

МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ  
КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Ордена Трудового Красного Знамени  
федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский технический университет связи и информатики»



На правах рукописи

Белов Никита Вадимович

**МЕТОДЫ И АЛГОРИТМЫ СОРТИРОВКИ ФЕРРОМАГНИТНЫХ  
ДЕТАЛЕЙ ПРОМЫШЛЕННЫМ МАНИПУЛЯТОРОМ С  
ИСПОЛЬЗОВАНИЕМ КОМПЬЮТЕРНОГО ЗРЕНИЯ**

Специальность: 2.5.4. - Роботы, мехатроника и робототехнические системы

Диссертация на соискание ученой степени

кандидата технических наук

Научный руководитель:

доктор физико-математических наук, профессор

Воронова Л.И.

Москва – 2025

## Оглавление

ВВЕДЕНИЕ.....	4
ГЛАВА 1. ИССЛЕДОВАНИЕ МЕТОДОВ УПРАВЛЕНИЯ ПРОМЫШЛЕННЫМ МАНИПУЛЯТОРОМ С ПРИМЕНЕНИЕМ ТЕХНОЛОГИИ КОМПЬЮТЕРНОГО ЗРЕНИЯ.....	15
1.1. Адаптивное и интеллектуальное управление роботами .....	15
1.2. Анализ систем управления промышленными роботами .....	22
1.3. Анализ исследований в области распознавания объектов в промышленной робототехнике.....	32
1.4. Методы определения пространственных координат объекта.....	35
Выводы по главе.....	38
ГЛАВА 2. РАЗРАБОТКА ПОДСИСТЕМЫ ДИСТАНЦИОННОГО УПРАВЛЕНИЯ ПРОМЫШЛЕННЫМ МАНИПУЛЯТОРОМ .....	40
2.1. Архитектура подсистемы дистанционного управления промышленным манипулятором.....	40
2.2. Программные модули на внешнем вычислительном устройстве и контроллере робота.....	48
2.3. Дистанционное управление промышленным манипулятором .....	54
Выводы по главе.....	57
ГЛАВА 3. РАЗРАБОТКА МЕТОДА КЛАССИФИКАЦИИ И ЗАХВАТА МАНИПУЛЯТОРОМ ПРОИЗВОЛЬНО РАСПОЛОЖЕННЫХ ФЕРРОМАГНИТНЫХ ДЕТАЛЕЙ ИЗ НАКОПИТЕЛЯ.....	59
3.1. Этапы метода классификации и захвата манипулятором произвольно расположенных ферромагнитных деталей из накопителя.....	59
3.2. Выбор камеры бинокулярного зрения и ее размещение в рабочей зоне робота .....	63
3.3. Определение границ накопителя в рабочей зоне манипулятора .....	66
3.4. Нейросетевая классификация объектов в накопителе .....	70
3.4.1.Способ автоматического формирования дообучающего набора данных.....	74

3.5. Методика определения объекта возле стенок накопителя .....	79
3.6. Алгоритм определения пространственных координат объекта.....	81
3.7. Магнитный пневматический захват для промышленного манипулятора .. .....	96
Выводы по главе.....	101
ГЛАВА 4. ЭКСПЕРИМЕНТАЛЬНЫЕ ИССЛЕДОВАНИЯ. ОБСУЖДЕНИЕ ПОЛУЧЕННЫХ РЕЗУЛЬТАТОВ .....	103
4.1. Автоматизированная сборка обучающего набора данных и дообучение нейронной сети.....	103
4.2. Тестирование алгоритма определения пространственных координат объекта .....	109
4.2.1.Тестирование способов определения координат центра объекта .....	109
4.2.2.Тестирование алгоритма определения углов наклона объекта захвата .....	115
4.3. Комплексное тестирование решения для сортировки ферромагнитных деталей промышленным манипулятором с использованием компьютерного зрения 3D зрения и нейросетевой модели.....	120
Выводы по главе.....	123
Заключение .....	125
СПИСОК ЛИТЕРАТУРЫ.....	127
Приложение А. Результаты экспериментов преобразования координат центра объекта.....	141
Приложение Б. Результаты экспериментов определения пространственных координат объекта.....	151
Приложение В. Результаты экспериментов решения для сортировки ферромагнитных деталей промышленным манипулятором.....	155
Приложение Г. Патент на полезную модель.....	158
Приложение Д. Акты внедрения .....	159

