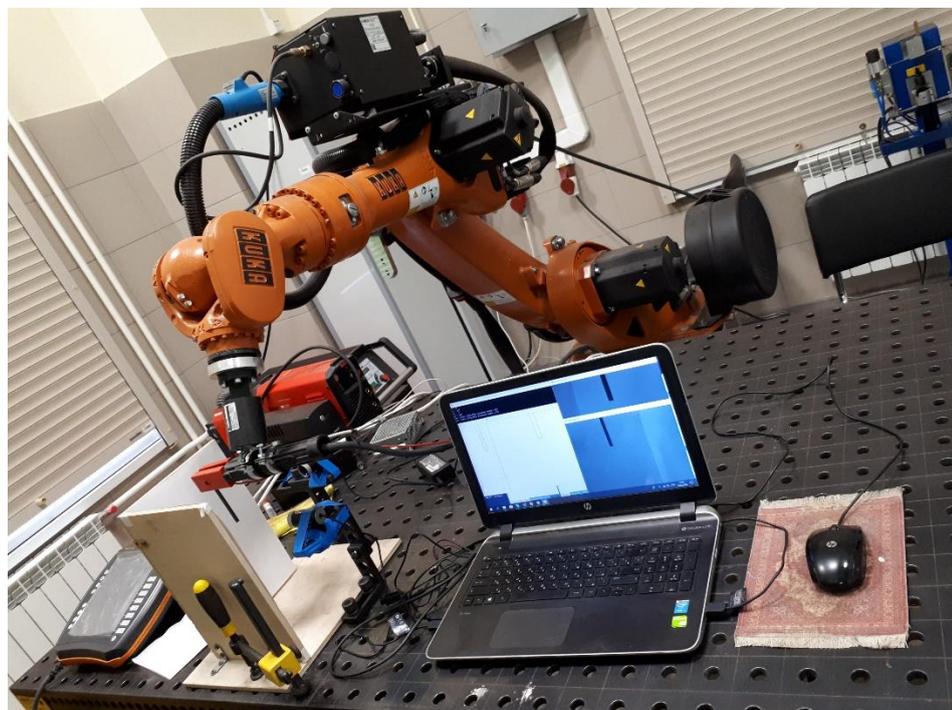


Рисунок 2.13 – Вид манипулятора на фоне конструкции

3. АНАЛИЗ БИБЛИОТЕКИ OPENCV

3.1 Знакомство с библиотекой OpenCV

OpenCV - библиотека компьютерного зрения с открытым исходным кодом. Библиотека написана на C и C++ и работает на компьютерах под управлением Linux, Windows, Mac OS X. Так же активно развиваются интерфейсы библиотеки для Python, Ruby, Matlab и других языков программирования.

Библиотека OpenCV была разработана с целью повышения вычислительной эффективности и с уклоном на приложения реального времени. OpenCV написана с использованием оптимизированного C и может использовать многоядерные процессоры. В дальнейшем, если потребуется автоматическая оптимизация на аппаратных платформах Intel, существует возможность покупки библиотеки IPP (Integrated Performance Primitives), которая состоит из процедур с низкоуровневой оптимизацией для различных алгоритмических областей. OpenCV будет автоматически использовать библиотеку IPP во время выполнения программы [14].

Одной из основных целей OpenCV является предоставление простого в использовании интерфейса, который позволит людям довольно таки быстро строить сложные приложения, использующие компьютерное зрение. Библиотека OpenCV содержит более 500 функций, которые охватывают многие области компьютерного зрения, такие как: инспекция фабричной продукции, медицина, безопасность, пользовательский интерфейс, калибровка камеры, стереозрение и робототехника. И все это благодаря тому, что компьютерное зрение и машинное обучение часто идут "рука об руку", к тому же OpenCV полностью включает в себя библиотеку общего назначения MLL (Machine Learning Library). MLL библиотека ориентирована на распознавание статических образов и кластеризацию. MLL очень полезна для задач компьютерного зрения, которые составляют основу OpenCV, но она довольно-таки обобщенная, чтобы решать конкретные

проблемы машинного обучения [22]. Архитектура самой OpenCV изображена на рисунке 3.1.

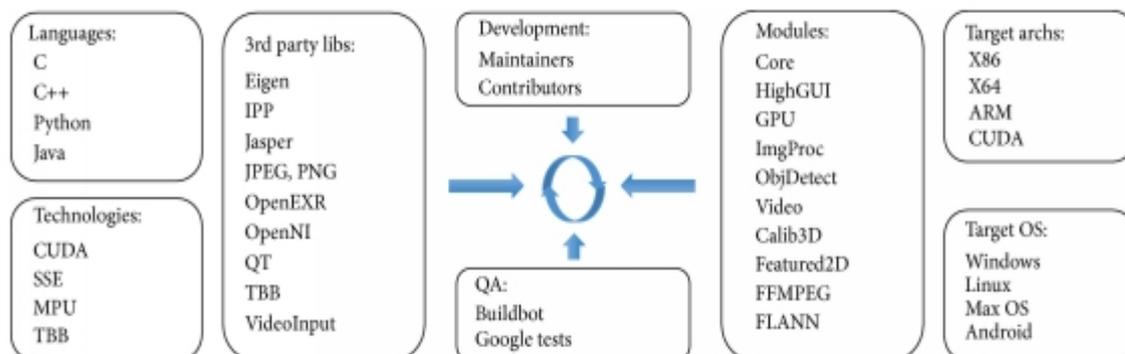


Рисунок 3.1 - Архитектура OpenCV

Основные модули библиотеки можно отнести к 4 группам (разделам):

- Модули Core, HighGUI, реализующие базовую функциональность (базовые структуры, математические функции, генераторы случайных чисел, линейная алгебра, быстрое преобразование Фурье, ввод/вывод изображений и видео, ввод/вывод в форматах XML, YAML и др.).

- Модули ImgProc, Features2D для обработки изображений (фильтрация, геометрические преобразования, преобразование цветовых пространств, сегментация, обнаружение особых точек и ребер, контурный анализ и др.).

- Модули Video, ObjDetect, Calib3D (калибровка камеры, анализ движения и отслеживание объектов, вычисление положения в пространстве, построение карты глубины, детектирование объектов, оптический поток).

- Модуль ML, реализующий алгоритмы машинного обучения (метод ближайших соседей, наивный байесовский классификатор, деревья решений, бустинг, градиентный бустинг деревьев решений, случайный лес, машина опорных векторов, нейронные сети и др.). На рисунке 3.2 представлена общая схема типичного приложения, предназначенного для решения той или иной задачи компьютерного зрения. В дальнейшем разберем некоторые возможности самой библиотеки [14].

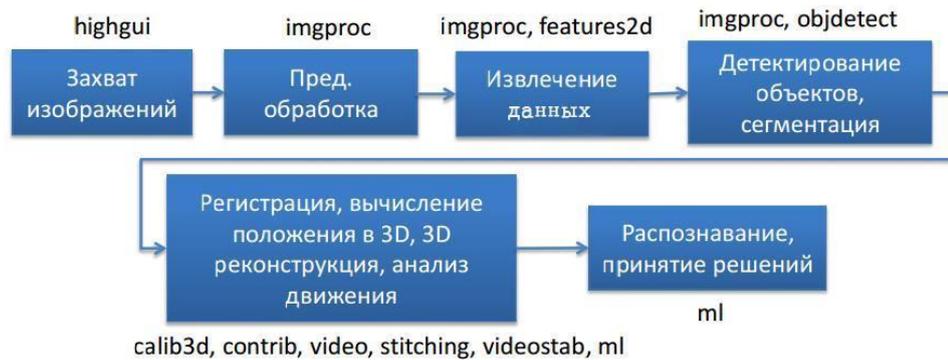


Рисунок 3.2 - Архитектура OpenCV

OpenCV позволяет читать разные типы изображения, а также видео с камеры. Все это возможно благодаря компоненту HighGUI, который включен в OpenCV. В дальнейшем часть этой программы будет использована для открытия изображения [13]. Код для загрузки изображения и отображение его на экране (рисунок 3.3):

```

#include "opencv\highgui.h"
int main( int argc, char** argv )
{
  IplImage* img = cvLoadImage( argv[1] ); // Имя изображения
  cvNamedWindow( "electrode", CV_WINDOW_AUTOSIZE ); // Создаем окно со
  стандартным размером
  cvShowImage( "electrode", img ); // Изображение показывается
  cvWaitKey(0); // Ожидание
  cvReleaseImage( &img ); // Освобождение памяти
  cvDestroyWindow( "electrode" ); // Удаление окна
}
  
```



Рисунок 3.3 – Изображение электрода

Данная программа просто загружает изображение. Теперь попробуем посмотреть работу программы с камерой. Код для включения камеры и снимка электрода (рисунок 3.4) [6]:

```
//загрузка библиотек
#include <opencv2\opencv.hpp>
#include<locale.h>
using namespace cv;
using namespace std;
int main(int, char**)
{
    // снимок с камеры
    vector<int> wr_params;
    wr_params.push_back(cv::IMWRITE_JPEG_QUALITY);
    wr_params.push_back(50);
    VideoCapture cap(2);
    if (!cap.isOpened())
        return -1;
    Mat edges;
    Mat frame;
    cap >> frame;
    //сохранение снимка
    imwrite("electrode_left.jpg", frame, wr_params);
    //Показ изображения
    imshow("electrode_left", frame);
}
```

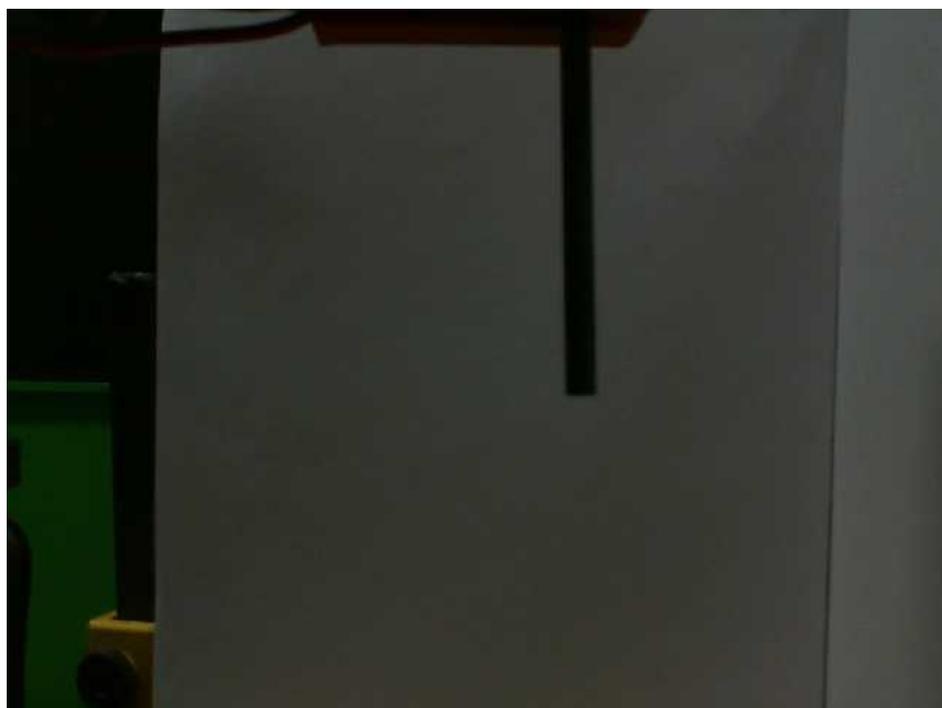


Рисунок 3.4 – Снимок электрода камерой

В ходе изучения работы с библиотекой были реализованы разные варианты преобразования изображения. Одним из вариантов была срезка части изображения. Если изображение имел лишний фон, то его можно было обрезать. В итоге оставалось только рабочая зона. Код обрезки изображения представленный ниже является продолжением предыдущего:

```
Mat img0 = imread("electrode_left.jpg", 1);
int x = 325,
y = 25,
width = 200,
height = 455;
img0 = img0(Rect(x, y, width, height));
imshow("electrode_left", frame);
```

Данный код позволяет обрезать изображение вдоль координаты, которую выбираешь. Обрезка происходит по высоте и ширине изображения. Далее полученное изображение пересохраняется, а старое заменяется обрезанным. Обрезанное изображение электрода рисунка 3.4 показан на рисунке 3.5 [9].



Рисунок 3.5 – Обрезанный снимок электрода

3.2 Калибровка камеры

Перед началом использования, камера должна быть откалибрована. Калибровка позволяет учесть искажения, вносимые оптической системой.

Калибровка камеры сводится к получению внутренних и внешних параметров камеры по имеющимся фотографиям или видео, полученным с ее помощью. Калибровка камеры обычно производится на начальном этапе решения многих задач компьютерного зрения. Кроме того, данная процедура позволяет исправить дисторсию на фотографиях и видео.

Как правило, для представления 2D-координат точки на плоскости используется вектор-столбец вида $[u \ v \ 1]^T$, а для задания положения 3D-точки в мировых координатах – $[x_w \ y_w \ z_w \ 1]^T$. Следует отметить, что эти выражения записаны в расширенной нотации однородных координат. В частности, в модели камеры-обскуры матрица камеры используется для проецирования точек трёхмерного пространства на плоскость изображения[15]

$$z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = A \begin{bmatrix} R & T \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}$$

где Z_c - произвольный масштабный коэффициент

Параметры внутренней калибровки[15]

$$A = \begin{bmatrix} \alpha_x & \gamma & u_0 \\ 0 & \alpha_y & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

Матрица внутренней калибровки A содержит 5 значимых параметров. Эти параметры соответствуют фокусному расстоянию, углу наклона пикселей и принципиальной точке. В частности, α_x и α_y соответствуют фокусному расстоянию, измеренному в ширине и высоте пикселя, u_0 и v_0 – координатам принципиальной точки, а, $\gamma = \alpha_y \cdot \tan\varphi$ где φ - угол наклона пикселя. Нелинейные параметры внутренней калибровки, такие как коэффициенты дисторсии, также имеют важное значение, хотя и не могут быть включены в линейную модель,

описываемую матрицей внутренней калибровки. Большинство современных алгоритмов калибровки камеры определяет их вместе с параметрами линейной части модели. Параметры внутренней калибровки относятся только к камере, но не к сцене, поэтому они изменяются только в том случае, когда меняются соответствующие настройки камеры.

Параметры внешней калибровки R, T (где R - вектор 3×1 или матрица 3×3 поворота, T - вектор 3×1 переноса) определяют преобразование координат точек сцены из мировой в систему координат камеры. Параметры внешней калибровки связаны непосредственно с фотографируемой сценой, поэтому (в отличие от параметров внутренней калибровки) каждой фотографии соответствует свой набор этих параметров. В данной работе будет реализована только калибровка внутренних параметров камеры.

При съемке свет со сцены фокусируется и захватывается камерой. Этот процесс уменьшает число измерений у данных, получаемых камерой, с трёх до двух (свет из трёхмерной сцены преобразуется в двухмерное изображение). Поэтому каждый пиксель на полученном изображении соответствует лучу света исходной сцены, соответственно во время калибровки камеры происходит поиск соответствия между трёхмерными точками сцены и пикселями изображения.

В случае идеальной камеры-обскуры для задания такого соответствия достаточно одной матрицы проекции. Однако в случае более сложных камер искажения, вносимые линзами, могут сильно повлиять на результат. Таким образом, функция проецирования принимает более сложный вид и часто записывается как последовательность преобразований, например:

$$x = I \cdot \text{Dist}(E \cdot X),$$

где $X = [x_w \quad y_w \quad z_w \quad 1]^T$ – координаты исходной точки сцены;

$x = [u \quad v \quad 1]$ – координаты пикселя на изображении;

$E = \begin{bmatrix} R & T \\ 0_3^T & 1 \end{bmatrix}$ – матрица внешней калибровки (где R – матрица поворота 3×3 , T -вектор переноса 3×1 ;

Dist – функция применения дисторсии;

$$I = \begin{bmatrix} \alpha_x & \gamma & u_0 \\ 0 & \alpha_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \text{ – матрица внутренней калибровки.}$$

Существует несколько различных подходов к решению задачи калибровки.

1. Классический подход — алгоритм Roger Y. Tsai [16], состоящий из двух этапов, на первом определяются параметры внешней калибровки, на втором — внутренней калибровки и дисторсии.

2. «Гибкая калибровка», предложена в работе [17] и основана на использовании плоского калибровочного объекта в виде шахматной доски (рисунок 3.5).

3. Автоматическая калибровка — получение калибровочных данных непосредственно по изображениям, без использования специальных калибровочных объектов [18].

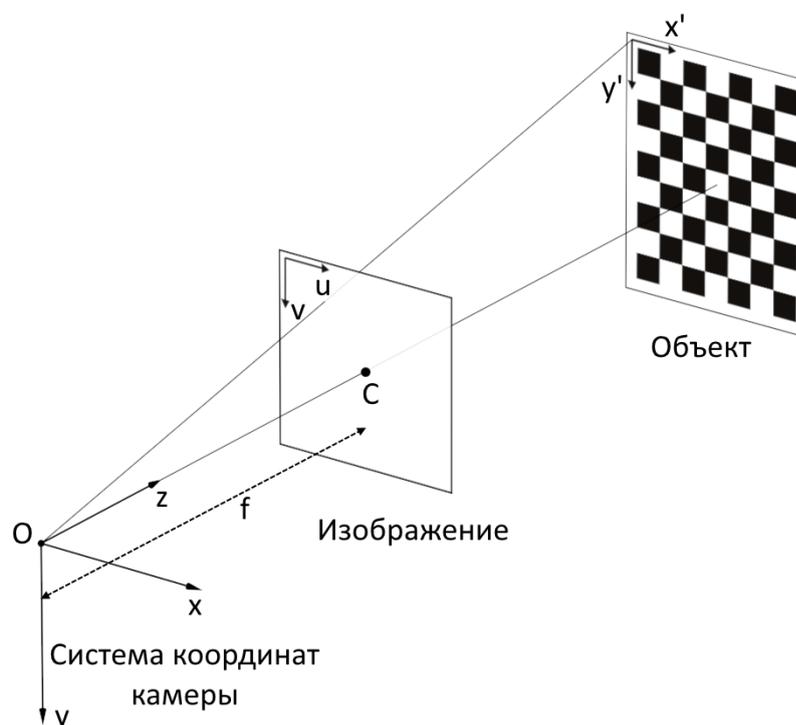


Рисунок 3.6 – Калибровка на основе шахматной доски

В данной работе используется «гибкая калибровка», в силу простоты реализации. Использование шахматной доски возможно благодаря тому, что известно, какие именно у нее точки – можно посмотреть, в какие точки они перешли на изображении.

Если много раз показать этот шаблон камере, то можно увидеть, куда проецируются точки, следовательно, можно установить эти параметры.

В результате возникает задача оптимизации. Известны трехмерные координаты углов (можно измерить размеры квадрата и записать трехмерные координаты шахматной доски), можно протектировать эти уголки на изображении – это стандартная задача. Таким образом, будет соответствие между трехмерной точкой и двухмерной точкой. И нужно будет найти такие параметры, чтобы трехмерные точки переходили именно в эти двухмерные точки. То есть нужно оптимизировать параметры камеры так, чтобы получалась правильная проекция. Для калибровки нужно показывать доску под разными углами (рисунок 3.7) ее можно поворачивать, ее нужно показать во всех частях камеры для того, чтобы надежно установить ее параметры. Потому что если показывать шаблон в одной и той же позиции в одном углу, то хорошо будет известно, как камера проецирует этот угол, но во всей остальной части изображения все может быть очень плохо.