## ВВЕДЕНИЕ

### Актуальность темы диссертации

При цифровизации производства в Российской Федерации внедрение промышленных роботов становится одним из ключевых аспектов оптимизации технологических процессов. Использование сквозных цифровых технологий, таких как Интернет вещей, искусственный интеллект и большие данные, позволяет интегрировать роботов в существующие производственные линии, что ведет к значительному улучшению производительности, увеличению объемов производства и минимизации ошибок, связанных с человеческим фактором. Существенный вклад в развитие теории робототехнических систем различного назначения внесли многие российские ученые: Глазунов В.А., Каляев И.А., Макарова И.М., Юревич Е.И., Кулешов В.С., Зенкевич С.Л., Ющенко А.С.

Согласно анализу Национальной Ассоциации участников рынка робототехники России в 2023 году одной из основных задач является существенное расширение функциональности промышленных роботов, в том числе связанное с перемещением материалов. При этом предполагается, что объекты могут быть разной формы, находится в заранее неопределенном положении и ориентации, а также находятся в ограниченном пространстве, например в накопителе. В настоящее время на производствах используются специализированные транспортно-накопительные и загрузочные устройства, позволяющие подавать детали роботу в строго определенной ориентации и в заранее фиксированном положении, используя для этого накопители с ложементами. В случае невозможности использования подобных устройств предприятия применяют ручной труд для рассортировки накопителя, что приводит к предельным нагрузкам работников, связанных с многократным подъемом и перемещением тяжестей.

Задача интеллектуальной сортировки, связанная с распознаванием произвольно расположенных разнородных деталей в накопителе с помощью машинного зрения, остается не решенной учеными в течении последних нескольких десятилетий по двум причинам:

- производители роботов ориентированы на работу с помощью ручных пультов управления, поддерживающих только специализированные роботоориентированные языки программирования и позволяющие писать программы управления манипулятором для перемещения в заранее заданные координаты;

- использующиеся в робототехнике системы компьютерного зрения управляются отдельным контроллером, являются проприетарными и используют классические методы машинного зрения поиска по шаблону и ключевым точкам, позволяющие распознавать объекты только в двумерном 2D пространстве.

Известны разработки российских и зарубежных ученых, таких как Hyun Jun Jo, Chen Yang, Н.А. Андриянов и др, посвященные сортировке объектов из накопителя. Однако, при этом существует ряд проблем, связанных: с нестабильностью захвата в сложных сценах; столкновением фланца робота с другими объектами при захвате; нахождением объекта возле стенки контейнера.

Для разработки интегрированных с манипулятором систем компьютерного зрения, позволяющих решать задачу сортировки и перемещения разнородных деталей, расположенных «навалом», необходимо использовать методы интеллектуального анализа данных и работать с 3D данными, что возможно только при обработке информации не на контроллере робота, а на удаленном устройстве, поддерживающего высокоуровневые языки программирования. Для связи контроллера манипулятора и удаленного устройства необходима система дистанционного управления, обеспечивающая отправку управляющих программ, содержащих результаты обработки рабочей зоны манипулятора с помощью компьютерного зрения в потоковом режиме.

### **Степень разработанности темы исследования**

Анализ современной научной литературы и практических разработок показывает, что область управления промышленными манипуляторами с использованием компьютерного зрения является активно исследуемой, но при этом сохраняет ряд нерешенных проблем. В работах российских и зарубежных ученых рассматриваются вопросы адаптивного и интеллектуального управления роботами,

включая задачи сортировки объектов. Однако существующие решения часто ограничены использованием проприетарных систем, работающих в 2D-пространстве, и не обеспечивают достаточной гибкости для работы с произвольно расположенными объектами.

Особое внимание в исследованиях уделено проблемам, связанным с нестабильностью захвата и столкновением фланца робота с другими объектами, а также сложностью определения пространственных координат объектов, особенно вблизи стенок накопителя. Разработки в области компьютерного зрения, такие как методы на основе нейронных сетей и использование RGB-D камер, позволяющие получить трехмерное изображение рабочей зоны манипулятора, что позволяет устранить неоднозначности, связанные с масштабом, окклюзиями и текстурой, требуют дальнейшего совершенствования для повышения точности и надежности в промышленных условиях.

Существующие системы управления промышленными роботами, таких производителей как KUKA, Fanuc и ABB, повсеместно используют специализированные языки программирования и ограничены в возможностях интеграции с внешними устройствами. Это затрудняет разработку адаптивных алгоритмов, основанных на обработке данных в реальном времени. В то же время, современные подходы, такие как задание команд управления с внешнего устройства и использование внешних планировщиков траектории, открывают новые перспективы, но их внедрение требует решения проблем совместимости с оборудованием различных производителей.

Несмотря на определенный прогресс в области робототехники и компьютерного зрения, задача создания универсальной, гибкой и высокоточной системы для сортировки ферромагнитных деталей промышленным манипулятором остается нерешенной.

Создание новых эффективных методов и алгоритмов, позволяющих решить выделенные проблемы, являются актуальными задачами исследования.

**Объект исследования** – система управления промышленным манипулятором на основе компьютерного зрения.

**Предмет исследования** – методы и алгоритмы автоматического распознавания, захвата и сортировки произвольно расположенных объектов в накопителе промышленным роботом.

**Целью диссертационной работы** является повышение эффективности и расширение функциональных возможностей промышленных манипуляторов для сортировки ферромагнитных деталей путем использования разработанных методов, алгоритмов и программно-аппаратных средств, обеспечивающих захват произвольно расположенных объектов в сложных сценах с использованием дистанционного управления, нейросетевых методов и компьютерного зрения.

**Задачи диссертационной работы:**

1. Разработка архитектуры подсистемы дистанционного управления промышленным манипулятором, позволяющей осуществлять коммуникацию контроллера робота и удаленного устройства с помощью сетевых протоколов и обеспечивающей двухстороннюю связь с системой компьютерного зрения в потоковом режиме.
2. Разработка метода классификации и захвата манипулятором произвольно расположенных разнородных ферромагнитных деталей.
3. Разработка алгоритма, обеспечивающего определение пространственных координат объекта, на основании 3D данных.
4. Разработка конструкции пневматического захватного устройства, обеспечивающего надежный захват ферромагнитных объектов и позволяющего компенсировать неточности определения координат центра объекта вдоль вертикальной оси.
5. Разработка модульной архитектуры системы управления промышленным манипулятором, обеспечивающей дистанционное задание управляющих программ и решение задачи адаптивного захвата случайно расположенных разнородных объектов с использованием системы компьютерного зрения.
6. Экспериментальное исследование эффективности разработанных метода и алгоритмов автоматического распознавания, захвата и сортировки

произвольно расположенных объектов в накопителе промышленным роботом в лабораторных условиях.

**Научная новизна исследования заключается в следующем:**

1. Предложена модульная архитектура системы управления манипулятором для сортировки, содержащая подсистему удаленного управления и методы компьютерного 3D зрения. **Показана** возможность за счет нейросетевых и методов компьютерного зрения разрабатывать адаптивные алгоритмы управления робототехническими системами.

2. Разработана архитектура подсистема дистанционного управления промышленным роботом, в которой для обеспечения двухсторонней связи и передачи команд управления в потоковом режиме между контроллером робота и удаленным устройством, используется многопоточность или асинхронное программирование. **Показана** возможность в автоматическом режиме передавать управляющие программы, полученные с помощью компьютерного зрения на контроллер робота.

3. Разработан метод классификации и захвата произвольно расположенных разнородных деталей в накопителе, включающий: определение границ накопителя в рабочей зоне манипулятора методом конкурного анализа, нейросетевой метод сегментации объектов, выбор объекта захвата по данным RGB-D сенсора, алгоритм определения пространственных координат объекта, движение манипулятора и управление захватным устройством. При этом в методе **предусмотрена** возможность захвата объектов, расположенного возле стенки накопителя. По результатам **лабораторных экспериментов установлено**, что комбинирование методов компьютерного зрения позволяет работать в сложных сценах и увеличить стабильность захвата объекта.