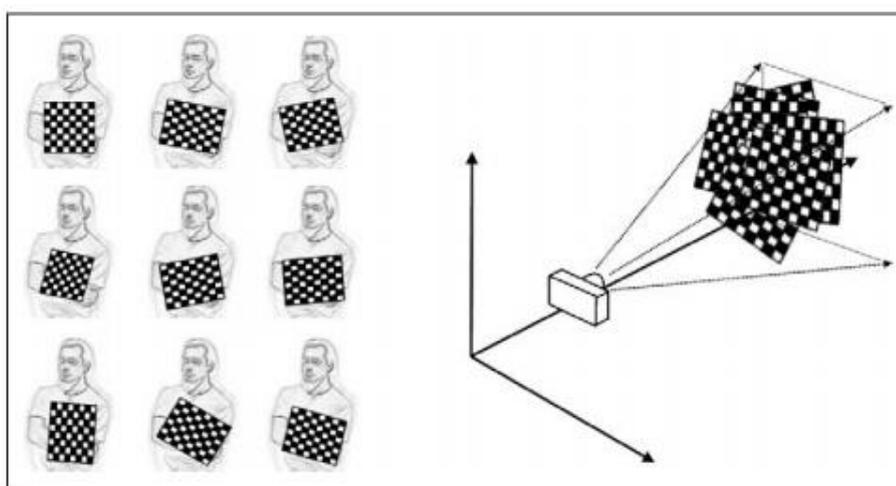


Рисунок 3.7 – Процесс калибровки камеры

Помимо шахматной доски можно использовать и другие шаблоны – например, шаблон из нарисованных кружков, он показывает более точные результаты, потому что центра круга можно найти с большей точностью, чем угол шахматной доски. Соответственно, калибрация получается более точной. Можно использовать более сложные шаблоны, например, трехмерные.

В обычных условиях достаточно одной калибровки для одной камеры. Производители камер предоставляют технические данные. Проблема в том, что для одной модели эти параметры будут немножко отличаться. Можно предоставить какие-то параметры по умолчанию и такие камеры существуют, для которых они посчитаны – для модели. Но для конкретной камеры данной модели они могут немного отличаться. Потому что если бы они не отличались, это означало бы, что матрица расположена абсолютно на одном и том же месте, абсолютно такие же линзы. На практике такое невозможно. Всегда будут небольшие отклонения, следовательно, параметры калибровки тоже будут немного другими. Поэтому для более точных параметров необходимо делать калибровку самому. Можно пытаться делать автокалибровку – снимать камерой обычное изображение – без шаблонов шахматной доски, и при этом пытаться понять, какие у нее внутренние параметры. Автоматически, без шаблонов, пытаться понять, как она искажает. Но на практике получается, что иногда это работает, иногда это не работает. В алгоритме автокалибровки есть несколько параметров – эти параметры нужно подбирать для того, чтобы запустить алгоритм, потому что при одних параметрах алгоритм работает хорошо, при других работает плохо. В ситуации же с использованием шаблонов с гораздо большей надежностью можно получить хорошую калибровку камеры.

Как было сказано ранее камеры-обскуры приводят к большим искажениям изображений. Двумя основными искажениями являются радиальные искажения(а) и тангенциальные искажения(в) (рисунок 3.8).

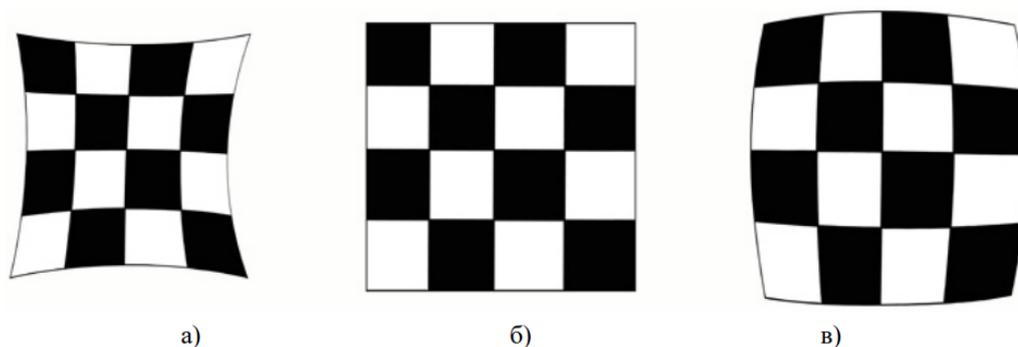


Рисунок 3.8 – Изображение шахматной доски а) тангенциальное искажение
 б) неискаженное в) радиальное искажение

Из-за радиального искажения прямые линии будут изогнутыми. Этот эффект будет увеличиваться с удалением от центра изображения. Например, на рисунке 3.9, где два края шахматной доски отмечены красными линиями. Но можно увидеть, что граница не является прямой и не соответствует красной строке. Все ожидаемо прямые линии изогнуты [19].

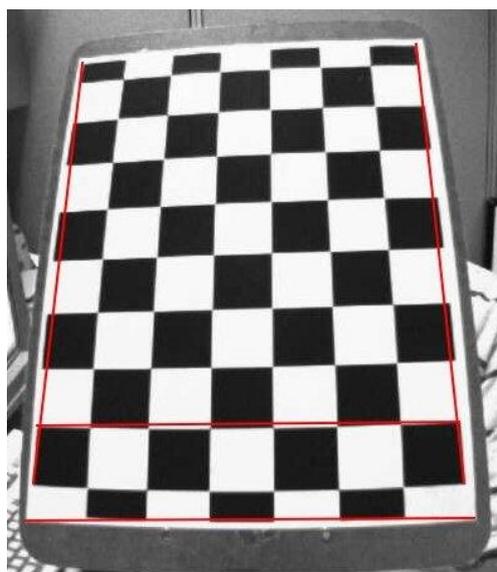


Рисунок 3.9 - Радиальное искажение на шахматной доске

Это искажение решается следующим образом:

$$X_{corrected} = x(1 + k_1 \cdot r^2 + k_2 \cdot r^4 + k_3 \cdot r^6)$$

$$Y_{corrected} = y(1 + k_1 \cdot r^2 + k_2 \cdot r^4 + k_3 \cdot r^6)$$

Другим искажением является тангенциальное искажение, которое происходит, потому что съемка изображения не выравнивается идеально параллельно

плоскости изображения. Поэтому некоторые области изображения могут выглядеть ближе, чем ожидалось. Оно решается следующим образом:

$$X_{corrected} = x + [2p_1 \cdot x \cdot y + p_2 \cdot (r^2 + 2x^2)]$$

$$Y_{corrected} = yx + [2p_2 \cdot x \cdot y + p_1 \cdot (r^2 + 2y^2)]$$

В итоге, нужно найти пять параметров, известных как коэффициенты искажения, определяемые:

$$Distortion\ coefficients = (k_1, k_2, p_1, p_2, k_3)$$

В дополнение к этому нам нужно найти еще некоторую информацию, например, внутренние и внешние параметры камеры. Внутренние параметры специфичны для камеры. Они включают информацию, такую как фокусное расстояние $f_x f_y$, оптические центры $c_x c_y$ и т. д. Их также называют матрицей камеры. Это зависит только от камеры, поэтому после вычисления она может храниться для будущих целей. Она выражается как матрица 3x3:

$$I = \begin{bmatrix} \alpha_x & \gamma & u_0 \\ 0 & \alpha_y & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

Внешние параметры соответствуют векторам вращения и трансляции, которые переводят координаты трехмерной точки в систему координат.

Для того чтобы корректно производить измерения эти искажения необходимо сначала скорректировать. Чтобы найти все эти параметры, нужно предоставить несколько образцовых изображений четко определенного шаблона (например, шахматной доски).

Как упомянуто выше, нам нужно как минимум несколько тестовых шаблонов для калибровки камеры. Для достижения лучших результатов, как советует сайт разработчиков библиотеки OpenCV необходимо сделать 10 тестовых шаблонов. В результате для камеры №1 было сделано пятнадцать шаблонов (рисунок 3.10), а для камеры №2 десять (рисунок 3.11).

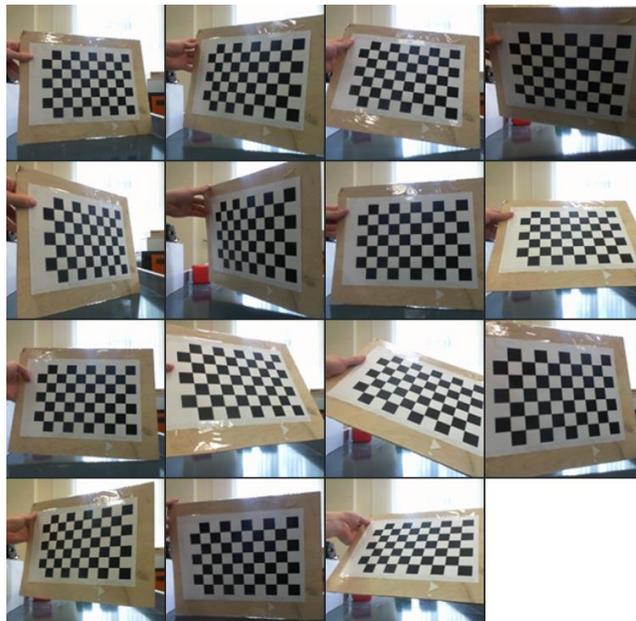


Рисунок 3.10 – Тестовые шаблоны для камеры №1

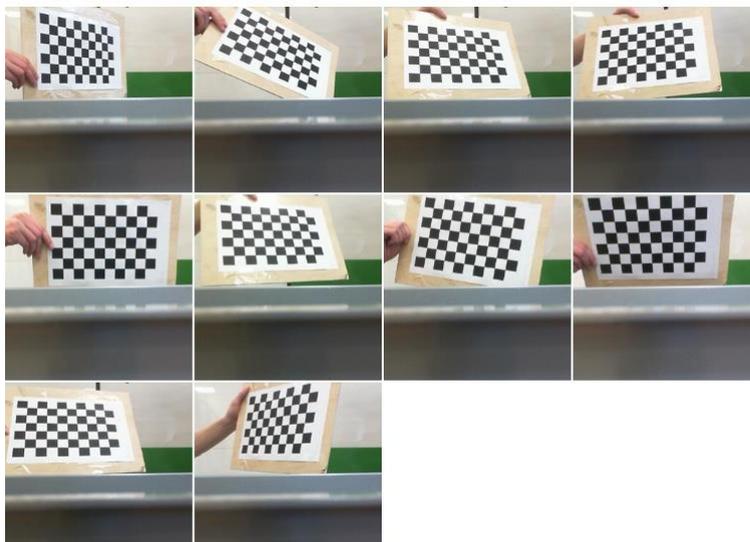


Рисунок 3.11 – Тестовые шаблоны для камеры №2

Важными входными данными, необходимыми для калибровки камеры, являются набор трехмерных точек реального мира и соответствующие им точки 2D-изображения. 3D-точки называются объектными точками, а точки 2D-изображения называются точками изображения.

Итак, чтобы найти шаблон на шахматной доске, мы используем функцию `cv2.findChessboardCorners()`. Нам также нужно задать, какой тип шахматной сетки используется, в данной работе используется сетка размерностью 6x9.

(обычно шахматная доска имеет 8x8 квадратов и 7x7 внутренних углов). Данная функция возвращает угловые точки, если шаблон был успешно найден. Эти углы будут размещены в порядке (слева направо, сверху вниз). Эта функция может оказаться не в состоянии найти требуемый шаблон во всех изображениях. Поэтому необходимо поочередно проверять каждый шаблон. Как только углы найдены, можно повысить точность их отображения, используя `cv2.cornerSubPix()`. После этого можем нарисовать их на шаблоне, используя `cv2.drawChessboardCorners()`. Все эти шаги включены в код ниже [20]:

```
#include <iostream>
#include <cstdlib>
#include <opencv2/core/core.hpp>
#include <opencv2/imgproc/imgproc.hpp>
#include <opencv2/highgui/highgui.hpp>
cv::VideoCapture m_camera;
cv::Mat m_currentFrame;
cv::Size m_cameraResolution;
std::vector<std::vector<cv::Point2f> > m_detectedPoints;
cv::Mat m_cameraMatrix( 3, 3, CV_64FC1 );
cv::Mat m_distortionCoefficients( 5,1,CV_64FC1 );
cv::Mat m_R;
cv::Mat m_T;
cv::Mat m_grayImage;
std::vector<std::vector<cv::Point3f> > m_realPoints;
size_t m_framesGrabed( 0 );
const cv::Size patternSize( 7,7 );
m_camera.open( CV_CAP_ANY );
// сохраняем разрешение камеры
m_cameraResolution = cv::Size( ( int ) m_camera.get( CV_CAP_PROP_FRAME_WIDTH ), ( int ) m_camera.get( CV_CAP_PROP_FRAME_HEIGHT ) );
// получаем текущую картинку с камеры
m_camera >> m_currentFrame;
// ищем шахматную доску на изображении
if( cv::findChessboardCorners( m_currentFrame, patternSize, m_detectedPoints[m_framesGrabed] ) ) {
    cv::cvtColor( m_currentFrame, m_grayImage, CV_BGR2GRAY );
    // проверяем координаты углов доски
    cv::cornerSubPix( m_grayImage, m_detectedPoints[m_framesGrabed],
                    cv::Size( 11,11 ), cv::Size( -1, -1 ), cv::TermCriteria( cv::TermCriteria::EPS | cv::TermCriteria::MAX_ITER, 30, 0.1 ) );
    cv::drawChessboardCorners( m_currentFrame, patternSize, m_detectedPoints[m_framesGrabed], true );
    ++ m_framesGrabed;
}
std::vector<cv::Mat> rvecs( m_realPoints.size() );
std::vector<cv::Mat> tvecs( m_realPoints.size() );
m_distortionCoefficients = cv::Scalar( 0 );
m_cameraMatrix=cv::Scalar( 0 );
int flags = cv::CALIB_FIX_K3;
// начальные параметры камеры
m_cameraMatrix.at( 0, 0 ) = m_initialFocusValue;
m_cameraMatrix.at( 0, 2 ) = m_cameraResolution.width >> 1;
```

```

m_cameraMatrix.at( 1, 1 ) = m_initialFocusValue;
m_cameraMatrix.at( 1, 2 ) = m_cameraResolution.height >> 1;
m_cameraMatrix.at( 2, 2 ) = 1.0;
flags |= cv::CALIB_USE_INTRINSIC_GUESS;
// выполняем калибровку
cv::calibrateCamera( m_realPoints, m_detectedPoints, m_cameraResolution, m_cameraMatrix, m_distortionCoefficients, rvecs, tvecs, flags );
m_R = rvecs[0];
m_T = tvecs[0];

```

Один из используемых шаблонов, с успешно найденными углами показан на рисунке 3.12.



Рисунок 3.12 – Калибровочный шаблон с найденными углами

Итак, теперь у нас есть точки объекта и точки изображения, которые были необходимы для калибровки камеры. В библиотеке OpenCV для калибровки используется функция `cv2.calibrateCamera()`. Она возвращает матрицу камеры, коэффициенты искажения, векторы вращения и трансляции и т. д.

```

cv::calibrateCamera( m_realPoints, m_detectedPoints, m_cameraResolution,
m_cameraMatrix, m_distortionCoefficients, rvecs, tvecs, flags );

```

После получение параметров калибровки, можно получить неискаженное изображение. Но до этого сначала нужно уточнить матрицу камеры на основе свободного параметра масштабирования, используя `cv2.getOptimalNewCameraMatrix ()`.

Функция вычисляет и возвращает оптимальную новую матрицу камеры на основе параметра свободного масштабирования(α). Изменяя этот параметр, можно получить только чувствительные пиксели $\alpha = 0$, сохранить все исходные пиксели изображения, если есть ценная информация в углах $\alpha = 1$ или получить что-то среднее между ними. Когда $\alpha > 0$, неискаженный результат, вероятно, будет иметь некоторые черные пиксели, соответствующие «виртуальным» пикселям за пределами захваченного искаженного изображения [21].

Одно из полученных неискаженных изображений представлено на рисунке на рисунке 3.13



Рисунок 3.13 - Неискаженное изображение шахматной доски

Как показано на рисунке выше оба метода дают одинаковый результат. В итоге можно видеть, что все грани прямые.

Точность измерения параметров камеры (коэффициенты дисторсии, матрица камеры) определяется средней величиной ошибки перепроецирования (ReEr, Reprojection Error). Она изображена на рисунке 3.14 и представляет собой расстояние (в пикселях) между проекцией P' на плоскость изображения точки P на поверхности объекта, и проекцией P'' этой же точки P , построенной после устранения дисторсии с использованием параметров камеры.

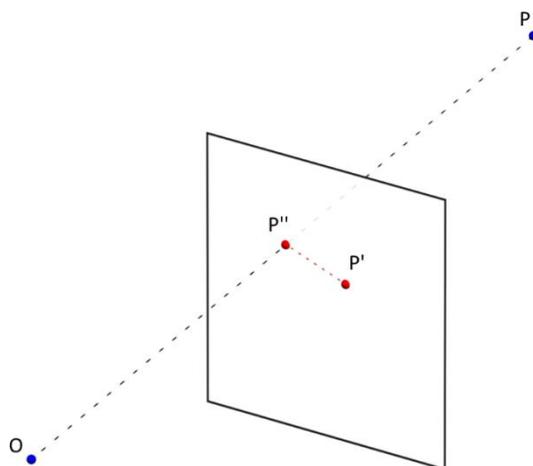


Рисунок 3.14 – Представление параметров камеры в виде плоскостей

Данная ошибка должно быть как можно ближе к нулю. Учитывая матрицы внутреннего, искажения, вращения и трансляции, сначала нужно преобразовать точку объекта в точку изображения, используя `cv2.projectPoints()`. Затем мы вычисляем абсолютную норму между тем, что мы получили с нашим преобразованием и алгоритмом поиска углов. Чтобы найти среднюю ошибку, мы вычисляем среднее арифметическое ошибок для всех калибровочных изображений.

Теперь можно хранить матрицу камеры и коэффициенты искажения, используя функции записи в Numpy (`np.savez`, `np.savetxt` и т. Д.) Для использования их в дальнейшем. Это означает что внутренние параметры камеры не нужно каждый раз заново пересчитывать, для определенной модели они постоянны.

3.3 Определение контуров с помощью детектора границ Canny

Детектор границ Canny назван в честь его изобретателя Джона Ф. Канни, который придумал алгоритм в 1986 году. Алгоритм гораздо более востребован, чем остальные и считается одним из популярных методов. Основные принципы, которыми управляет детектор Canny, остаются неизменными. Это означает, что мы будем использовать значения градиента в качестве индикаторов того, принадлежит ли пиксель к области потенциального края или нет. Для понимания того, как работает детектор границ Canny приведены следующие шаги [23]:

- неагрессивное подавление: этот шаг выполняется после вычисления величины и направления градиента (G и Θ). Непрерывное подавление в основном используется для тонких контуров. Для каждого пикселя в градиентном изображении мы сравниваем его градиентную величину с малой площадью пикселей, которые расположены в том же направлении, что и градиент. Если градиентная величина пикселя действительно самая высокая среди всех таких соседей, значение сохраняется; в противном случае он подавляется. Этот шаг необходим, потому что края, которые были выведены только из значений градиента, довольно размыты (толстые) вокруг краевых областей. Таким образом, не максимальное подавление улучшает качество контуров, делая их как можно более тонкими и максимально приближенными к реальной жизни;

- двойной порог: после выполнения не максимального подавления мы используем метод двойного порога для уменьшения эффекта ложных срабатываний. Ложные срабатывания - это те области, которые были обнаружены как контуры, но на самом деле не являются. Это происходит главным образом из-за присутствия шума в изображениях. Мы определяем два пороговых значения: низкий и высокий порог. После определения значений мы будем подвергать каждый пиксель следующим критериям классификации: 1) если величина градиента выше высокого порога, она классифицируется как контурный пиксель и называется сильным контурным пикселем; 2) если величина градиента ниже

нижнего порога, она отбрасывается; 3) если величина лежит между низким и высоким порогом, то пиксель классифицируется как слабый контурный пиксель. Слабые контурные пиксели подчиняются процессу, называемому гистерезисом, где они сохраняются, только если один из их восьми соседей является пикселем с сильным краем; в противном случае они отбрасываются [24].

Теперь, когда мы закончили с объяснением детектора Canny, пришло время увидеть его в действии. Следующий фрагмент кода применяет алгоритм Canny для одноканального изображения электрода (рисунок 3.15) в оттенках серого:

```
// загрузка библиотек
#include <iostream>
#include "opencv2/core/core.hpp"
#include "opencv2/highgui/highgui.hpp"
#include "opencv2/imgproc/imgproc.hpp"
using namespace std;
using namespace cv;
int main()
{
    // считывание изображения электрода
    Mat img0 = imread("electrode.jpg");
    // использование оттенков серого цвета
    cvtColor(img0, img0, CV_BGR2GRAY);
    Mat edges;
    // настройки для определения контура
    Canny(img0, edges, 100, 300, 3, false);
    // показ изображения контура на экране
    imshow("Canny detector", edges);
    waitKey(0);
    return 0;
}
```

Результат работы определения контуров с помощью детектора Canny представлен на рисунке 3.16.