4. Разработан алгоритм определения пространственных координат объекта на основе интеграции нейросетевого метода и облака точек; на первом этапе алгоритма выполняется определение центра объекта захвата на основе фокусного расстояния камеры и расстояния до объекта по вертикальной оси, а на втором определение наклона объекта относительно манипулятора на основе

сравнения 3D модели объекта в формате полигональной сетки и облака точек. В алгоритме **предусмотрена** возможность определения 3D координат в случае частичного перекрытия объекта. По **результатам лабораторных экспериментов** подтверждено, что полученная точность достаточна для безошибочного захвата объекта.

5. Разработана конструкция пневматического магнитного захвата, позволяющая осуществлять захват ферромагнитных изделий, без смещения центра тяжести, относительно фланца робота, а также содержащего подпружиненный демпфер, который увеличивает допустимую погрешность сближения с захватываемой деталью. Такое решение **позволяет** предотвратить возможные аварии при захвате изделия, габариты/ориентация которого отличны от установленных, с минимальным понижением исходной грузоподъемности робота.

**Теоретическая значимость работы** заключается в развитии научного направления создания интеллектуальных систем управления промышленными манипуляторами, с использованием адаптивных алгоритмов управления, нейросетевых методов и методов компьютерного зрения.

**Практическая значимость работы** заключается в разработке новых архитектурных решений и алгоритмов, а также в создании подходов к автоматизированному формированию наборов данных, нейросетевой сегментации и точному определению пространственных координат и ориентации произвольно расположенных объектов в сложных сценах.

1. Архитектура подсистемы дистанционного управления промышленным роботом, позволяет реализовать децентрализованное управление и производить обработку информации не на контроллере робота, а на удаленном устройстве без необходимости модернизации аппаратного обеспечения самого робота, а также повышает гибкость, масштабируемость и адаптивность системы за счёт распределённой обработки информации.

2. Метод классификации и захвата произвольно расположенных разнородных деталей в накопителе для захвата промышленным манипулятором, позволяет снизить процент неудачного схвата объектов (более чем на 15% по

сравнению с существующими), за счет возможности определения расположения объектов возле стенки накопителя.

3. Автоматизированный способ сборки набора данных для обучения нейронной сети позволяет сократить временные затраты, а также позволяет повысить качество обучения и улучшить результаты работы нейронных сетей, за счет исключения ошибок, связанных с человеческим фактором, при подготовке набора данных

4. Алгоритм, обеспечивающий определение пространственных координат объекта, позволяет сократить погрешности определения центра объекта на 0,5 миллиметра и сократить погрешность определения положения объекта более чем на  $2^\circ$  по сравнению с существующими методами

5. Разработанная конструкция пневматического магнитного захвата позволяет увеличить допустимую погрешность сближения с захватываемой деталью, что позволяет использовать его в системах, которые имеют погрешность вычисления расстояния до объекта по вертикальной оси.

**Методы исследования.** В диссертационной работе применены методы теории проектирования робототехнических систем, теории автоматического управления, математической статистики и математического анализа, методы планирования эксперимента, методы интеллектуального анализа данных, в том числе компьютерного зрения. Для научно-практической реализации результатов применялись современные методы объектно-ориентированного программирования, императивного и функционального программирования.

#### **Положения, выносимые на защиту:**

1. Модульная архитектура системы управления промышленным манипулятором для сортировки ферромагнитных деталей, включающая подсистему удаленного управления с реализацией методов двусторонней связи и параллельной обработки команд, а также подсистему компьютерного 3D-зрения, обеспечивающую определение пространственного положения объектов за счет комбинации RGB-D сенсоров и нейросетевых алгоритмов, специально

оптимизированных для задач сегментации ферромагнитных деталей в условиях производственной среды.

2. Архитектура подсистемы дистанционного управления промышленным роботом, обеспечивающая двухстороннюю связь и передачу команд управления в потоковом режиме между контроллером робота и удаленным устройством и позволяющая в автоматическом режиме передавать управляющие программы, полученные с помощью компьютерного зрения на контроллер робота.