Рисунок 3.15 – Исходное изображение для определения

Как видно из программы сперва загружаются библиотеки для их использования. Далее загружается изображение электрода, который находится в корневой папке OpenCV. С использованием команды `cvtColor` достигаем использование оттенков серого цвета. Команда `Canny` позволяет менять чувствительность определения контура для нахождения контура электрода. Меняя коэффициенты можно получить довольно приемлемый результат определения контура объекта. В конце на экран выводится результат работы с помощью команды `imshow`.



Рисунок 3.16 – Определение контура электрода с детектором Canny

Для того чтобы убедиться, что с помощью данного метода определения контура можно добиться максимального определения, попробуем найти контуры других объектов. К примеру найденные контуры Чеширского кота и карандаша представлены на рисунках 3.17 и 3.18.



Рисунок 3.17 – Определение контура Чеширского кота с детектором Canny

Как видно по результатам если сам объект является большим, имеет несколько геометрических фигур и на заднем фоне изображения имеются фигуры, то детектор Canny ошибочно может их определить. При определении контура карандаша и электрода таких ошибок не замечено. Из этого следует что при работе с однотонным задним фоном и произвольным небольшим объектом данный метод практически не имеет ошибок. В нашем случае для определения контура электрода были созданы все условия для уменьшения ошибок нахождения их.

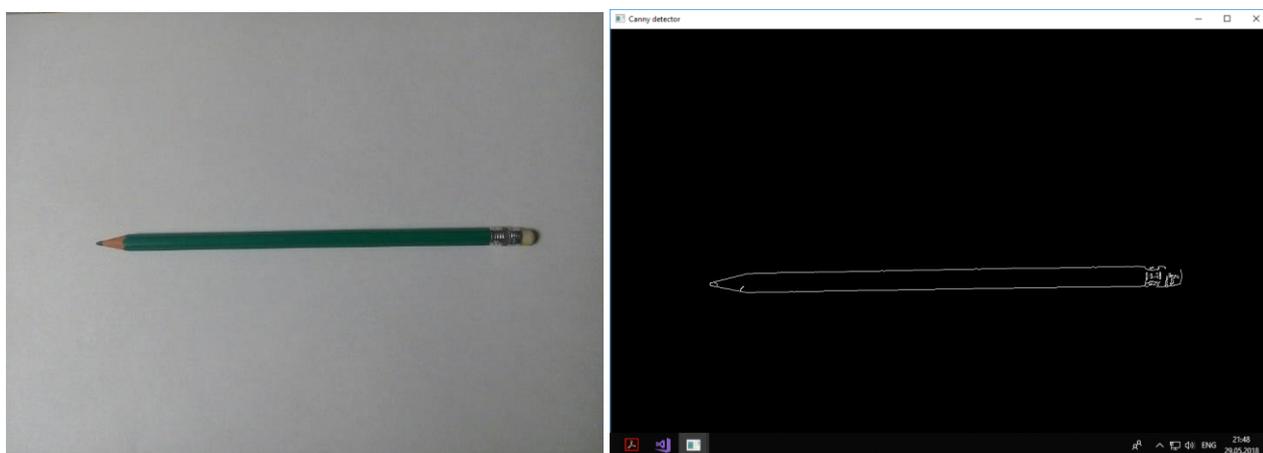


Рисунок 3.18 – Определение контура карандаша с детектором Canny

3.4 Определение контуров с помощью оператора Лапласа

Существует еще один метод, который доступен для определения контуров: оператор Лапласа. Оператор Лапласа - это вторая производная изображения, где вторая производная находится производной от производной. Математически это представляется следующим образом:

$$dst = \frac{d^2src}{dx^2} + \frac{d^2src}{dy^2}$$

Когда ищутся контуры с использованием первой производной, можно заметить, что области, которые являются потенциально контурными областями, имеют достаточно большую величину производной (градиент). Как оказалось, в тех же контурных областях вторая производная равна нулю. Это явление используется в качестве критерия для обнаружения контуров с помощью оператора Лапласа [25].

Функция `Laplacian()` в OpenCV реализует оператор Лапласа, который мы обсуждали. Фактически, один вызов `Laplacian()` будет обрабатывать как размеры `x`, так и `y`. Внутри он вызывает функцию `Sobel()` для вычисления градиентов. Фрагмент кода, показывающий реализацию `Laplacian()`, выглядит следующим образом [27]:

```
// загрузка библиотек
#include <iostream>
#include <cstdlib>
#include <opencv2/core/core.hpp>
#include <opencv2/imgproc/imgproc.hpp>
#include <opencv2/highgui/highgui.hpp>
using namespace std;
using namespace cv;
int main()
{
    Mat input_image = imread("cat.jpg",
    IMREAD_GRAYSCALE); // считывание изображения кота
    Mat output, scaled_output;
    // использование оператора Лапласа
    Laplacian(input_image, output, CV_16S, 3);
    convertScaleAbs(output, scaled_output);
    // показ изображения
    imshow("Laplacian", scaled_output);
}
```

```
waitKey(0);  
return 0;  
}
```

Результат работы нахождения Чеширского кота с помощью оператора Лапласа представлен на рисунке 3.19. Цветное изображение показана на рисунке 3.17.

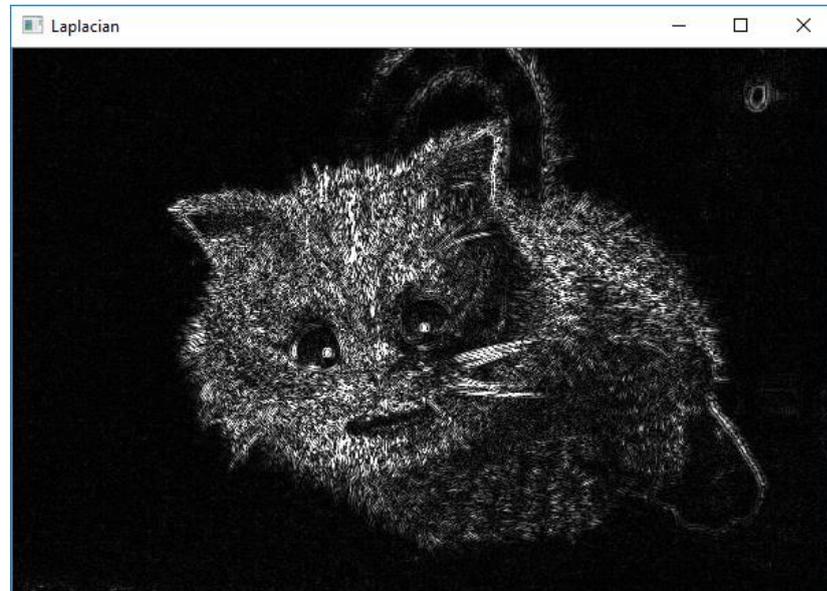


Рисунок 3.19 – Определение контура Чеширского кота оператором Лапласа

Так же ниже будет приведены результат нахождения контура ручки, карандаша и электрода. Результат их работы представлен на рисунках 3.20, 3.21 и 3.22 соответственно.



Рисунок 3.20 – Определение контура ручки оператором Лапласа

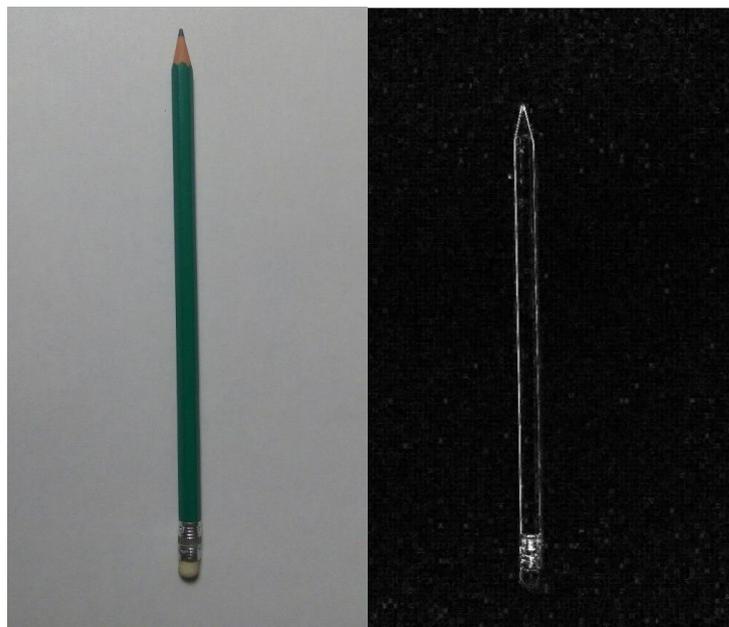


Рисунок 3.21 – Определение контура ручки оператором Лапласа

Основным минусом данного метода является нахождения контура путем изменения изображения. Как вы видите изображения получаются с дополнительными блики, либо белые пятна. В ходе работы с координатами данный метод не очень эффективен.

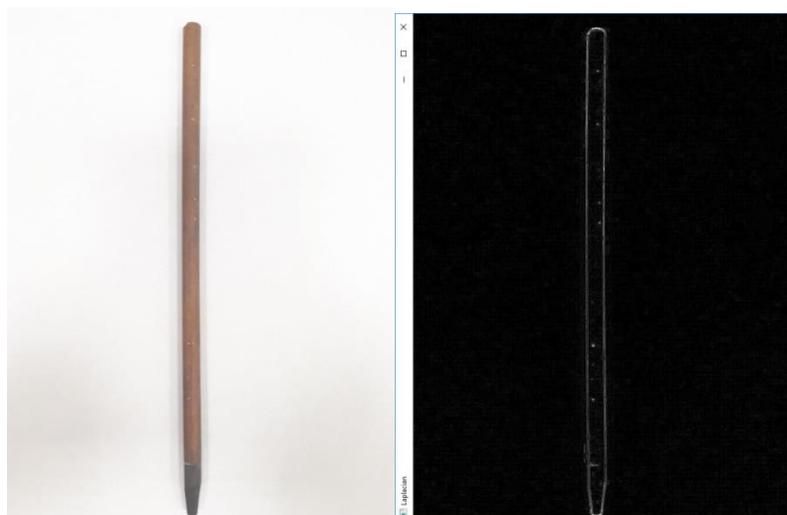


Рисунок 3.22 – Определение контура электрода оператором Лапласа

3.5 Определение контуров с помощью детектора границ Собеля

Самый первый шаг для обнаружения контура, с которой все начинается - это вычисление производных. Производные вдоль направления x и y вычисляются отдельно и хранятся в двух разных матрицах. Таким образом, для каждого

пикселя (x, y) во входном изображении мы имеем по существу градиент в направлениях x и y : G_x и G_y . Из этих двух значений градиентов мы вычисляем так называемую градиентную величину в точках (x, y) . Величина градиента определяется следующей формулой:

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

Это то же самое, что и величина двумерного вектора, который имеет компоненты G_x и G_y . Фактически, производные (или градиенты) в нескольких измерениях часто визуализируются концептуально как двумерные векторы. Кроме того, в дополнение к величине градиент также имеет направление, заданное формулой [26]:

$$\theta = \tan^{-1}\left(\frac{G_y}{G_x}\right)$$

Однако мы будем не будем использовать направление градиента в наших вычислениях на данный момент. Возвращаясь к нашей структуре обнаружения контура, для каждого местоположения пикселя теперь мы имеем величину градиента в этой точке: G . Поскольку G учитывает градиенты обоих направлений, это хорошая количественная оценка количества вариаций в значениях интенсивности пикселей вокруг каждой точки (x, y) .

В нашем определении контуров мы уже заявили, что контур - это те области, где изменение интенсивностей пикселей велико, и мы имеем количественную меру величины этого изменения для каждого пикселя в изображении. Все, что еще предстоит сделать, это выяснить, какие пиксели имеют значение G , которые мы считаем достаточно высокими, чтобы их классифицировали как контуры. И способ, которым мы решаем это, - это сделать функцию `threshold`, как последний шаг нашей рамки обнаружения границ. На рисунке 3.23 представлена блок-схема алгоритма работы детектора Собеля [27].

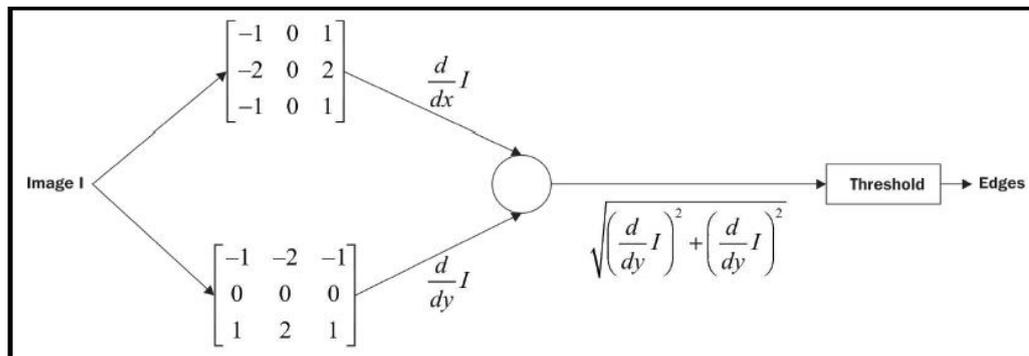


Рисунок 3.23 – Блок-схема алгоритма работы детектора контуров Собеля

Эта блок-схема на самом деле изображает то, что мы называем детектором контуров Собеля. Это связано с тем, что, если вы внимательно заметите, ядра, которые были использованы для вычисления производных, являются ядрами Собеля. Фрагмент кода, реализующая алгоритм работы детектора контуров Собеля, представлен ниже [27]:

```
#include <iostream>
#include <cstdlib>
#include <opencv2/core/core.hpp>
#include <opencv2/imgproc/imgproc.hpp>
#include <opencv2/highgui/highgui.hpp>
using namespace std;
using namespace cv;
// функция градиента getGradientMagnitude
Mat getGradientMagnitude(Mat xGrad, Mat yGrad) {
    CV_Assert((xGrad.rows == yGrad.rows) && (xGrad.cols == yGrad.cols));
    Mat gradient_magnitude(xGrad.rows, xGrad.cols, CV_16S);
    for (int i = 0; i < xGrad.rows; ++i) {
        for (int j = 0; j < xGrad.cols; ++j)
            gradient_magnitude.at<short>(i, j) =
                abs(xGrad.at<short>(i, j)) + abs(yGrad.at<short>(i,
                    j));}
    return gradient_magnitude; }
// функция threshold
Mat thresholdGradientMagnitude(Mat gradient_magnitude, int threshold) {
    Mat edges = Mat::zeros(gradient_magnitude.rows,
        gradient_magnitude.cols, CV_8U);
    for (int i = 0; i < gradient_magnitude.rows; ++i) {
        for (int j = 0; j < gradient_magnitude.cols; ++j) {
            if (gradient_magnitude.at<short>(i, j) >= threshold)
                edges.at<uchar>(i, j) = 255; } }
    return edges; }

int main() //основная часть программы с командой Sobel
{
    Mat input_image = imread("cat.jpg",
        IMREAD_GRAYSCALE);
    Mat x_gradient, y_gradient;
    Sobel(input_image, x_gradient, CV_16S, 1, 0);
    Sobel(input_image, y_gradient, CV_16S, 0, 1);
    Mat gradient_magnitude = getGradientMagnitude(x_gradient, y_gradient);
```

```
Mat edges_output = thresholdGradientMagnitude(gradient_magnitude, 200);
imshow("Edges", edges_output);
waitKey(0);
return 0; }
```

Функция `getGradientMagnitude()` принимает два объекта `Mat`, которые представляют собой градиент x и y нашего входного изображения в качестве параметров, и возвращает другой `Mat`, который содержит величину градиента для каждой точки пикселя. Размеры всех трех объектов `Mat` (два аргумента и возвращаемого) одинаковы и равны размерам исходного изображения.

Функция `CV_Assert()` принимает выражение как аргумент и выдает исключение, если оно оценивается как ложное во время выполнения. Как вы могли догадаться, используя `CV_Assert()`, мы попытались обеспечить тот факт, что два входных аргумента имеют одинаковый размер (который в следующей строке также становится размером `Mat`, который должен быть возвращен). `CV_Assert()` весьма полезен, если вы хотите обеспечить соблюдение таких условий в своем коде. Фактически, эта функция широко используется в большинстве внутренних реализаций библиотеки `OpenCV`. В части функции `getGradientMagnitude()` так же написано вычисление G , которая упрощена от корней до суммы модуля каждого G_x и G_y . Делается это для простоты, так как работать с корнями дает огромные значения, а с суммой G_x и G_y выдает хорошие результаты. Выбранная формула выглядит так [27]:

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

Полная реализация оператора Собеля описана выше. Есть еще один аспект, который можно изменить в коде. Можно запустить этот код с разными значениями пороговых значений. Результаты работы детектора Собеля представлена на рисунках 3.23, 3.24 и 3.25.

В ходе работы с разными методами определения контура был выбран детектор `Canny`. Основным преимуществом является нахождение контура при любом типе освещения. Так же код очень удобен для работы с другими функциями.

Основные недостатки других методов является чувствительность к освещению. Для детектора Собеля написание лишнего кода для вычисления градиентов.

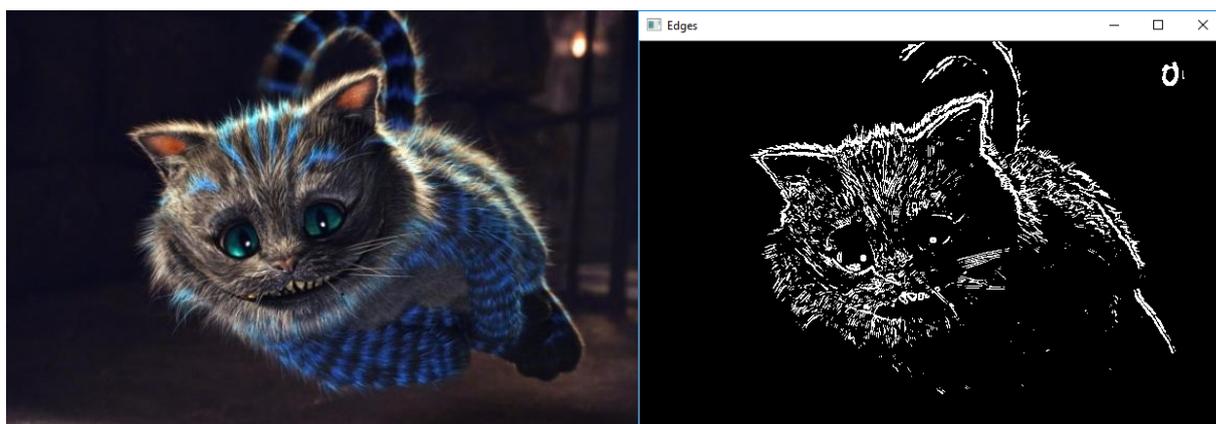


Рисунок 3.23 – Определение контура Чеширского кота с детектором Собеля



Рисунок 3.24 – Определение контура ручки с детектором Собеля



Рисунок 3.25 – Определение контура электрода с детектором Собеля

4. РЕЗУЛЬТАТЫ ПРАКТИЧЕСКОЙ ЧАСТИ

4.1 Реализованный поиск контура электрода

В ходе работы для определения контуров был выбран детектор контуров Canny. Данный метод очень хорошо подходит для работы в помещениях с изменяющимся освещением. Как было сказано ранее для улучшения определения контура электрода были добавлены на задний фон листы А4. В ходе настройки камер в случае получения изображения другого фона, кроме белого, небелый фон вырезался программным путем. Пример рабочей зоны с настроенной камерой и запущенной программой представлен на рисунке 4.1.