3. Метод классификации и захвата манипулятором произвольно расположенных разнородных деталей из накопителя, обеспечивающий повышение стабильности захвата за счет учета возможности расположения объектов возле стенки накопителя, а также применению разработанной конструкции захватного устройства, позволяющего учитывать погрешность сближения с захватываемой деталью.

4. Алгоритм определения пространственных координат объекта на основе интеграции нейросетевого метода и облака точек; на первом этапе алгоритма выполняется определение центра объекта захвата на основе фокусного расстояния камеры и расстояния до объекта по вертикальной оси, а на втором определении наклона объекта относительно манипулятора на основе сравнения 3D модели объекта в формате полигональной сетки и облака точек.

### **Реализация и внедрение.**

1. Результаты диссертационной работы используются в учебном процессе в кафедры «Интеллектуальные системы в управлении и автоматизации» Московского технического университета связи и информатики (МТУСИ) при подготовке магистров по направлению подготовки 15.04.04 «Автоматизация технологических процессов и производств», программа «Системы искусственного интеллекта промышленного интернета вещей», а именно:

- на основе полученных результатов модифицирована рабочая программа учебной дисциплины «Программная и аппаратная производственная интеграция. Тестирование» для магистрантов по направлению 15.04.04 «Автоматизация технологических процессов и производств», программа «Системы искусственного

интеллекта промышленного интернета вещей». В нее введен раздел «Применение компьютерного зрения для промышленных манипуляторов», включающий методические подходы и рекомендации по интеграции систем стереозрения и методов определения координат объектов захвата для промышленных манипуляторов.

- система дистанционного управления промышленным манипулятором Kuka, предложенная в диссертации, применяется при разработке интеллектуальных программно-аппаратных комплексов магистрантами направления 15.04.04 «Автоматизация технологических процессов и производств» при написании магистерских диссертаций с использованием оборудования центра робототехники МТУСИ, в частности двух промышленных манипуляторов Kuka.

2. Полученные в диссертационной работе результаты внедрены и используются в научно-производственной деятельности ООО «Геларм» (г. Москва), а именно

- подсистема дистанционного управления промышленным манипулятором Kuka, для удаленного взаимодействия, в режиме потоковой передачи данных, с манипулятором;

- алгоритм распознавания объектов при обработке видеопотока на удаленном устройстве, обеспечивающий расчет координат объекта в рабочей зоне манипулятора, с помощью компьютерного зрения.

**Соответствие паспорту научной специальности.** Область исследования и содержание диссертации соответствуют паспорту специальности 2.5.4 - «Роботы, мехатроника и робототехнические системы» в части:

– п. 5 «Методы, алгоритмы, программные и аппаратные средства управления роботами, робототехническими и мехатронными системами, включая адаптивное, оптимальное, распределенное, интеллектуальное и супервизорное управление»;

– п. 6 «Математическое и программное обеспечение, компьютерные методы и средства обработки информации в реальном времени в роботах, робототехнических и мехатронных системах».

**Степень достоверности и апробация результатов.** Достоверность результатов обеспечивается экспериментальным исследованием разработанных метода и алгоритмов сортировки ферромагнитных деталей промышленным манипулятором с использованием компьютерного зрения.

Основные результаты диссертационного исследования представлены на международных научно-технических конференциях:

1. СXXXII Студенческая международная научно-практическая конференция «Научное сообщество студентов XXI столетия. ОБЩЕСТВЕННЫЕ НАУКИ», Новосибирск, 2023;
2. Conference of Russian Young Researchers in Electrical and Electronic Engineering, St. Petersburg, Russia, 2023;
3. International Conference "Quality Management, Transport and Information Security, Information Technologies", Petrozavodsk, Russia, 2023;
4. International Conference "Quality Management, Transport and Information Security, Information Technologies", Nalchik, Russia, 2024;
5. Intelligent Technologies and Electronic Devices in Vehicle and Road Transport Complex (TIRVED), Moscow, Russia, 2024;
6. International Russian Smart Industry Conference (SmartIndustryCon), Sochi, Russia, 2024;
7. XV Молодёжный научный форум МТУСИ «Телекоммуникации и информационные технологии», Москва 2024;
8. Международная научно-техническая конференция "Автоматизация", Сочи, Россия, 2025.