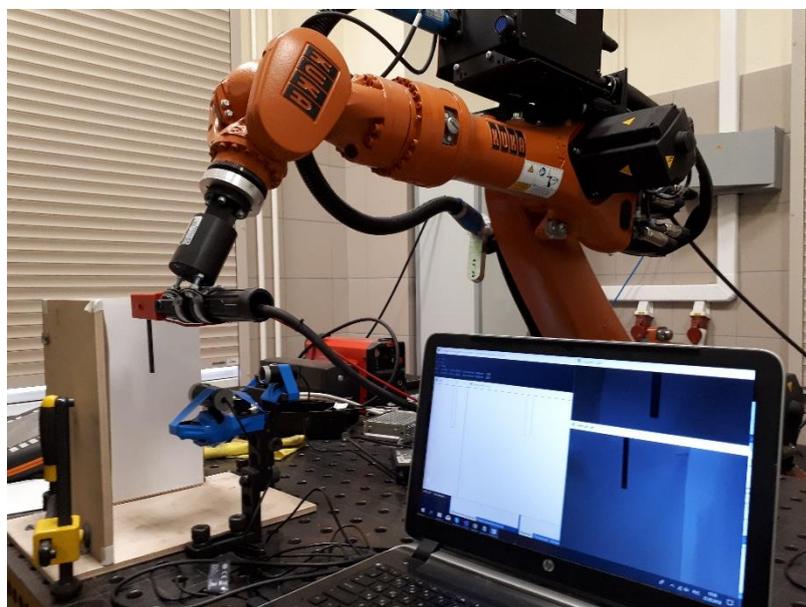


Рисунок 4.1 – Рабочая зона с настроенными камерами и программой

Программный код определения контура представлен ниже:

```
#include <opencv2/opencv.hpp>
using namespace cv;
using namespace std;
int main() {
    // читаем электрод (можно изменить путь в изображению)
    Mat img0 = imread("electrode.jpg", 1);
    Mat img1;
    cvtColor(img0, img1, CV_RGB2GRAY);
    // настройка фильтра
    Canny(img1, img1, 100, 200);
    // нахождение контуров
    vector< vector<Point> > contours;
    findContours(img1, contours, CV_RETR_EXTERNAL, CV_CHAIN_APPROX_NONE);
    // используем mask для выделения электрода
    Mat mask = Mat::zeros(img1.rows, img1.cols, CV_8UC1);
```

```

// CV_FILLED fills the connected components found
drawContours(mask, contours, -1, Scalar(255), CV_FILLED);
// создание нового изображения
Mat crop(img0.rows, img0.cols, CV_8UC3);
// задание цвета заднего фона
crop.setTo(Scalar(0, 0, 0));
img0.copyTo(crop, mask);
normalize(mask.clone(), mask, 0.0, 255.0, CV_MINMAX, CV_8UC1);
// показ изображений на экране
imshow("original", img0);
imshow("mask", mask);
imshow("canny", img1);
waitKey();
return 0;
}

```

Результаты работы программной части для определения контуров электрода представлен на рисунке 4.2 и 4.3. В ходе работы так же использовалась функция `crop.setTo`, позволяющая задать цвет заднего фона.

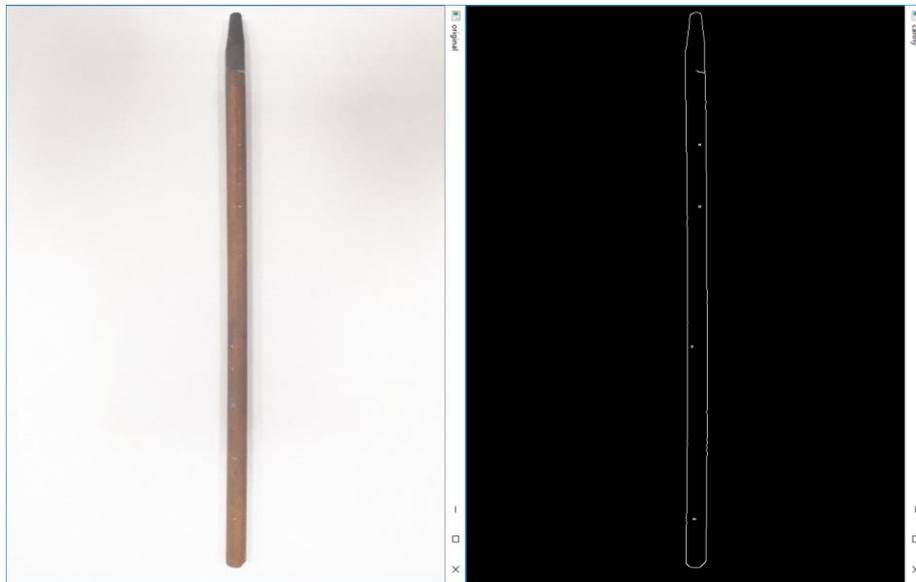


Рисунок 4.2 – Определение контура электрода



Рисунок 4.3 – Выделение белым цветом контура с помощью `mask`

4.2 Программная часть работы двух камер

Перед работой были настроены и откалиброваны обе камеры (рисунок). При сборке конструкции были учтены подвижность камер и подобраны крепления для их устойчивости. Так же при работе с электродом был настроен сварочный манипулятор: была задана оптимальная высота электрода относительно двух камер.

При написании программы полученные снимки сохранялись в корневой папке. Для каждой камеры были заданы имена: `electrode_left` и `electrode_right`. Название подобраны в зависимости от расположения камер. Пример фрагмента кода работы обеих камер представлен ниже:

```
#include <opencv2/opencv.hpp>
using namespace cv;
using namespace std;
int main(int, char**)
{
    // снимок с первой камеры
    vector<int> wr_params;
    wr_params.push_back(cv::IMWRITE_JPEG_QUALITY);
    wr_params.push_back(50);
    VideoCapture cap(0);
    if (!cap.isOpened())
        return -1;
    Mat edges;
    Mat frame;
    cap >> frame;
    //сохранение снимка 1
    imwrite("electrode_left.jpg", frame, wr_params);
    // снимок со второй камеры
    VideoCapture cap1(2);
    if (!cap1.isOpened())
        return -1;
    Mat edges1;
    Mat frame1;
    cap1 >> frame1;
    //сохранение снимка 2
    imwrite("electrode_right.jpg", frame1, wr_params);
    // читаем снимок 1 электрода
    Mat img0 = imread("electrode_left.jpg", 1);
    int x = 325,
        y = 25,
        width = 200,
        height = 455;
    img0 = img0(Rect(x, y, width, height));
    Mat img1;
    cvtColor(img0, img1, CV_RGB2GRAY);
    // настройки фильтра
    Canny(img1, img1, 10, 100);
```

```

// нахождение контура
vector< vector<Point> > contours;
findContours(img1, contours, CV_RETR_EXTERNAL, CV_CHAIN_APPROX_NONE);
Mat mask = Mat::zeros(img1.rows, img1.cols, CV_8UC1);
drawContours(mask, contours, -1, Scalar(255), CV_FILLED);
crop.setTo(Scalar(255, 255, 255));
img0.copyTo(crop, mask);
normalize(mask.clone(), mask, 0.0, 255.0, CV_MINMAX, CV_8UC1);
// читаем снимок 2 электрода
Mat img2 = imread("electrode_right.jpg", 1);
Mat img3;
cvtColor(img2, img3, CV_RGB2GRAY);
Canny(img3, img3, 10, 100);
vector< vector<Point> > contours1;
findContours(img3, contours1, CV_RETR_EXTERNAL, CV_CHAIN_APPROX_NONE);
Mat mask1 = Mat::zeros(img3.rows, img3.cols, CV_8UC1);
drawContours(mask1, contours1, -1, Scalar(255), CV_FILLED);
crop1.setTo(Scalar(255, 255, 255));
img0.copyTo(crop1, mask);
normalize(mask1.clone(), mask1, 0.0, 255.0, CV_MINMAX, CV_8UC1);
// вывод изображений на экран
imshow("electrode_left", frame);
imshow("electrode_right", frame1);
imshow("electrodeR_canny", crop1);
imshow("electrodeL_canny", crop);
imwrite("era1_canny.jpg", img3);
imwrite("era1_mask.jpg", mask1);
waitKey();
return 0;
}

```

Результатом работы является снимок электрода обеими камерами. Нахождение их контуров с помощью детектора Canny и использование mask для улучшения изображения. Полученные изображения представлены на рисунках 4.4 и 4.5.

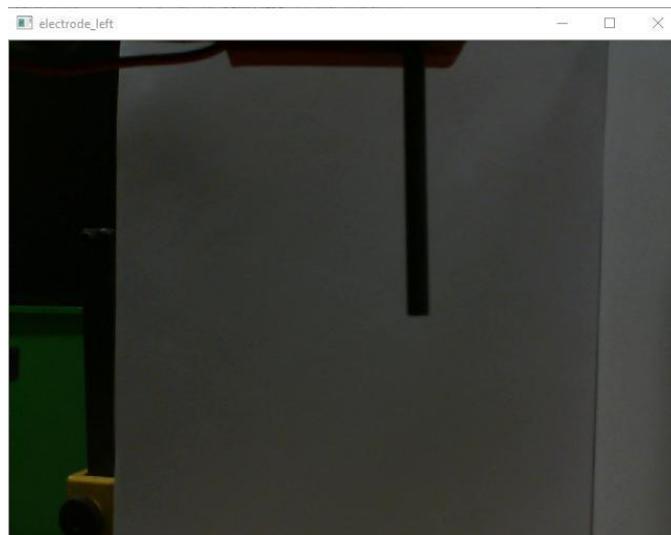


Рисунок 4.4 – Снимок электрода левой камерой

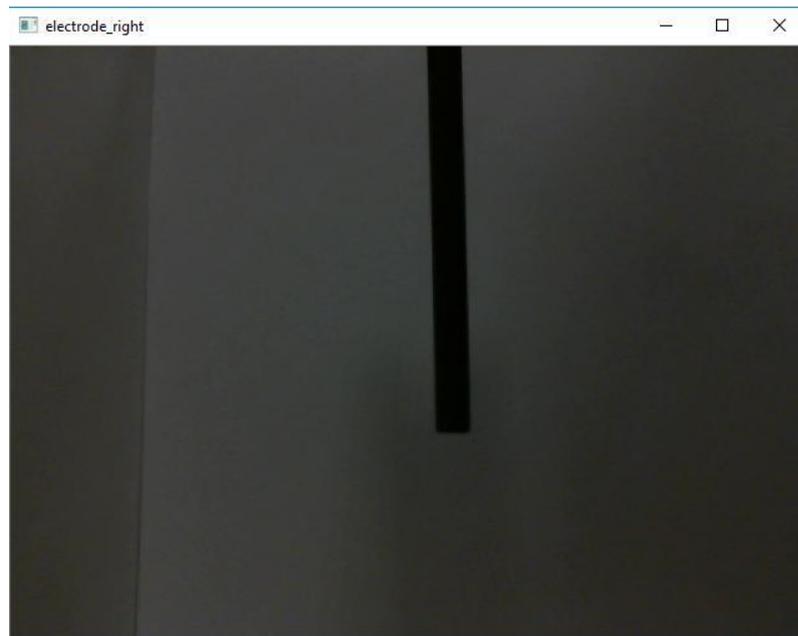


Рисунок 4.5 – Снимок электрода правой камерой

Ниже приведены найденные контуры (рисунок 4.6 и 4.7), полученные с помощью детектора Canny. Для левой камеры сделана обрезка заднего фона для распознавания только электрода. Фрагмент кода для обрезки представлен выше.

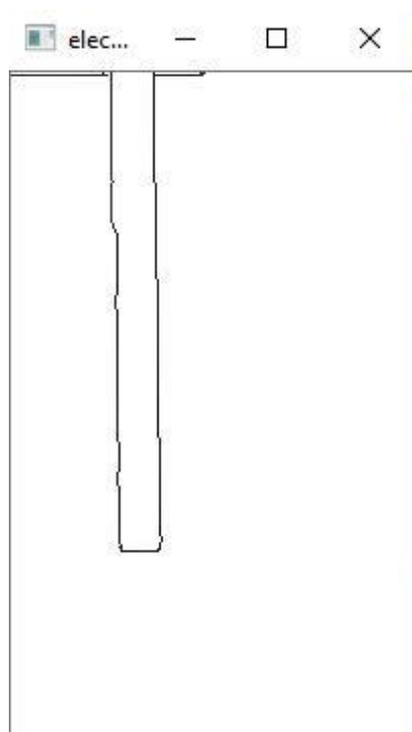


Рисунок 4.6 – Определение контура электрода левой камерой

Для определения эффективности работы нахождения контура был поставлен эксперимент. Программа запускалась 20 раз и определяла контур изображение с учетом изменения освещенности. В ходе практического эксперимента было получено 19 положительных результатов. Эффективность работы программы получается 95 %.

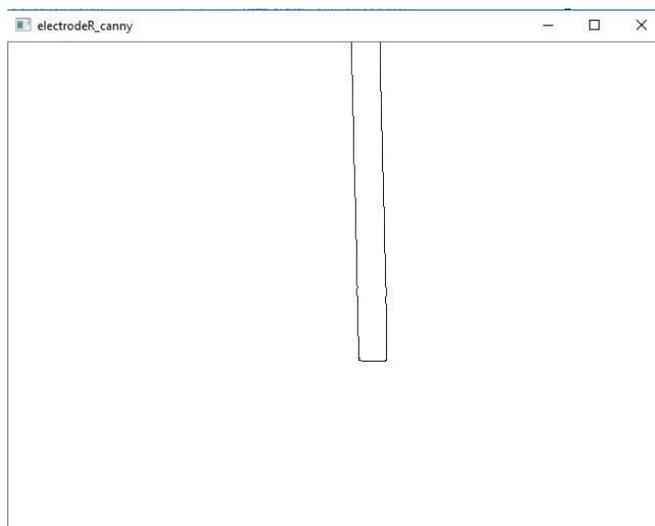


Рисунок 4.7 – Определение контура электрода правой камерой

4.3 Поиск пиксельных координат

Цифровое изображение, полученное камерой, может быть сохранено в виде массива в компьютере. Значение каждого элемента массива (так называемый пиксель) - это яркость точки изображения (или шкала серого, если цветное изображение, яркость пикселя изображения будут красного, зеленого и синего цветов трех цветов). Координаты (u, v) каждого пикселя определяются декартовой системой координат $u - v$, то есть количеством столбцов и числом строк пикселя в массиве. (U, v) - координаты системы координат изображения в пикселях, как показано на рисунке 4.8.

Основная задача заключалась в разработке метода поиска пиксельных координат. Конечный результат – это нахождения координат каждого контура электрода. В ходе разработки было решено написать программную часть для

нахождения пиксельных координат, а точнее координаты первого черного пикселя.

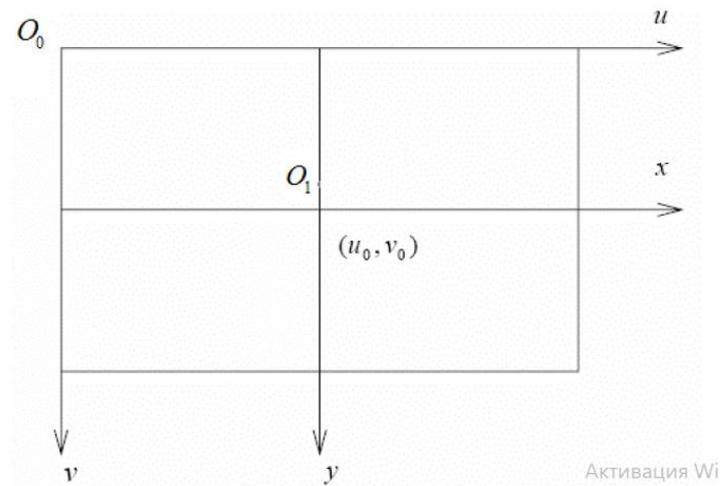


Рисунок 4.8 - Изображение системы координат и плоскости изображения системы координат

Фрагмент кода поиска пиксельных координат представлен ниже:

```
for (int i = 0; i < crop.rows; i++)
    for (int j = 0; j < crop.cols; j++)
        if (crop.at<Vec3b>(i, j)[0] < 255 &&
            crop.at<Vec3b>(i, j)[2] < 255 &&
            crop.at<Vec3b>(i, j)[1] < 255)
        {
            crop.at<Vec3b>(i, j)[0] = 0;
            crop.at<Vec3b>(i, j)[1] = 0;
            crop.at<Vec3b>(i, j)[2] = 0;
            k = i;
            l = j;
            i = crop.rows;
            break;
        }

cout << "x = " << k << endl << "y = " << l << endl;
```

Здесь представлен цикл для нахождения черного пикселя. Для начала задаемся начальными координатам – это 0 и 0. В нашем коде это столбцы i и строки j . Далее у нас задано условие для наших i и j : увеличивать значение пока не найдется пиксель не белого цвета. При нахождении черного пикселя значения

координат i и j перезаписываются в k и l . В итоге мы получаем координаты черного пикселя.

Кроме этого в коде добавлена функция определения пиксельных координат путем нажатия клавиши мыши на любую координату для обеих изображений, полученных с камер. Координаты выводятся на экран. Фрагмент кода, учитывающая данное условие представлен ниже:

```
#include <opencv2\opencv.hpp>
#include<locale.h>
using namespace cv;
using namespace std;
int k, l, m, n = 0;
void onMouse(int event, int x, int y, int flags, void* param)
{
    cv::Mat *im = reinterpret_cast<cv::Mat*>(param);
    // отправка event
    switch (event) {
        // event для движения мышки вниз
        case cv::EVENT_LBUTTONDOWN:
            // вывод пикселей в точках (x,y)
            std::cout << "при точках (" << x << ", " << y << ") значение
равно: "
                << static_cast<int>(im->at<uchar>(cv::Point(x, y))) <<
std::endl;
            break;
    }
}
```

Так же в конце кода нужно вызвать функцию `setMouseCallback`. Эта функция устанавливает функцию обратного вызова, которая вызывается каждый раз, когда в указанном окне Windows происходят нажатия мыши. Ниже приведено подробное объяснение каждого параметра вышеуказанной функции OpenCV:

- `winname` - имя окна OpenCV. Все события мыши, связанные с этим окном, будут зарегистрированы;
- `onMouse` - функция обратного вызова. Всякий раз, когда происходят нажатия мыши по окну эта функция обратного вызова будет вызвана;
- `userdata` - этот указатель будет передан функции обратного вызова.

Фрагмент кода для функции обратного вызова представлен ниже:

```
{
imshow("electrodeR_canny", crop1);
imshow("electrode_right", frame1);
imshow("electrode_left", frame);
setMouseCallback("electrodeL_canny", onMouse, reinterpret_cast<void*>(&crop));
setMouseCallback("electrodeR_canny", onMouse, reinterpret_cast<void*>(&crop1));
waitKey();
return 0;
}
```

Результат работы полной программы и его запуск через Visual Studio изображен на рисунке 4.9.

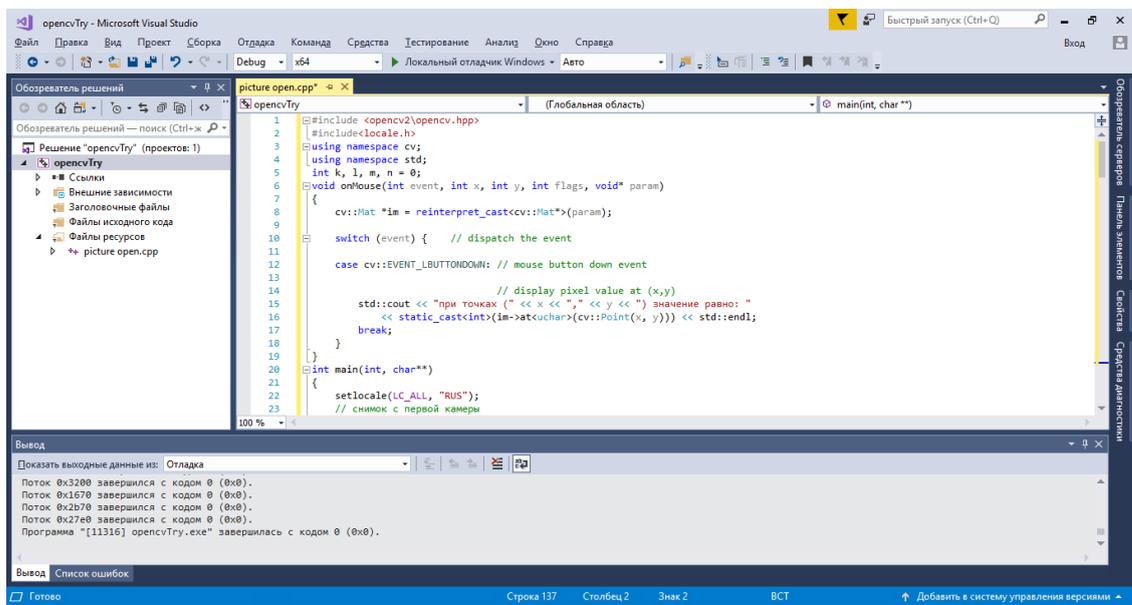


Рисунок 4.9 – Окно запуска программы в Visual Studio

Результаты нахождения координат представлены на рисунке 4.10. Так же ниже будут представлено изображение работы всей программы:

- открытие оригинальных изображений, снятые камерами (рисунок 4.11);
- найденные контуры с помощью детектора Canny (рисунок 4.11);
- полученные пиксельные координаты (рисунок 4.11);

- общая работа всей программы (рисунок 4.11).