**Публикации.** По результатам работы опубликовано 8 статей, в том числе входящих в перечень ВАК – 2, в научных изданиях, индексируемых в международной базе Scopus – 4, в научных изданиях, индексируемых в РИНЦ - 2 работы. Свидетельства о государственной регистрации программы для ЭВМ – 3. Свидетельство о регистрации полезной модели – 1.

**Личный вклад.** В опубликованных научных трудах диссертантом внесен основной вклад, касающийся разработки методов, алгоритмов и их реализации.

Вклад соавторов ограничивался постановкой задач на исследования и обсуждением результатов.

**Структура и объем работы.** Диссертационная работа состоит из введения, четырех глав, заключения, списка используемой литературы и приложений. Общий объем диссертации: 160 страниц. Диссертационная работа содержит 8 таблиц, 60 рисунков и 3 приложения. Список литературы включает 127 источников.

## **ГЛАВА 1. ИССЛЕДОВАНИЕ МЕТОДОВ УПРАВЛЕНИЯ ПРОМЫШЛЕННЫМ МАНИПУЛЯТОРОМ С ПРИМЕНЕНИЕМ ТЕХНОЛОГИИ КОМПЬЮТЕРНОГО ЗРЕНИЯ**

В главе проведен анализ современных методов управления промышленными манипуляторами, включая технологии компьютерного зрения. Рассмотрены основные подходы к управлению промышленными манипуляторами, их преимущества и ограничения. Проанализированы методы определения пространственного положения объектов.

### **1.1. Адаптивное и интеллектуальное управление роботами**

Цифровая трансформация промышленности является определяющим направлением развития мировой экономики. Согласно ГОСТ Р 70988-2023 [1] цифровая трансформация промышленности - это приоритетное направление развития отечественной экономики, связанное со стратегической трансформацией бизнеса и реализацией управленческих и производственных процессов посредством сквозных цифровых технологий, позволяющее создавать умное производство и обеспечивать новые формы цифрового взаимодействия предприятий промышленности и смежных отраслей, включая создание цепей добавленной стоимости.

Мировой рынок цифровой трансформации достиг в 2021 г. объема в 521,5 млрд долл. и, как ожидается, достигнет 1247,5 млрд долл. к 2026 г. при совокупном годовом темпе роста 19,1 % в течение 2021–2026 гг. Благодаря исследовательским данным, в 2021–2026 гг. ожидается увеличение глобального рынка технологий «Индустрии 4.0» с 64,9 до 165,5 млрд долл. при среднегодовом темпе роста в 20,6 %. В свою очередь, такой сегмент, как рынок «умного производства» вырастет за указанный период с 214,7 до 384,8 млрд долл. при среднегодовом темпе роста 12,4 % [2]. Российский рынок цифровой трансформации можно оценить примерно в 6 трлн руб. (2021 г.). он включает три сегмента: ПО и ИТ-услуги, аппаратное обеспечение (ИКТ-оборудование, микроэлектронику, потребительскую

электронику) и телеком-услуги (доступ к интернету, мобильную и фиксированную связь) [3].

Автоматы и роботы берут на себя значительную часть задач на предприятиях, а нередко и полностью заменяют людей. По данным Международной федерации робототехники [4], в 2014–2019 годах число промышленных роботов на заводах увеличилось на 85%. В 2020-м на предприятиях трудились 2,7 млн таких машин.

Они помогают компаниям нарастить объемы выпуска продукции, обеспечить сверхточное производство и одновременно сокращают потребность в дефицитных кадрах. Поэтому их число на предприятиях продолжит расти [5].

Как отмечается в исследованиях [6,7], ключевым ограничением при внедрении промышленных роботов остается их недостаточная гибкость, что особенно проявляется в условиях неструктурированных производственных сред. Данная проблема имеет два принципиальных аспекта:

— Производители роботов чаще всего предоставляют не только собственное оборудование, но и собственные программные решения [8]. В таком случае для координации между различными устройствами потребуются многочисленные интерфейсы прикладного программирования, и может даже потребоваться специальное программное обеспечение.