Изображение электрода с найденными пиксельными координатами путем нажатия ЛКМ и вывода их на экран представлен на рисунке 4.12.

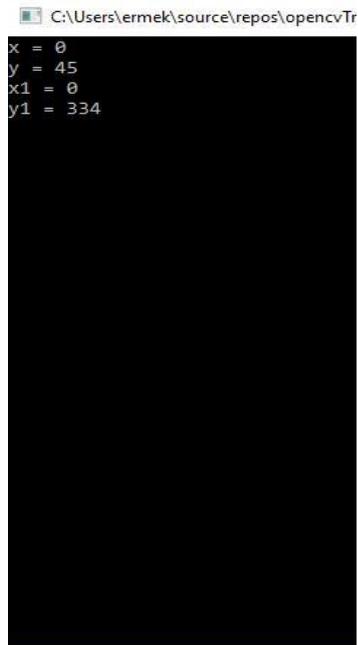


Рисунок 4.10 – Нахождение пиксельных координат изображения электрода

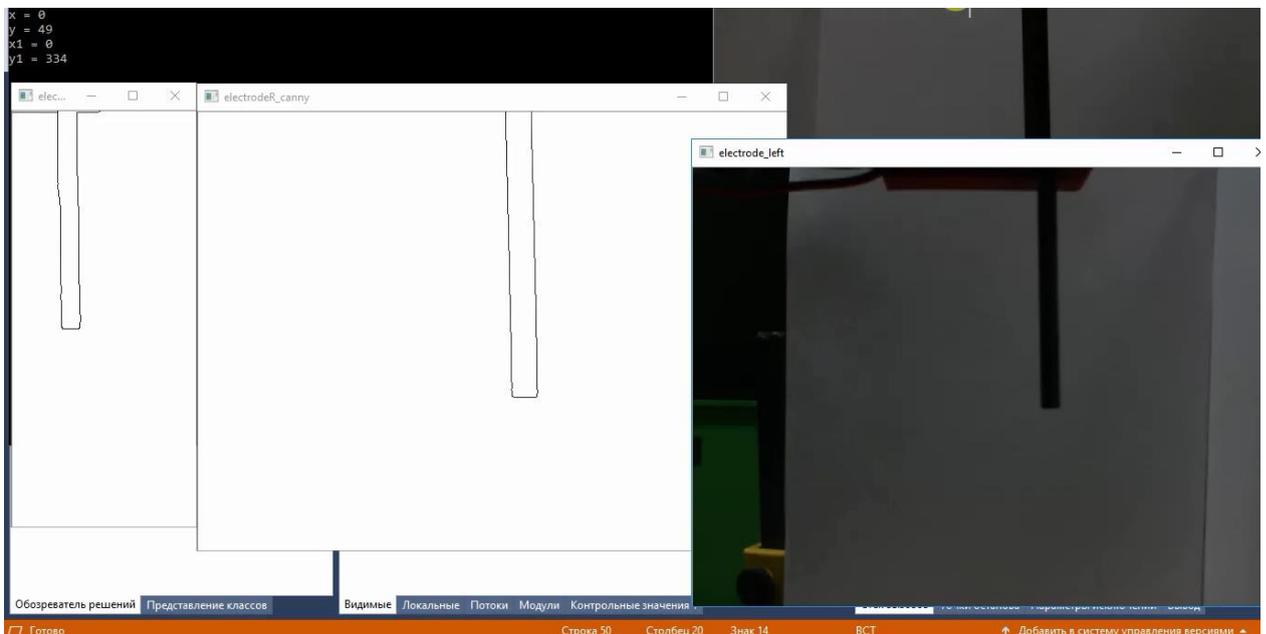


Рисунок 4.11 – Открытие оригинальных изображений и определение их контуров

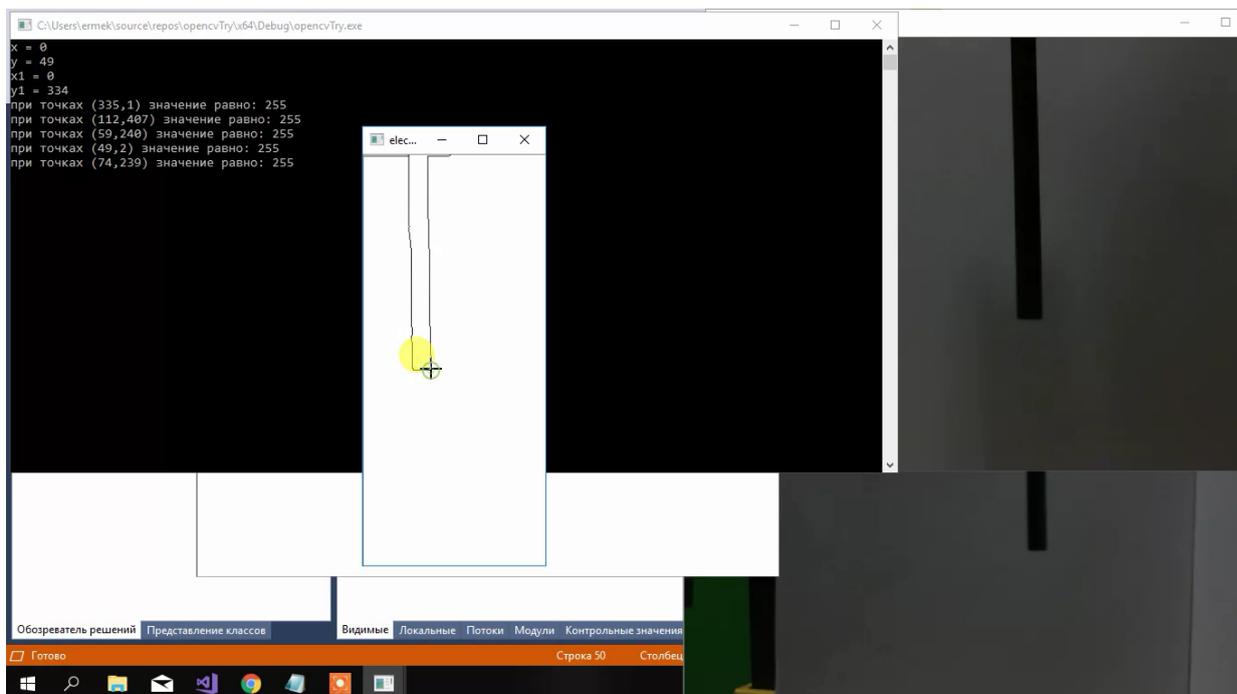


Рисунок 4.12 – Нахождение пиксельных координат изображения электрода путем вызова функции `setMouseCallback`

В ходе работы была реализована система для определения координат электрода. Данная система получает изображение электрода с камер. Далее находит контур электрода с помощью детектора контуров `Canny`. В конце обрабатывая полученное изображения контуров электрода находит координаты черных пикселей и выводит их на экран.

Система была полностью проверена на эффективность. В дальнейшем данные с этих координат будут отправлены на контроллер `Kuka`. В лаборатории `Kuka` была реализована приложение для соединения с контроллером `Kuka`, которая преобразует координаты в координаты самого манипулятора, что позволяет изменить его положение. Она так же написана на языке `C++`.

5. СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ

С развитием научного прогресса безопасность жизнедеятельности человека играет огромную роль на производстве. В соответствии с требованиями безопасности была создана наука, которая помогает обеспечить БЖД. БЖД – это комплекс мероприятий, направленных на обеспечение безопасности человека в окружающей его среде обитания, сохранение его здоровья, разработку методов и средств защиты путём снижения влияния вредных и опасных факторов до допустимых значений, выработку мер по ограничению ущерба в ликвидации последствий чрезвычайных ситуаций мирного и военного времени.

Социальная ответственность [29] – это ответственность организации за воздействие ее решений и деятельности на общество и окружающую среду через прозрачное и этическое поведение, которое содействует устойчивому развитию, включая здоровье и благосостояние общества; учитывает ожидания заинтересованных сторон; соответствует применяемому законодательству и согласуется с международными нормами поведения; интегрировано в деятельность всей организации и применяется в ее взаимоотношениях.

С каждым годом возрастает интенсивность применения компьютерной техники в сферах жизнедеятельности человека. При работе с компьютером человек подвергается воздействию ряда опасных и вредных производственных факторов: электромагнитных полей, радиочастотному (высоких, ультравысоких и средних частот), инфракрасному излучению, шуму и вибрации, статическому электричеству. Работа с компьютером характеризуется значительным умственным напряжением, высокой напряженностью зрительной работы и большой нагрузкой на кисти рук при работе с периферийными устройствами ЭВМ.

5.1 Техногенная безопасность

5.1.1 Микроклимат

Благоприятные (комфортные) метеорологические условия на производстве являются важным фактором в обеспечении высокой производительности труда и в профилактике заболеваний. При несоблюдении гигиенических норм микроклимата снижается работоспособность человека, возрастает опасность возникновения травм и ряда заболеваний, в том числе профессиональных.

По степени физической тяжести работа инженера-программиста относится к категории лёгких работ. В соответствии с временем года и категорией тяжести работ определены параметры микроклимата согласно требованиям [30] и приведены в Таблице 5.1.

Таблица 5.1 – Оптимальные параметры микроклимата рабочего места

Период года	Категория работ	Температура воздуха, °С	Относительная влажность воздуха, %	Скорость движения
Холодный	Категория 1а	23-25	40-60	0,1
Теплый	Категория 1а	20-22	40-60	0,1

В зимнее время в помещении предусмотрена система отопления. Она обеспечивает достаточное, постоянное и равномерное нагревание воздуха. В соответствии с характеристикой помещения определен расход свежего воздуха согласно [30] и приведен в Таблице 5.2.

Таблица 5.2 – Нормы подачи свежего воздуха в помещения, где расположены компьютеры

Характеристика помещения	Объемный	расх
	подаваемого	в помещен

Объем до 20 м ³ на человека	Не менее 30
20...40 м ³ на человека	Не менее 20

5.1.2 Естественная освещенность

Правильное освещение помещений и рабочих зон одно из главных условий создания безопасных и благоприятных условий труда. При недостаточной освещенности развивается утомление зрения, понижается общая работоспособность и производительность труда, возрастает количество брака, повышается опасность производственного травматизма, низкая освещенность способствует развитию близорукости.

Рабочая зона или рабочее место разработчика освещается таким образом, чтобы можно было отчетливо видеть процесс работы, не напрягая зрения, а также исключается прямое попадание лучей источника света в глаза. Кроме того, уровень необходимого освещения определяется степенью точности зрительных работ. Наименьший размер объекта различения составляет 0.5 - 1 мм. В помещении отсутствует естественное освещение. По нормам освещенности [31] и отраслевым нормам, работа за ПК относится к зрительным работам высокой точности для любого типа помещений

Таблица 5.3 – Нормирование освещённости для работы за ПК [31]

1	2	3	4	5	6	7	8	9	10	11
Выс ок ой точн	От 0,3	Б	1	Бол ее	300	100	40	15	3,0	1,0
	От 0,5		2	Мен ее	200	75	60	20	2,5	0,7

Где:

- 1 -характеристика зрительных работ;
- 2 -наименьший или эквивалентный размер объекта различения, мм;
- 3-разряд зрительной работы;

- 4 -подразряд зрительной работы;
 - 5 -относительная продолжительность зрительной работы, %;
 - 6 -освещенность на рабочей поверхности от системы общего искусственного освещения, лк;
 - 7 -цилиндрическая освещенность, лк;
 - 8-показатель дискомфорта;
 - 9-коэффициент пульсации освещенности, %;
 - 10-КЕО при верхнем освещении, %;
 - 11-КЕО при боковом освещении, %
- Требования к освещению на рабочих местах, оборудованных ПК, представлены в таблице 5.4. [32]

Таблица 5.4 – Требования к освещению на рабочих местах, оборудованных ПК [32]

Освещенность на рабочем столе	300-500 лк
Освещенность на экране ПК	не выше 300 лк
Блики на экране	не выше 40 кд/м ²
Прямая блескость источника света	200 кд/м ²
Показатель ослеплённости	не более 20
Показатель дискомфорта	не более 15
Отношение яркости:	
– между рабочими поверхностями	3:1–5:1
– между поверхностями стен и оборудования	10:1
Коэффициент пульсации:	не более 5%

Освещаемая площадь помещения составляет 160 м², количество ламп 24, каждая 2500 люмен. Тогда $E=60000/160=375$ лк. Это является достаточным для выполнения условий, приведенных в таблице 5.4.

5.1.3 Уровень шума

Одним из важных факторов, влияющих на качество выполняемой работы, является шум. Шум ухудшает условия труда, оказывая вредное действие

на организм человека. Работающие в условиях длительного шумового воздействия испытывают раздражительность, головные боли, головокружение, снижение памяти, повышенную утомляемость, понижение аппетита, боли в ушах и т. д. Такие нарушения в работе ряда органов и систем организма человека могут вызвать негативные изменения в эмоциональном состоянии человека вплоть до стрессовых. Под воздействием шума снижается концентрация внимания, нарушаются физиологические функции, появляется усталость в связи с повышенными энергетическими затратами и нервно- психическим напряжением, ухудшается речевая коммутация. Все это снижает работоспособность человека и его производительность, качество и безопасность труда. Длительное воздействие интенсивного шума (выше 80 дБ(А)) на слух человека приводит к его частичной или полной потере.

При выполнении основной работы на ПЭВМ уровень шума на рабочем месте не должен превышать 50 дБ. Допустимые уровни звукового давления в помещениях для персонала, осуществляющего эксплуатацию ЭВМ при разных значениях частот, приведены в [33].

Для устранения шума необходимо прочистить вентилятор от пыли или заменить полностью.

5.1.4 Уровень электромагнитных излучений

Энергетическое влияние электромагнитного излучения может быть различной степени и силы. От неощутимого человеком (что наблюдается наиболее часто) до теплового ощущения при излучении высокой мощности, но подобный эффект проявляет себя только при превышении допустимого уровня. Помимо электромагнитного излучения, при работе компьютера образуется электростатическое поле, оно способно деионизировать окружающий воздух. Компьютер, при долгой работе, нагревается, что делает воздух не только слабо ионизированным, но и сухим. Такой воздух является очень вредным, он тяжелый для дыхания и способствует, при благоприятной среде, развитию многих аллергических заболеваний и, соответственно болезней органов

дыхания.

В данной выпускной работе будет рассматриваться такой источник электромагнитного поля как персональный компьютер.

По [34] энергетическая экспозиция за рабочий день не должна превышать значений, указанных в таблице 5.5.

Таблица 5.5 –Предельно допустимые значения энергетической экспозиции

Диапазоны частот	Предельно допустимая энергетическая экспозиция		
	По электрической составляющей, $(В/м)^2 \times ч$	По магнитной составляющей, $(А/м)^2 \times ч$	По плотности потока энергии $(мкВт/см^2) \times ч$
30 кГц - 3 МГц	20000,0	200,0	х
3 - 30 МГц	7000,0	Не разработаны	х
30 - 50 МГц	800,0	0,72	х
50 - 300 МГц	800,0	Не разработаны	х
300 МГц - 300 ГГц	х	х	200,0

Для обеспечения меньшего электромагнитного излучение использован жидкокристаллический монитор. Необходимо учитывать расстояние до монитора, так как при большем расстоянии от человека оказывается меньшее влияние. В связи с тем, что электромагнитное излучение от стенок монитора намного больше, необходимо ограничивать его стенами, т.е. ставить в углу. Необходимо чтобы компьютер был заземлен, а также необходимо по возможности сокращать время работы за компьютером.

5.1.5 Электробезопасность

Даже самая надежная изоляция теряет свои свойства при длительной эксплуатации в сложных производственных условиях. Опасность поражения электрическим током существенно возрастает, если человек находится в помещении с металлическим полом или контактирует с оборудованием, находящимся вблизи заземленных металлических предметов. Согласно разделу

1.1.13 правил устройства электроустановок производственные цеха, торговые и бытовые помещения принято делить на три отдельных класса, характеризующих степень опасности. Рассмотрим каждый из них в отдельности.

Первый класс - "помещения без повышенной опасности"

В данную категорию входят помещения, характеризующиеся пониженной влажностью воздуха (до 75%), оборудованные при необходимости вентиляционной системой и отоплением.

Кроме того, полы в таких помещениях должны быть не токопроводящими. Под понятием токопроводящие полы подразумевают металлические, железобетонные, земляные и т.д. Для причисления производственного цеха в разряд безопасных, его пол следует покрыть диэлектрическим материалом. К наглядным примерам безопасных объектов можно отнести обычные жилые и офисные помещения, кладовые для хранения инструментов, лаборатории, а также производственные цеха приборостроительных предприятий, проект которых изначально предусматривал наличие изолированного пола, мощные воздушные фильтры для устранения пыли и систему регулирования температуры воздуха.

Второй класс - "помещения с повышенной опасностью"

Действующая классификация помещений по опасности поражения электрическим током определяет объект к категории повышенной опасности при наличии в них одного из следующих условий:

- сырость (помещения, с относительной влажностью больше 75 %);
- токопроводящая пыль (постоянное образование пыли с токопроводящими свойствами);
- помещения с токопроводящими полами (наличие железобетонных, металлических, кирпичных и иных типов токопроводящих напольных покрытий);
- высокий уровень температуры (помещения в которых температура постоянно превышает +35⁰С);

- условия (возможность), когда человек может одновременно прикоснуться к металлическим корпусам электрооборудования и к заземленным металлоконструкциям зданий (из примеров можно привести случай, когда человек может взяться одной рукой за батарею отопления - второй за корпус станка).

Третий класс - "особо опасные помещения"

К наиболее опасной категории относятся помещения с высоким уровнем влажности, концентрированной взвесью химически активных веществ в воздухе, а также не менее двух дополнительных факторов из категории помещений повышенной опасности.

- помещения с «особой сыростью» (относительная влажность близка к 100 %). Не путать с пунктом выше;
- помещения, в которых присутствует химическая активность и органическая среда (в следствии отложений приводят к разрушению изоляции электрооборудования);
- два и больше условий из пункта №2 (для помещений с повышенной опасностью).

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

При работе с компьютером существует опасность электропоражения:

- при прикосновении к нетоковедущим частям, оказавшимся под напряжением (в случае нарушения изоляции токоведущих частей ПЭВМ);
- при прикосновении с полом, стенами, оказавшимися под напряжением;
- имеется опасность короткого замыкания в высоковольтных блоках: блоке питания и блоке дисплейной развертки.

В зависимости от условий в помещении опасность поражения человека электрическим током увеличивается или уменьшается. По [35] помещение, в

котором находится рабочее место, относится к категории помещений без повышенной опасности. Его можно охарактеризовать, как сухое, непыльное, с токопроводящими полами и нормальной температурой воздуха. Температурный режим, влажность воздуха, химическая среда не способствуют разрушению изоляции электрооборудования.