— Современные промышленных манипуляторы долго и сложно перепрофилировать на выполнение других задач. В настоящее время чаще всего используются системы с программным управлением роботами, реализующие простейший способ управления, когда априорно достаточно точно известно все о внешней среде и самом роботе, что позволяет заранее спланировать и запрограммировать работу робота. Иными словами, такое управление предполагает полностью детерминированные внешние и внутренние условия.

В настоящее время роботы все активнее применяются для решения задач, в которых невозможно заранее определить внешние и внутренние условия:

- Динамический захват объектов. Для операций с произвольно расположенными или движущимися объектами, по данным [9], необходимы системы технического зрения с большим количеством кадров в секунду, которые

зависят от скорости перемещения объектов, а также автоматическое наведение захвата манипулятора на объект.

- Навинчивание гайки на болт — для успешного выполнения этой задачи нужно сперва "наживить" гайку на резьбу [10].
- Вставка стержня в отверстие с высокой точностью, превышающей пределы позиционирования при программном управлении, что требует выполнения пробных поисковых движений [11].
- Дуговая сварка швов, положение которых может изменяться — важно вести сварочную головку вдоль шва и с заданной угловой ориентацией [12].
- Передвижение в неизвестной заранее среде с обходом или преодолением неожиданных препятствий [13] и так далее.

Для проведения таких операций важно оперативно получать данные о среде в процессе их выполнения, чтобы использовать эту информацию в режиме реального времени для управления движением. Это управление, основанное на актуальных данных о внешней среде, называется адаптивным. Ключевым условием его реализации является наличие сенсорных систем, которые предоставляют эту информацию, то есть «ощущение» робота [14]. Адаптация увеличивает функциональные возможности робота, позволяя ему решать задачи, которые нельзя решить с помощью программы, и упрощает его использование, не требуя точных знаний о модели внешней среды и самом роботе.

Адаптивное управление применяется в следующих вариантах:

— определение значений заранее неизвестных параметров для ввода их в управляющую программу системы. Здесь адаптивное управление является надстройкой над системой программного управления, осуществляя ее самонастройку, — это параметрическая адаптация системы программного управления;

— выбор управляющей программы из готового набора программ или сборка ее из набора типовых подпрограмм, в том числе и в ходе выполнения

операции, на основе оценки текущей ситуации по сенсорной информации. Здесь адаптивное управление тоже дополняет систему программного управления;

— изменение (переключение) структуры системы управления, т. е. ее алгоритма, на основе оценки текущей обстановки. Например, изменение способа коррекции в системе программного управления или переход от одной сенсорной системы к другой в системе адаптивного управления.

Последние два варианта — это структурная адаптация системы программного управления (самоорганизация системы);

— беспрограммное управление движением в реальном времени только по сенсорной информации. Датчики (энкодеры, тензометры) нужны, чтобы гарантировать, что реальное движение привода точно соответствует запланированному, несмотря на помехи (нагрузку, трение). Например, выполнение операции наведения рабочего органа манипулятора на произвольно расположенный объект [15].

Таким образом, адаптивное управление может быть реализовано как с помощью системы программного управления, так и путем прямого воздействия на приводы. В процессе выполнения определенных технологических операций адаптивное управление чаще всего используется именно в сочетании с программным управлением, что позволяет изменять параметры или структуру программного управления, а в некоторых случаях полностью заменять его, когда требуется переход к управлению на основе актуальной сенсорной информации.

Принцип адаптивного управления может быть применен не только для координации работы приводов манипулятора или движения робота, но и в системе управления отдельным приводом. В этом случае системы управления робота в целом и его исполнительных механизмов остаются программными, поскольку адаптация относится не к общей системе, а к конкретной локальной задаче [16].

В случае адаптивного управления алгоритм операции захвата объекта, который находится в случайном положении или движется может выглядеть, как на рис. 1.1.