Безопасность при работе с электроустановками обеспечивается применением различных технических и организационных мер. Основные технические средства защиты от поражения электрическим током:

- изоляция токопроводящих частей и ее непрерывный контроль;
- установка оградительных устройств;
- отдельные помещения для манипуляторов;
- предупредительная сигнализация и блокировки;
- защитное заземление;
- зануление;
- защитное отключение по [35].

5.1.6 Безопасность в чрезвычайных ситуациях

В процессе работы в помещении осуществляется водоснабжение и образование бытовых сточных вод (помещения гигиены и водопотребления). Сточные воды передаются в сточные системы. Для утилизации макулатуры рядом с принтером установлена специальная урна, предназначенная для накопления всех ненужных печатных материалов. После наполнения урны, работник предприятия обязан изыскивать средства для того, чтобы отправить данные накопители бумажного мусора на станцию вторсырья. Далее вся эта бумага идет на переработку как вторичное сырье.

5.2 Организационные мероприятия обеспечения безопасности

5.2.1 Эргономические требования к рабочему месту

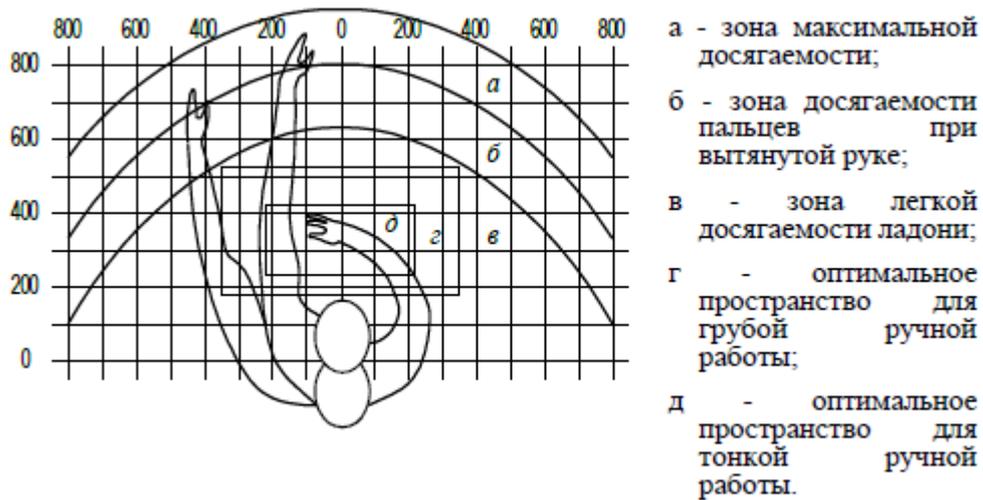


Рисунок 5.1 – зоны досягаемости рук в горизонтальной плоскости

Оптимальное размещение предметов труда и документации в зонах досягаемости:

- дисплей размещается в зоне *а* (в центре);
- системный блок размещается в предусмотренной нише стола;
- клавиатура – в зоне *г/д*;
- «мышь» – в зоне *в* справа;
- сканер в зоне *а/б* (слева);
- принтер находится в зоне *а* (справа);
- документация, необходимая при работе – в зоне легкой досягаемости ладони – *в*, а в выдвижных ящиках стола – литература, неиспользуемая постоянно.

На рисунке 5.2 показан пример размещения основных и периферийных составляющих ПК на рабочем столе программиста.

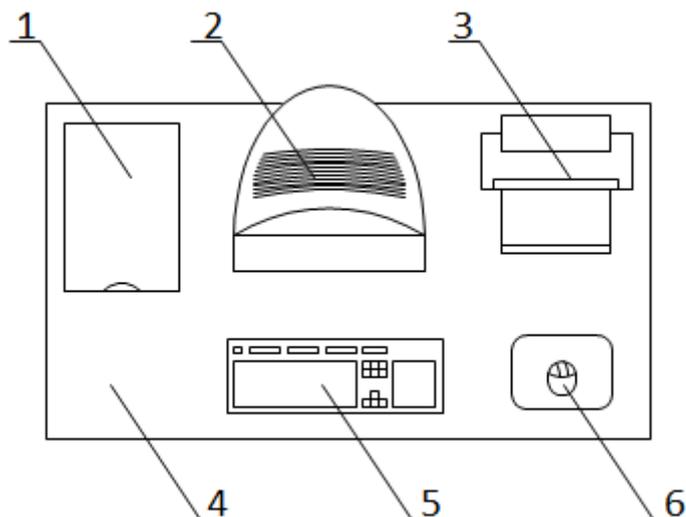


Рисунок 5.2 – Размещение основных и периферийных составляющих ПК.

1 – сканер, 2 – монитор, 3 – принтер, 4 – поверхность рабочего стола,
5 – клавиатура, 6 – манипулятор типа «мышь».

В помещениях, где находится компьютер, необходимо обеспечить следующие величины коэффициента отражения для потолка 60-70, для стен 40-50, для пола около 30.

5.3 Пожарная безопасность

Пожарная безопасность может быть обеспечена мерами пожарной профилактики и активной пожарной защиты. Пожарная профилактика включает комплекс мероприятий, направленных на предупреждение пожара или уменьшение его последствий. Активная пожарная защита – меры, обеспечивающие успешную борьбу с пожарами или взрывоопасной ситуацией.

Возникновение пожара в помещении, где установлена вычислительная и оргтехника, приводит к большим материальным потерям и возникновению чрезвычайной ситуации. Чрезвычайные ситуации приводят к полной потере информации и большим трудностям восстановления всей информации в полном объёме.

Согласно нормам технологического проектирования [36], в зависимости от характеристики используемых в производстве веществ и их количества, по пожарной и взрывной опасности помещения подразделяются на категории А, Б, В, Г, Д.

Данное помещение относится к категории В [29], производства, связанные с обработкой или применением твердых сгораемых веществ и материалов.

Для исключения возникновения пожара необходимо:

- вовремя выявлять и устранять неисправности;
- не использовать открытые обогревательные приборы, приборы кустарного производства в помещении лаборатории;
- определить порядок и сроки прохождения противопожарного инструктажа и занятий по пожарно-техническому минимуму, а также назначить ответственного за их проведения.

В случае возникновения пожара необходимо отключить электропитание, вызвать по телефону пожарную команду, произвести эвакуацию и приступить к ликвидации пожара огнетушителями. При наличии небольшого очага пламени можно воспользоваться подручными средствами с целью прекращения доступа воздуха к объекту возгорания.

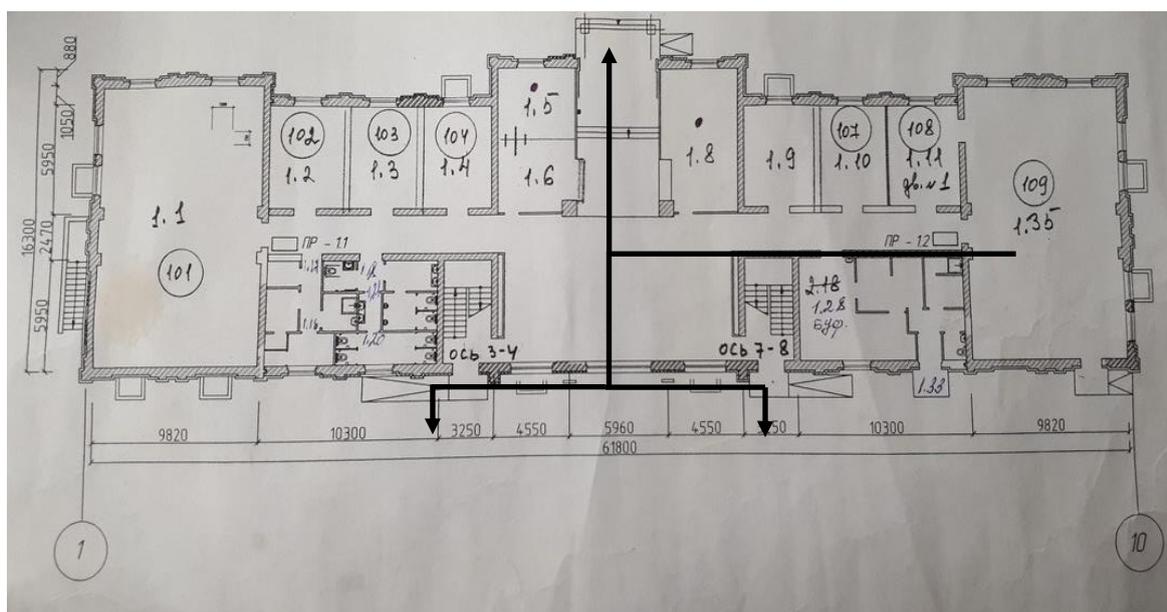


Рисунок 5.3 – План эвакуации при пожаре и других ЧС из помещений учебного корпуса

Для тушения пожаров в помещении необходимо установить углекислотный огнетушитель типа ОУ-5.

Покидать помещение необходимо согласно плану эвакуации. На рисунке 5.3 изображен план первого этажа Научного парка ТПУ. На первом этаже

имеются запасные выходы у лестниц с каждой стороны.

5.4 Особенности законодательного регулирования проектных решений

Государственный надзор и контроль в организациях независимо от организационно-правовых форм и форм собственности осуществляют специально уполномоченные на то государственные органы и инспекции в соответствии с федеральными законами.

К таким органам относятся Федеральная инспекция труда, Государственная экспертиза условий труда Федеральная служба по труду и занятости населения (Минтруда России Федеральная служба по экологическому, технологическому и атомном надзору (Госгортехнадзор, Госэнергонадзор, Госатомнадзор России) Федеральная служба по надзору в сфере защиты прав потребителей и благополучия человека (Госсанэпиднадзор России) и др.

Так же в стране функционирует Единая государственная система предупреждения и ликвидации чрезвычайных ситуаций (РСЧС), положение о которой утверждено Постановлением Правительства Российской Федерации [37], в соответствии с которым, система объединяет органы управления, силы и средства.

- Продолжительность рабочего дня не должна превышать 40 часов в неделю.
- не рекомендуется работать за компьютером более 6 часов за смену; рекомендуется делать перерывы в работе за ПК продолжительностью 10 минут через каждые 50 минут работы;
- продолжительность непрерывной работы за компьютером без регламентированного перерыва не должна превышать 2 часов;
- во время регламентированных перерывов целесообразно выполнять комплексы упражнений.
- при нерегламентированной работе повышенной интенсивности возможны головные боли, нервные срывы и др.

6. ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И РЕСУРСОСБЕРЕЖЕНИЕ

6.1 Организация и планирование работ

Для оценки процесса реализации проекта необходимо составить линейный график, исполнителями являются научный руководитель(НР) и студент (С). Для каждого из выделенных этапов определим его исполнителей и долю участия каждого из исполнителей в его выполнении. Полученные результаты сведем в таблицу 6.1.

Таблица 6.1 – Перечень выполняемых работ с указанием доли участия исполнителей

Этапы работы	Исполнители	Загрузка исполнителей
1. Постановка целей и задач исследования	НР, С	НР – 70% С – 30%
2. Анализ готовых технических решений	НР, С	НР – 30% С – 70%
3 Разработка и утверждение технического задания (ТЗ)	НР, С	НР – 30% С – 70%
4. Изучение основ проектирования и программирования систем компьютерного зрения	С	С – 100%
5. Подбор используемого оборудования	С	С – 100%
6. Разработка календарного плана выполнения работ	НР, С	НР – 20% С – 80%
7. Составление алгоритма работы системы управления	НР, С	НР – 30% С – 70%

8.Выбор методов технического зрения	С	С – 100%
9. Оформление расчетно-пояснительной записки	С	С – 100%
10. Подведение итогов	НР	НР – 100%

Дадим краткое описание каждому из этапов работ с указанием результатов.

Первым этапом является анализ предметной области, с последующими выводами о том, какие цели и задачи нужно поставить. Так же производится четкое распределение нагрузки и обязанностей между научным руководителем и студентом. По окончании изучения всех предложенных материалов, необходимых для достижения поставленной цели, студент просматривает дополнительные источники и предлагает максимально детальный план работ. Данный план корректируется и подтверждается в последствие научным руководителем. Основными пунктами данного плана являются задачи, прописанные предварительно студентом.

Второй этап подразумевает просмотр уже реализованных проектов, в данной области. Сравнение реализованных решений помогает более взвешенно подобрать оборудование, программное обеспечение и выбрать алгоритмы работы системы.

Результатом третьего этапа станут технические требования к разрабатываемой системе управления, в частности к её функционалу. Так же обоснование целесообразности и актуальности разработки.

Результатом выполнения четвертого этапа является глубокое понимание вопроса по изучаемой тематике, подбор нужного материала для исследования, выбор необходимого программного обеспечения и определение направления последующей работы.

Результатом выполнения пятого этапа является подбор необходимого оборудования. В данном случае осуществляется выбор камеры по таким параметрам, как фокусировка и разрешение объектива.

Результатом выполнения шестого этапа будет календарный план, в котором содержится расписание выполнения всех видов деятельности по достижению цели

- разработки технического зрения для системы управления сортировочным роботом.

В седьмом этапе происходит формирование основных концепций и идей, закладываемых в создаваемую систему управления. Результатом выполнения является готовый алгоритм работы системы, который показывает, как происходит взаимодействие методов технического зрения с работой манипулятора.

В восьмом этапе происходит подбор методов технического зрения и связывание их воедино. Результатом выполнения является программная реализация выбранных методов и отладка готовой программы.

Девятый этап направлен на оформление всего изученного, смоделированного и изготовленного материала в текстовом формате для защиты и так же для продолжения исследований в данном направлении.

На десятом этапе студент предоставляет свои результаты на оценку руководителю и получает обратную связь о проделанной им работе.

6.2 Продолжительность этапов работ

Так как отсутствует нормативная база по проводимым работам, а также достоверная информация о процессе выполнения подобных работ иными исполнителями, воспользуемся экспертным способом оценки продолжительности выполнения запланированных работ. Произведем оценку минимального и максимального времени выполнения каждого из этапов. Рассчитаем ожидаемое время выполнения работ, воспользовавшись формулой

$$t_{ожі} = \frac{2 \cdot t_{max} + 3 \cdot t_{min}}{5}$$

где $t_{ожі}$ – ожидаемое время выполнения i -го этапа работ в чел.-дн., t_{min} – минимальная продолжительность работы, дн.; t_{max} – максимальная продолжительность работы, дн.

Ожидаемое, минимальное и максимальное время исполнения в предложенной выше формуле, оцениваются в рабочих днях на человека. Произведем перевод этих величин в календарные дни, воспользовавшись следующей формулой:

$$T_{кд} = T_{к} \cdot T_{рд}$$

где $T_{кд}$ – продолжительность выполнения этапа в календарных днях; T_k – коэффициент календарности, позволяющий перейти от длительности работ в рабочих днях к их аналогам в календарных днях, и рассчитываемый по формуле

$$T_k = \frac{T_{кал}}{T_{кал} - T_{вд} - T_{пд}}$$

где $T_{кал}$ – календарные дни ($T_{кал} = 365$); $T_{вд}$ – выходные дни ($T_{вд} = 52$); $T_{пд}$ – праздничные дни ($T_{пд} = 10$).

$$T_k = \frac{365}{365 - 52 - 10} = 1,48$$

В свою очередь рабочие дни рассчитываются по следующей формуле:

$$T_{рд} = \frac{t_{ож}}{K_{вн}} \cdot K_d$$

$K_{вн}$ – коэффициент выполнения работ, учитывающий влияние внешних факторов на соблюдение предварительно определенных длительностей, в частности, возможно $K_{вн} = 1$; K_d – коэффициент, учитывающий дополнительное время на компенсацию непредвиденных задержек и согласование работ ($K_d = 1–1,2$; в этих границах конкретное значение принимает сам исполнитель).

Для простоты расчетов примем K_d и $K_{вн}$, равными единице. Тогда формула для расчета календарных дней преобразуется в следующую:

$$T_{кд} = T_k \cdot T_{рд} = t_{ож} \cdot T_k = \frac{3t_{min} + 2t_{max}}{5} \cdot 1.48$$

Воспользовавшись данными из таблицы 4, приведенными выше формулами, произведем расчет продолжительности выполнения работ научным руководителем и студентом в календарных днях. Результаты расчетов представлены в таблице 6.2. На основе данной таблице составим линейный график работ, представленный в таблице 6.3.

Таблица 6.2 – Расчет трудозатрат на выполнение проекта

Этап	Исполнитель	Продолжительность работ, дни			Трудоемкость работ по исполнителям чел.-дн.			
					Т _{рд}		Т _{кд}	
		t_{min}	t_{max}	$t_{ож}$	НР	С	НР	С
1	2	3	4	5	6	7	8	9
1. Постановка целей и задач исследования	НР, С	2	4	2,8	1,96	0,84	2,9008	1,2432
2. Анализ готовых технических решений (ТЗ)	НР, С	2	4	2,8	0,84	1,96	1,2432	2,9008
3. Разработка и утверждение технического задания (ТЗ)	С	4	7	5,2	1,56	3,64	2,3088	5,3872
4. Изучение основ проектирования и программирования систем компьютерного зрения	НР, С	40	50	44	0	44	0	65,12
5. Подбор используемого оборудования	НР, С	2	3	2,4	0	2,4	0	3,552
6. Разработка календарного плана выполнения работ	НР, С	2	4	2,8	0,56	2,24	0,8288	3,3152
7. Составление алгоритма работы системы управления	НР, С	10	15	12	3,6	8,4	5,328	12,432
8. Выбор методов технического зрения	НР, С	14	20	16,4	0	16,4	0	24,272
9. Оформление расчетно-пояснительной записки	С	12	18	14,4	0	14,4	0	21,312
10. Подведение итогов	НР, С	3	5	3,8	3,8	0	5,624	0
Итого:				106,6	12,32	94,28	18,2336	139,5344

Таблица 6.3 – Линейный график работ

Этап	НР	С	Январь			Февраль			Март			Апрель			Май		
			10	20	30	40	50	60	70	80	90	100	110	1120	130	140	150
1	2,91	1,24	■														
2	1,24	2,9	■	■													
3	2,31	5,39	■	■	■												
4	0	65,12		■													
5	0	3,55								■							
6	0,83	3,31								■	■						
7	5,33	12,43									■	■					
8	0	24,27										■					
9	0	21,31												■			
10	5,62	0														■	

НР – ■; С – ■

6.3 Расчет сметы затрат на выполнение проекта

Расчет затрат для выполнения комплекса работ по проекту производится по следующим статьям:

- материалы и покупные изделия;
- заработная плата;
- социальный налог;
- расходы на электроэнергию;
- амортизационные отчисления;
- командировочные расходы;
- оплата услуг связи;
- арендная плата за пользование имуществом;
- прочие услуги (сторонних организаций);
- прочие (накладные) расходы.

В процессе выполнения работ по проекту исполнителям не требуется производить оплату услуг сторонних организаций, так как они являются подразделениями и/или подрядными организациями ТПУ, а также нет необходимости производить оплату за аренду имущества. Исполнители также не производят оплату командировок, услуг связи. Таким образом, касательно проекта, необходимо произвести расчет материалов и покупных изделий, заработную плату исполнителей, социальный налог, расходы на электроэнергию, амортизацию и накладные расходы.

6.3.1 Расчет затрат на материалы и покупные изделия

Работа над проектом заключается в проведении теоретических исследований в своем начале и практических – в конце. Для теоретических исследований необходимо наличие программной среды Python 3.6, библиотеки компьютерного зрения OpenCV, MS Office. Первые два продукта распространяются бесплатно, а лицензия на MS Office приобретена ТПУ и доступна в открытом доступе для студентов, следовательно, затраты на использование данных программ не учитываются. Так как затрат на все программное обеспечение удалось избежать, то материальные затраты в данной части состоят только из расходов

на канцелярские принадлежности и расходные материалы для принтера.

При проведении практических исследований необходимо было произвести закупку такого оборудования, как манипулятор и цифровая камеры.

Данное оборудование уже было в стенах ТПУ, и было предоставлено для использования, поэтому материальных затрат на закупку оборудования тоже удалось избежать. Единственной тратой была покупка изделий для сортировки, а конкретно детских кубиков.

В материальные затраты также входят транспортно-заготовительные расходы (ТЗР), которые обуславливаются затратами на совершение сделок купли-продажи материалов, их доставку к месту использования. Обычно ТЗР составляют от 5% до 20% от общей цены материалов. Положим норму ТЗР равной 10%.

С учетом вышесказанного, составим таблицу 6.4, в которой учтены затраты на первый и второй этапы работы.

Таблица 6.4 – Расчет затрат на материалы

Наименование материалов	Цена за ед., руб.	Кол-во	Сумма
Затраты на этапе теоретических исследований			
Бумага для принтера, упак. 500 листов	257	2	514
Тетрадь общая, формат А4	93	3	279
Мультифоры, упак. 100 шт.	100	2	200
Скобы для степлера, упак. 500 шт.	84	1	84
Итого			1077
Итого (с учетом ТЗР)			1184,7
Затраты на этапе практических исследований			
Набор детских кубиков, шт.	600	1	600
Итого			600
Итого (с учетом ТЗР)			660
Общие затраты			
Итого			1677
Итого (с учетом ТЗР)			1844,7

6.3.2 Расчет заработной платы

Данная статья расходов включает в себя заработную плату студента (исполнитель проекта) и научного руководителя, а также премии, входящие в фонд

заработной платы. Расчет заработной платы выполняется на основе трудоемкости выполнения каждого этапа и величины месячного оклада исполнителя. Величина месячных окладов (MO) для сотрудников ТПУ находится в общем доступе и её можно найти на портале. Оклад студента принимается равным окладу соответствующего специалиста низшей квалификации в организации, в которой выполняются подобные виды работ или же там, где студент проходил преддипломную практику.

Для расчета заработной платы студента примем величину месячного оклада, равную окладу младшего научного сотрудника на полставки, т.е. $MO_C = 7437,225$ руб.; а величина заработной платы руководителя – величина месячного оклада доцента, д.т.н. $MO_{HP} = 27484,86$.

Для дальнейшего расчета зарплаты необходимо вычислить среднедневную ставку с учетом среднего количества рабочих дней в месяце. В 2018 году 247 рабочих дней, следовательно, количество рабочих дней в месяце равно 20,6. Таким образом, среднедневная тарифная заработная плата рассчитывается по формуле:

$$ЗП_{\text{дн}} = MO/20,6$$

Для студента эта величина составит примерно 361 рубль, а для научного руководителя 1334,22 рубля в день. Зная среднедневную оплату труда, на можно произвести расчет основной заработной платы за период выполнения работ. Затем, учитывая премии, дополнительную заработную плату и районный коэффициент, можно рассчитать полную заработную плату по следующей формуле:

$$ЗП_{\text{полн}} = ЗП_{\text{дн}} * T_{PD} * K_{PP} * K_{доп} * K_p$$

где T_{PD} – трудоемкость проекта для сотрудника в рабочих днях; $K_{PP} = 1,1$ – коэффициент премирования; $K_{доп} = 1,13$ – коэффициент дополнительной заработной платы для пятидневной рабочей недели; $K_p = 1,3$ – районный (северный) коэффициент. Выполнив расчеты по приведенным формулам, результаты приведены в таблице 6.5.

Таблица 6.5 – Затраты на заработную плату

Исполнитель	Оклад, руб./мес.	Среднедневная ставка, руб./день	Затраты времени, раб. дни	Коэффициент	Фонд з/платы, руб.
НР	27484,8	1334,22	18	1,62	38905,9
С	7437,22	361	140		81874,8
Итого					120780,6

6.3.3 Расчет затрат на электроэнергию

Данная статья учитывает затраты на электроэнергию, которая потребляется всем оборудованием в течение работы над проектом. Затраты на электроэнергию рассчитываются по формуле:

$$C_{эл.об.} = P_{об} \cdot t_{об} \cdot Ц_{Э},$$

где $P_{об}$ – мощность, потребляемая оборудованием, кВт; $Ц_{Э}$ – тариф на 1 кВт·час ($Ц_{Э} = 5,782$ руб/кВт·ч); $t_{об}$ – время работы оборудования, час. Время работы оборудования определяется по формуле:

$$t_{об} = T_{рд} * K_t,$$

где $K_t \leq 1$ – коэффициент использования оборудования по времени, равный отношению времени его работы в процессе выполнения проекта к $T_{рд}$, определяется исполнителем самостоятельно. В ряде случаев возможно определение $t_{об}$ путем прямого учета, особенно при ограниченном использовании соответствующего оборудования.

Для выполнения работ студентом использовался персональный компьютер, принтер. Определим затраты на потребленную оборудованием электроэнергию и сведем результаты расчетов в таблицу 6.6.

Таблица 6.6 – Затраты на электроэнергию

Наименование оборудования	Мощность оборудования, кВт	K_t	Время работы оборудования, ч	Затраты на электроэнер- гию
Персональный компьютер	0,35	0,9	1054,3	1920,23
Принтер	0,25	0,005	1240	8,96
Итого				1929,19

6.3.4 Расчет затрат на социальный налог

Затраты на единый социальный налог (ЕСН), который включает в себя отчисления в пенсионный фонд, на социальное и медицинское страхование, составляют 30% от полной заработной платы по проекту и рассчитывается по формуле $C_{\text{соц}} = C_{\text{эл}} * 0,3$. Итак, для проведения исследования затраты на социальный налог составляют $C_{\text{соц}} = 155490,6 * 0,3 = 36234,21$ руб.

6.3.5 Расчет амортизационных расходов

Расчет амортизационных расходов производится по следующей формуле:

$$C_{\text{ам}} = \frac{H_A \cdot Ц_{\text{об}} \cdot t_{\text{рф}} \cdot n}{F_d},$$

где H_A – годовая норма амортизации единицы оборудования; $Ц_{\text{об}}$ – балансовая стоимость единицы оборудования с учетом ТЗР, F_d – действительный годовой фонд времени работы соответствующего оборудования, берется из специальных справочников или фактического режима его использования в текущем календарном году, $t_{\text{рф}}$ – фактическое время работы оборудования в ходе выполнения проекта, учитывается исполнителем проекта; n – число задействованных однотипных единиц оборудования.

При работе над проектом использовались по 1 единице оборудования: персональный компьютер, принтер. Эти два типа оборудования входят в одну группу – вычислительная техника, следовательно, они имеют одинаковый срок полезного использования, равный 2-3 годам. Так как компьютер эксплуатировался более 3 лет уже к началу работ, то срок его полезного использования истек, следовательно, амортизационные расходы на данный вид оборудования равны нулю. Принтер эксплуатировался к началу проведения работ более 5 лет, следовательно, амортизационные расходы на него тоже нулевые.

6.3.6 Расчет прочих (накладных) расходов

В статье «Прочие расходы» отражены расходы на выполнение проекта, которые не учтены в предыдущих статьях, их следует принять равными 10% от суммы всех предыдущих расходов, т.е.

$$C_{\text{проч.}} = (C_{\text{мат}} + C_{\text{зп}} + C_{\text{соц}} + C_{\text{эл.об.}} + C_{\text{ам}}) \cdot 0,1 = (1844,7 + 120780,6 + 36234,21 + 1929,19) \cdot 0,1 = 160788,7 \cdot 0,1 = \mathbf{16078,87 \text{ руб.}}$$

6.3.7 Расчет общей себестоимости разработки

Рассчитав сумму всех расходов на выполнение запланированных работ, произведем расчет общей себестоимости проекта. Результаты расчетов представлены в таблице 6.7.

Таблица 6.7 – Расчет общей себестоимости проекта

Статья затрат	Обозначение	Сумма, руб.
Материалы и покупные изделия	$C_{\text{мат}}$	1844,7
Зарботная плата	$C_{\text{зп}}$	120780,6
Отчисления в социальные фонды	$C_{\text{соц}}$	36234,21
Расходы на электроэнергию	$C_{\text{эл.}}$	1929,19
Прочие расходы	$C_{\text{проч}}$	16078,87
ИТОГО:		176867,57

Таким образом, затраты на разработку составили $C = \mathbf{176867,57}$ руб.

Заключение

В ходе проделанной работы была разработана система для определения координат электрода манипулятора KUKA KR – 16 – 2. Для достижения цели работы были выполнены следующие задачи:

1. Проведен обзор существующих решений в области технического зрения сварочных роботов.
2. Разработана техническая часть в виде конструкции и креплений для электрода и камер.
3. Анализ библиотеки OpenCV, в которую входит как калибровка камеры, так и подбор различных методов для определения контуров объекта.
4. Написан программный код на языке C++ работы системы определения координат.
5. Проведены эксперименты для определения эффективности работы системы.
6. Проводились определение социальной ответственности, расчет финансовой эффективности проекта.

В лаборатории KUKA были проведены тестовые испытания работы, которые показали работоспособность технического решения.

Разработанная система может использоваться на производствах, где используют сварочных роботов. Изменив техническую часть, она так же может использоваться для различных целей технического и компьютерного зрения.

Список использованной литературы

1. Системы технического зрения сварочных роботов, Астахов В. Ю., Ярышкин Н. В., магистранты гр. РТМ-151, II курс Научный руководитель: Курьшкин Н. П., к.т.н., доцент Кузбасский государственный технический университет имени Т. Ф. Горбачёва г. Кемерово.
2. <http://svarkagid.com/svarochnyj-manipuljator/>.
3. <http://www.robotosvarka.ru/tehlichesкое-zrenie-fanuc-irvision>.
4. <https://www.kuka.com/ru-ru>.
5. Техническое зрение FANUC iRVision [Электронный ресурс]. - Режим доступа: <http://www.robotosvarka.ru/tehlichesкое-zrenie-fanuc-irvision> (дата обращения 29.04.18).
6. <https://habr.com/post/253645/>
7. <https://vektor-grupp.ru/articles/1199/>.
8. https://ru.wikipedia.org/wiki/Microsoft_Visual_Studio.
9. <https://docs.opencv.org/3.0-beta/doc/tutorials/tutorials.html>.
10. <https://www.kuka.com/en-us/about-kuka>.
11. <https://new.abb.com/>.
12. LifeCam Cinema [Электронный ресурс]. - Режим доступа: <https://www.microsoft.com/accessories/ru-ru/products/webcams/lifecam-cinema/h5d-00015> (дата обращения 5.05.18).
13. Datta _ _Learning_OpenCV_3_Application_Development, 2016.
14. OpenCV [Электронный ресурс]. - Режим доступа: <https://ru.wikipedia.org/wiki/OpenCV> (дата обращения 5.05.18).
15. Hartley R., Zisserman A. Multiple View Geometry in Computer Vision // Cambridge University Press, 2004.
16. Roger Y. Tsai A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses // IEEE Journal of Robotics and Automation, RA-3(4), 1987. – P. 323–344.

17. Zhang B., Li Y. F. Automatic Calibration and Reconstruction for Active Vision Systems, Intelligent Systems, Control, and Automation // Science and Engineering. V. 57, Springer, 2012. – P. 175.
18. Svoboda T., Martinec D., Pajdla T. A convenient multicamera self-calibration for virtual environments // PRESENCE: Teleoperators and Virtual Environments 14, 4, 2005. – P. 407-422.
19. Camera Calibration [Электронный ресурс].- Режим доступа: https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_calib3d/py_calibration/py_calibration.html#calibration (дата обращения 10.05.18).
20. <http://robot-develop.org/archives/2026>.
21. Компьютерное зрение. Лекция для Малого ШАДа Яндекса [Электронный ресурс].-Режим доступа: <https://habr.com/company/yandex/blog/203136/> (дата обращения 15.05.18).
22. OpenCV шаг за шагом. Поиск объекта по цвету - RGB [Электронный ресурс].-Режим доступа: <http://robocraft.ru/blog/computervision/365.html> (дата обращения 20.05.18).
23. OpenCV шаг за шагом. Нахождение контуров и операции с ними [Электронный ресурс].-Режим доступа: <http://robocraft.ru/blog/computervision/640.html> (дата обращения 20.05.18).
24. Обработка изображений с помощью OpenCV 2016.
25. Practical OpenCV (Technology in Action).
26. mastering-opencv-3-2nd.
27. Learning_Image_Processing_with_OpenCV_proglib.
28. KUKA_KUKA_small_capacity.
29. ГОСТ Р ИСО 26000-2012 Руководство по социальной ответственности.
30. СанПиН 2.2.4.548 – 96. Гигиенические требования к микроклимату производственных помещений. М.: Минздрав России, 1997.
31. СП 52.13330.2016 Свод правил. Естественное и искусственное освещение. Актуализированная редакция СНиП 23-05-95.

32. СН 2.2.4/2.1.8.562 – 96. Шум на рабочих местах, в помещениях жилых, общественных зданий и на территории застройки.

33. СанПиН 2.2.1/2.1.1.1278 – 03. Гигиенические требования к естественному, искусственному и совмещённому освещению жилых и общественных зданий. М.: Минздрав России, 2003.

34. ГОСТ Р 50377-92 (МЭК 950-86) Безопасность оборудования информационной технологии, включая электрическое конторское оборудование.

35. СанПиН 2.2.2/2.4.1340 – 03. Санитарно-эпидемиологические правила и нормативы «Гигиенические требования к персональным электронно-вычислительным машинам и организации работы». – М.: Госкомсанэпиднадзор, 2003.

36. Постановление Правительства РФ от 25.04.2012 N 390 (ред. От 30.12.2017) "О противопожарном режиме" (вместе с "Правилами противопожарного режима в Российской Федерации").

37. СП 12.13130.2009 Определение категорий помещений, зданий и наружных установок по взрывопожарной и пожарной опасности.

Приложение А

Раздел 2

WELDING ROBOTS WITH TECHNICAL VISION

Студент:

Группа	ФИО	Подпись	Дата
8ЕМ61	Лашкарёв Ермек Эдуардович		

Консультант кафедры отделения АйР:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Леонов Сергей Владимирович	к.т.н.		

Консультант – лингвист кафедры ОИЯ ШБИП:

Должность	ФИО	Подпись	Дата
Ст. преподаватель	Горбатова Татьяна Николаевна		

WELDING ROBOTS WITH TECHNICAL VISION

Examples of welding robots with technical vision

At present, the most popular industrial robots are welding robots. Welding robots allow you to optimize the entire welding process. As mentioned above, it is very convenient and profitable, but there are also disadvantages. The main disadvantage is a time, when robot is welding a part. In a conventional manipulator, the accuracy of positioning falls when the seam is displaced, where a person in turn might guess, but not a robot. In large companies, there are various solutions to such situations. The examples of welding robots with technical vision will be given below [1].

Welding robot of company Fanuc Corporation

Fanuc Corporation is one of the leading manufacturers of automation equipment. The company is very popular and has many staffs around the world. The main areas are CNC, industrial robots and laser equipment.

The company is very popular thanks to a series of its welding robots with technical vision iRVision. The manipulator with the technical vision iRVision of Fanuc Corporation is presented in figure 1 and in figure 2 [3].



Figure 1 – Fanuc Corporation Manipulator with Vertical Camera



Figure 2 – Fanuc Corporation’s manipulator with built-in camera

Let's look at some of the features of the iRVision system used in robotic welding [5]:

1. VISION SHIFT is used in contact and arc welding to compensate for the error between the positioning of the template part by which the robot was programmed and the positioning of subsequent parts on which the program is to be executed.

Features of VISION SHIFT:

- the function uses the technical vision capabilities for automatic 3D calibration of the robot program;
- it is possible to automatically calibrate the tool's working point (TCP), which previously could only be done manually;
- for high-precision non-contact calibration programs instead of a few hours takes only up to 30 minutes. Thus, this function significantly shortens the start-up time;
- supports the installation of the central point of the instrument and the tip of the electrode to control the robot with great accuracy.
- can be used to calibrate groups of cooperating robots (Dual-, Triple-, and Quadarm);
- allows you to do the alignment of the FANUC robots after the change of motors, encoders, etc.

2. FANUC iRVISION Weld Tip serves to visually inspect and check the condition of the working end of the electrode for spot welding for damage. This specialized software can perform a wide variety of checks, including measuring the wear of the detachable end of the electrode and determining whether it has been replaced or re-filled.

3. FANUC iRVISION Torch Mate is equipped with programming capabilities in production and industrial camera. It prevents the occurrence of alignment problems by keeping the tool center point on the toolpath. The iRVISION system ensures a stable welding quality, and also increases the uptime by eliminating the need for pre-emptive actions.

4. 2D Multiview Vision - technical vision with the use of several cameras to determine the exact position of the part. Allows you to accurately determine the position of large parts.

The R30-iA controller in its software contains an Arctool application designed to simplify and standardize the installation and management of welding applications. Let's consider some functions of the application:

- weaving with oscillations. Provides oscillatory motion of the torch relative to the welding line, increasing the required thickness of the weld;

- weld Resume - when the robot stops while driving through the welding track and moves back to the track, the required distance is set, and welding continues.

- electrode protection - compensates for the value of the tip of the electrode with a wire after a long time of welding or replacing the electrode tip. In these cases, welding operations can not be performed correctly. Protecting the electrode helps prevent the center point of the instrument from failing (TCP), otherwise it must be overwritten;

- automatic adjustment of the tool center point - the automatic burner recovery function automatically adjusts the center point of the instrument (TCP). Automatic electrode restoration automatically compensates for the welding electrode bend and tip wear, reducing the number of defects and increasing productivity;

- torch guard collision detection - the parameters for the wrist are very sensitive to electrode collisions. The function stops the robot until the electrode, after colliding with the obstacle, breaks;

- arc sensor TAST - (at the time of motion along the seam) the seam control method used to control the position of the surface of the part, which is very important for robotic arc welding. This method is used as a welding current control sensor in the event of a change in arc length. The robot will compensate for the change in the position of the welding torch;

- monitoring the position of the TAST burner allows you to monitor the position of the electrode by changing the welding current for amplitude motion. The program determines the center of the seam, comparing the left and right halves by current. It is also possible to control the height of the electrode position. The electrode position is set (according to the current value) and is stored in the middle of the amplitude;

- touch sensing consists of: a) movement of the central point of the instrument (TCP) to the object, using robot motion control, speed and direction; b) use of incoming signals to determine the beginning of the contact of the robot with the object; c) preserving the object's position in the position registers; d) use of information about stored positions by the robot. Touch Sensing determines the position of deflection of welding seams at the time of welding with the help of Arc Sensor TAST. Touch sensing allows the robot to change the direction of the track automatically when moving in the event that the surface of the part is inaccurate, or the part occupies an incorrect position.

- the Automatic Voltage Control (AVC) option compensates for the variation in the shape of the part and the deviation during the TIG welding process (tungsten electrode welding in an inert gas) or other DC processes, for example, when performing plasma cutting. AVC compensates for the difference between the optimum and the actual welding parameters during the movement along the welding track. The AVC function demonstrates the voltage during arc welding of the TIG process. It allows you to change the height of the burner over the piece to maintain a constant welding voltage [3].

Welding robot of company ABB Robotics

ABB Robotics is a company that manufactures industrial equipment. Including robotic complexes. Although the company was founded in 1988, it is already one of the leaders in the world. The company has many different manipulators: for assembly and disassembly, palletizing, spot and arc welding, varnishing and painting, and others. One of the robots is the manipulator for arc welding IRB 1520ID. The new IRB 1520ID, a high-performance robot with built-in welding equipment, designed specifically for arc welding. All cables and feed channels are laid inside the manipulator, which simplifies programming and provides optimal protection for all elements. The IRB 1520ID is shown in Figure 3 [11].



Figure 3 – Electric arc welder of ABB Robotics Company

The company is researching technical vision. One example is the robot Roberta. Robots called Roberta, have been specifically designed for small and medium-sized enterprises who want to achieve efficient industrial automation. The goal of the developers was to create flexible, lightweight and ergonomic robots that can be easily moved around the manufactory [11].

Key Features of Roberta:

- a light weight;
- 6 degrees of freedom without a point of singularity;
- heavy lifting capacity.

Roberta robots present several integrated security concepts. Roberta can work with a certain capture, which is absolutely safe for humans. Special sensors, another aspect of safety, which reduces the risk of any damage. The robot is also equipped with technical vision. That is, he can see the object he is working with. The Roberta series is available in 3 different models with a load capacity of 4.0 kg, 8.0 kg and 12.0 kg. Differences between models - in maximum reach and carrying capacity.

Welding robot of company KUKA Roboter

Industrial robots Kuka - one of the most popular robot welders in the world. They successfully replace manual labor in many European enterprises. For welding works, the company has both high-capacity models and low-capacity welding robots. Let's list some models of welding robots Kuka [10]:

- model KR 5 arc is recommended when relatively simple robots are needed for welding (arc), soldering, painting and other manipulations. Its useful load is 5 kg, the additional load is 12, and the maximum range is 1412 mm. The model is shown in figure 4;



Figure 4 – Model KR 5 arc

- robotized welding complex in the environment of protective gases allows to work even with very complex details of large dimensions, as well as at great depth of parts. Various sets of hoses are used to feed the gas. This is the KR 5-2 arc HW model with the same weight and working radius parameters as the previous one. Such robots can be mounted both on the floor and on the ceiling,

effectively performing soldering and welding, including electric arc welding. This also includes KR 16 L8 arc HW [10];

- KR 16 L8 arc HW enters the L series together with the KR 16 L6-2. The extended manipulator allows to increase the working area up to 2016 mm and 1911 mm, respectively, but the load capacity is reduced (8 and 6 kg);
- a multifunctional industrial robot represented by models KR 6-2 and KR 16-2 (in the Kuka catalog is designated as KR 6-3 and KR 16-3) can be used for a variety of operations, which can be tuned for dozens of different operations. The first model has a payload of 6 kg and an additional load of 10, with a maximum radius of 1611 mm. The second model has the same limits of additional load, but the useful is 16 kg, and the working area is 1610 mm;
- For high-speed work where increased power is required, the robot KR 16-3 S is created, whose time of the cycle is reduced by 18%. It has a load capacity of 16 kg and a reach of 1611 mm [7].

There are also other models with different capabilities. For light vehicles, the cargo weight can be from 5 to 16 kg, and the maximum load within the range of 36-48 kg. All these robots can be equipped with cameras to perform work related to technical sight. The company KUKA Roboter has a number of solutions in this area, but these solutions are not in the public domain.

Technical part

The methods for solving the problem were analyzed. At first the possible solutions in articles related to technical vision, or rather systems for determining objects and their coordinates, have been studied. However, as such there were no solutions.

To determine the coordinates of the electrode, there was a place for them and for the manipulator. In this connection, a design was invented that would allow to take pictures of the manipulator electrode. The G-shaped structure is shown in figure 5.

This design was modeled in Autodesk Inventor. As you can see the lower part of the construction is 31.8 cm x 31.8 cm. The dimensions for the plywood sheets are taken under A4: 21.8 cm and 31.8 respectively. For each sheet taken safety margin: 0.8



Figure 6 – Robotic cell for milling KUKA KR 120 R2700

The technical part also includes 2 Microsoft webcams «LifeCam Cinema» 6CH-00002. These cameras are very sensitive to light and have a bending part for fastening. Technical characteristics of the Microsoft webcams «LifeCam Cinema» 6CH-00002 are presented in table 1. The camera is shown in figure 7 [12].

Table 1 – Technical characteristics of Microsoft «LifeCam Cinema»

Supported video resolutions	1280x720
Max frame rate	30 Hz
Type of connection	USB 2.0
Accommodation	Mounting on the LCD monitor
Type of matrix	CMOS
Number of megapixels of the matrix	0.9
Supported photo resolutions	1280x960
Focusing	Auto Focus
Connectors and interfaces	USB Type A plug (USB 2.0)
Housing material	Plastic
Dimensions (WxHxD), mm	46x40x56
Weight, g	95



Figure 7 – Webcams Microsoft «LifeCam Cinema» 6CH-00002

After working with the milling manipulator, pieces of cut plywood assembled the G-structure. For fixing small screws were used, because on the surface the plywood should not have unevenness. In addition, small screws made it possible to place the construction on the manipulator table without problems, because they could not be seen, and the construction was flat.

After the pieces of plywood were assembled together, it remained to fix them on the table. The laboratory used fasteners, which were already pre-designed for the manipulator tables. These fastenings are very stable to reliable. The type of the received construction is shown in figure 8. The figure shows the very yellow fasteners behind the plywood. Also a view of the design on the background of the welding manipulator KUKA KR - 16 - 2 is shown in figure 9.

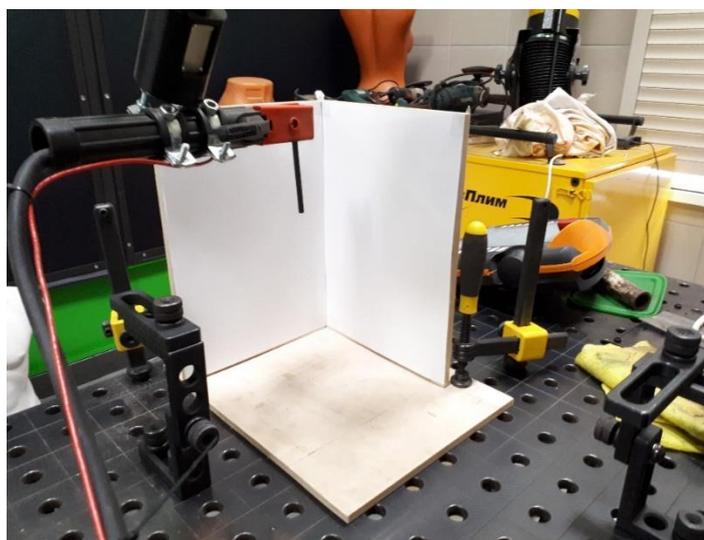


Figure 8 – Assembled plywood construction

In addition, opposite to each of the plywood is fixed G-shaped metal fasteners. They are needed for the stability of Microsoft's "LifeCam Cinema" cameras. The location of the cameras is shown in figure 10. The cameras themselves are very sensitive to external influences. Figure 11 shows a construction with 2 cameras.

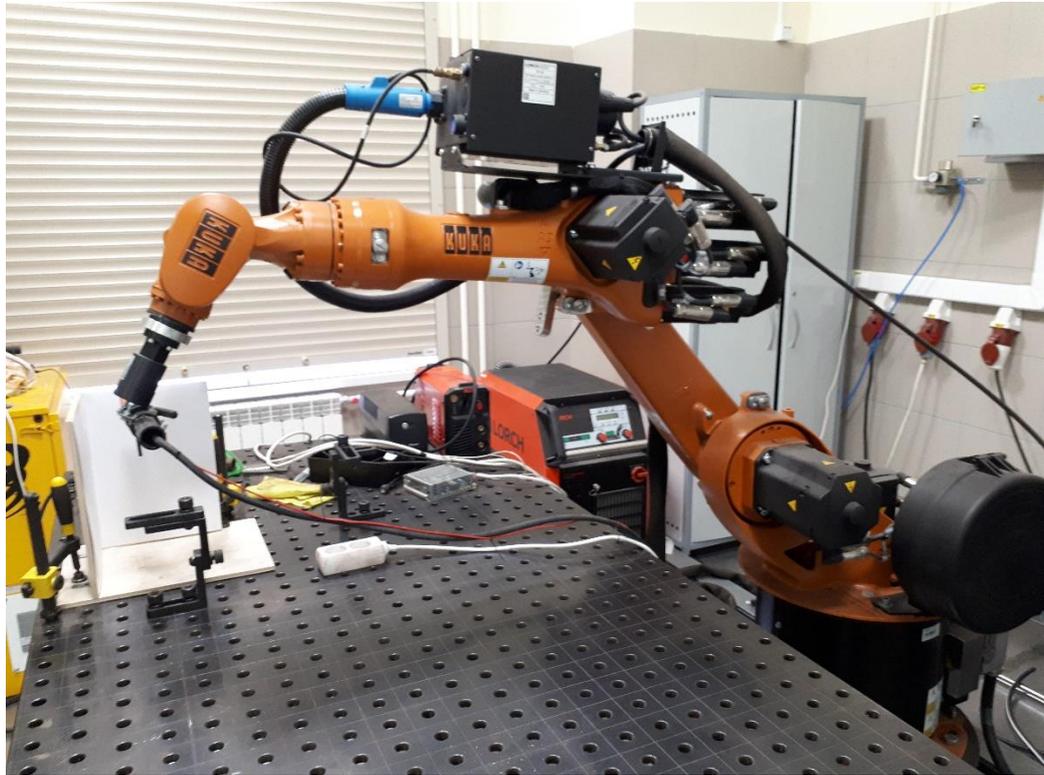


Figure 9 – Type of construction against welding manipulator KUKA KR – 16 – 2

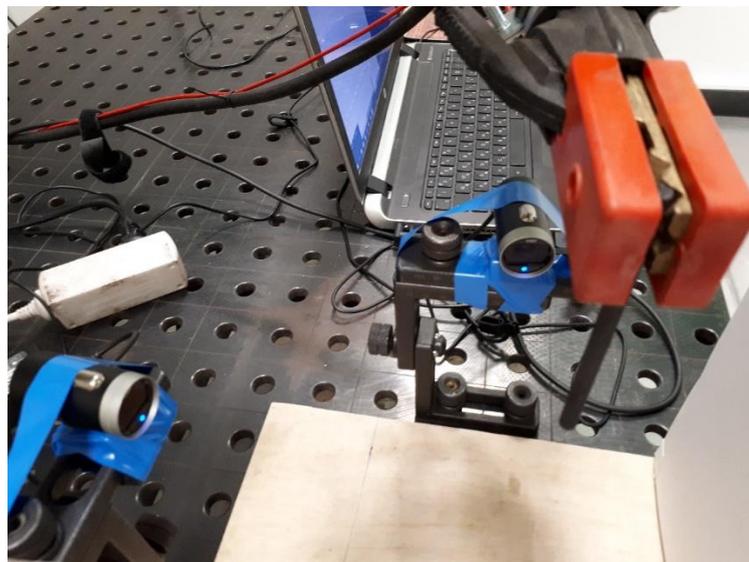


Figure 10 – Cameras Microsoft, mounted on fixtures

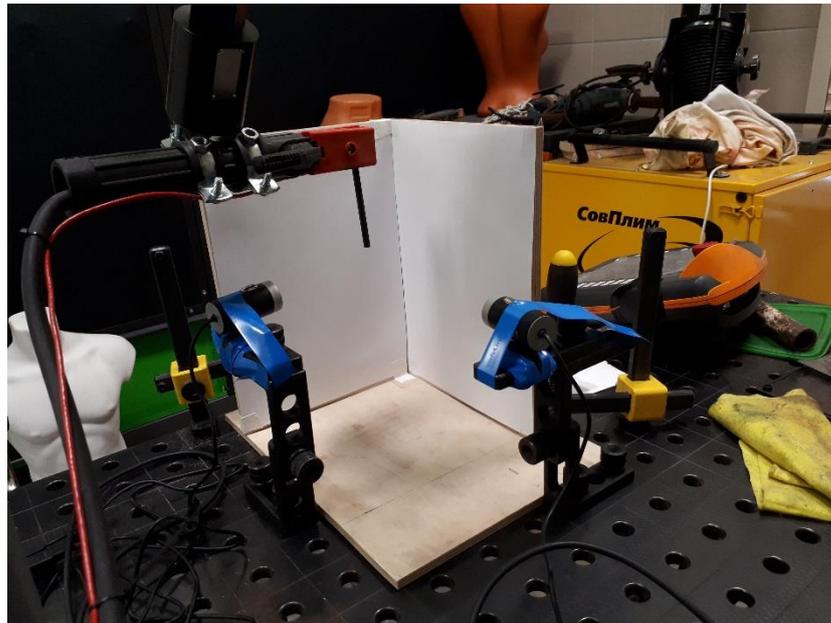


Figure 11 – Construction of plywood with cameras

Also in figure 12, the electrode of the welding manipulator is visible. Automatic setting, setting the motion of the manipulator using the control panel. In addition, in the figure you can see A4 sheets, fixed on each side of the plywood. Lighting varies throughout the day. Because of this, shadows appear, glare on the image itself, which is very difficult to solve. One of the solutions is mandatory lighting in the form of a searchlight, but unfortunately, this technique is costly. A4 sheets are white, which makes it easier to search for objects, in our electrode case.



Figure 12 – Welding arm electrode

For a more intuitive view of the manipulator against the background of the assembled structure with fixed cameras and working area, figure 13 is presented. During operation, the initial position of the electrode was set manually. After that, work was done with the adjustment of the cameras and their calibration. During operation, the manipulator was turned off for safe operation. As you can see in figure 13, the cameras are connected to the laptop. In the notebook itself, the adjustment of the software part in C++ is performed. The data received from the cameras are stored in the root folder. After each picture, the old image is deleted and replaced with a new image of the electrode. In parallel, you can change the camera positions for easy viewing. If you can not adjust the size of the images received from the camera, they can be trimmed using the built-in commands of the OpenCV library.

The work of the program part begins after all the settings of the cameras, and also after the cameras are placed in a fixed position. If the settings are not fulfilled, the program part will have errors when finding the contours and electrode coordinates.

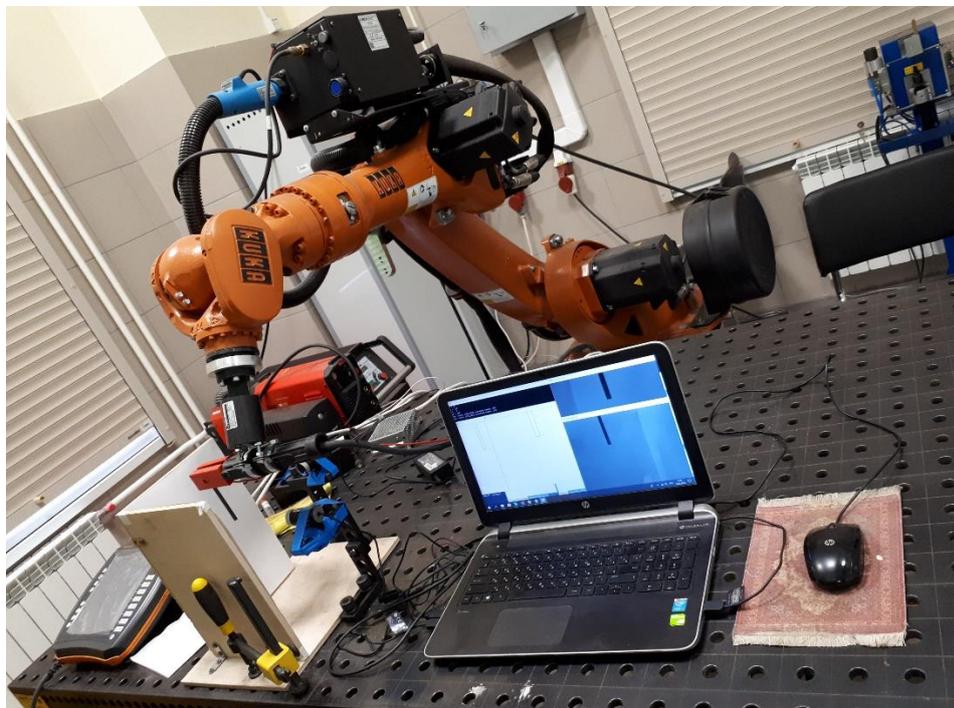


Figure 13 – The view of manipulator on a background of construction

Приложение Б

Листинг программного кода для определения координат электрода

```
#include <opencv2\opencv.hpp>
#include<locale.h>
using namespace cv;
using namespace std;
int k, l, m, n = 0;
void onMouse(int event, int x, int y, int flags, void* param)
{
    cv::Mat *im = reinterpret_cast<cv::Mat*>(param);
    switch (event) {
        case cv::EVENT_LBUTTONDOWN: // функция движение мыши вниз
            // вывод пиксельных координат в точке (x,y)
            std::cout << "при точках (" << x << ", " << y << ") значение равно: "
                << static_cast<int>(im->at<uchar>(cv::Point(x, y))) << std::endl;
            break;
    }
}
int main(int, char**) {
    setlocale(LC_ALL, "RUS");
    // снимок с первой камеры
    vector<int> wr_params;
    wr_params.push_back(cv::IMWRITE_JPEG_QUALITY);
    wr_params.push_back(50);
    VideoCapture cap(2);
    if (!cap.isOpened())
        return -1;
    Mat edges;
    Mat frame;
    cap >> frame;
    //сохранение снимка 1
    imwrite("electrode_left.jpg", frame, wr_params);
    // снимок со второй камеры
    VideoCapture cap1(1);
    if (!cap1.isOpened())
        return -1;
    Mat edges1;
    Mat frame1;
    cap1 >> frame1;
    //сохранение снимка 2
    imwrite("electrode_right.jpg", frame1, wr_params);
    // чтение снимка 1 электрода
    Mat img0 = imread("electrode_left.jpg", 1);
    int x = 325,
        y = 25,
        width = 200,
        height = 455;
    img0 = img0(Rect(x, y, width, height));
    Mat img1;
    cvtColor(img0, img1, CV_RGB2GRAY);
    // добавление фильтра
    Canny(img1, img1, 50, 100);
    // нахождение контуров
    vector< vector<Point> > contours;
    findContours(img1, contours, CV_RETR_EXTERNAL, CV_CHAIN_APPROX_NONE);

    Mat mask = Mat::zeros(img1.rows, img1.cols, CV_8UC1);
    drawContours(mask, contours, -1, Scalar(255), CV_FILLED);
    Mat crop(img0.rows, img0.cols, CV_8UC3);
    // установка заднего фона белым цветом
    crop.setTo(Scalar(255, 255, 255));
}
```

```

// копирование crop
img0.copyTo(crop, mask);

normalize(mask.clone(), mask, 0.0, 255.0, CV_MINMAX, CV_8UC1);
// вычисление пиксельных координат
for (int i = 0; i < crop.rows; i++)
    for (int j = 0; j < crop.cols; j++)
        if (crop.at<Vec3b>(i, j)[0] < 255 &&
            crop.at<Vec3b>(i, j)[2] < 255 &&
            crop.at<Vec3b>(i, j)[1] < 255)
            {
                crop.at<Vec3b>(i, j)[0] = 0;
                crop.at<Vec3b>(i, j)[1] = 0;
                crop.at<Vec3b>(i, j)[2] = 0;
                k = i;
                l = j;
                i = crop.rows;
                break; }
cout << "x = " << k << endl << "y = " << l << endl;
imshow("electrodeL_canny", crop);
imwrite("electrodeL_canny.jpg", img1);
imwrite("electrodeL_mask.jpg", mask);
// читаем снимок 2 электрода
Mat img2 = imread("electrode_right.jpg", 1);
Mat img3;
cvtColor(img2, img3, CV_RGB2GRAY);
// добавление фильтра
Canny(img3, img3, 50, 100);
// нахождение контуров
vector< vector<Point> > contours1;
findContours(img3, contours1, CV_RETR_EXTERNAL, CV_CHAIN_APPROX_NONE);
Mat mask1 = Mat::zeros(img3.rows, img3.cols, CV_8UC1);
drawContours(mask1, contours1, -1, Scalar(255), CV_FILLED);
Mat crop1(img2.rows, img2.cols, CV_8UC3);
// установка заднего фона белым цветом
crop1.setTo(Scalar(255, 255, 255));
img2.copyTo(crop1, mask1);
normalize(mask1.clone(), mask1, 0.0, 255.0, CV_MINMAX, CV_8UC1);
// вычисление пиксельных координат
for (int i = 0; i < crop1.rows; i++)
    for (int j = 0; j < crop1.cols; j++)
        if (crop1.at<Vec3b>(i, j)[0] < 255 &&
            crop1.at<Vec3b>(i, j)[2] < 255 &&
            crop1.at<Vec3b>(i, j)[1] < 255)
            {
                crop1.at<Vec3b>(i, j)[0] = 0;
                crop1.at<Vec3b>(i, j)[1] = 0;
                crop1.at<Vec3b>(i, j)[2] = 0;
                m = i;
                n = j;
                i = crop1.rows;
                break; }
cout << "x1 = " << m << endl << "y1 = " << n << endl;
// показ изображений
imshow("electrodeR_canny", crop1);
imshow("electrode_right", frame1);
imshow("electrode_left", frame);
setMouseCallback("electrodeL_canny", onMouse, reinterpret_cast<void*>(&crop));
setMouseCallback("electrodeR_canny", onMouse, reinterpret_cast<void*>(&crop1));
waitKey();
return 0;
}

```