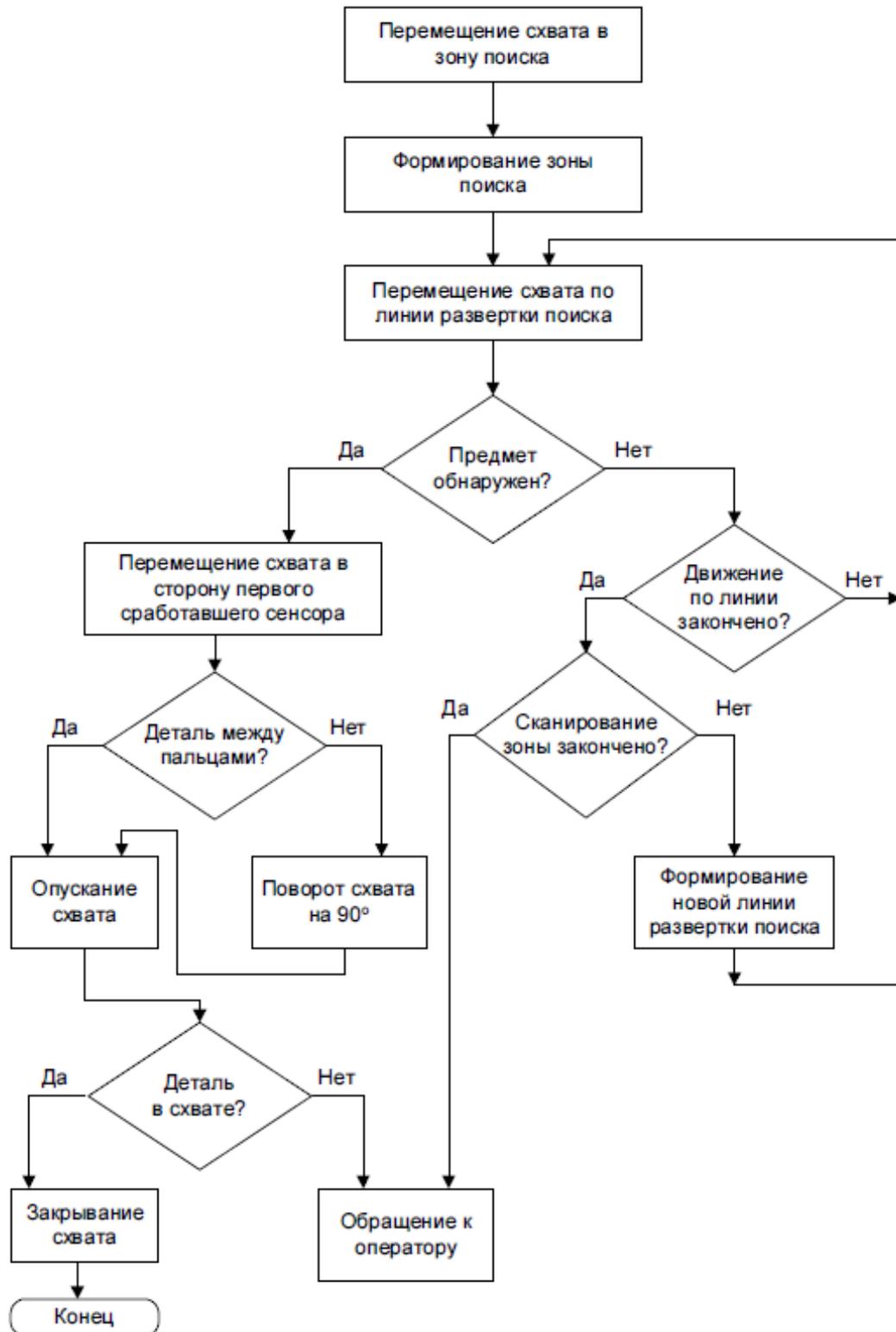


Рисунок 1.1 - Алгоритм операции захвата объекта, который находится в случайном положении или движется [17]

При этом схват должен быть оснащен сенсорами в виде дальномеров, расположенных на его пальцах, и датчиком наличия предмета между пальцами

[18]. После вывода схвата по программе в зону предполагаемого нахождения предмета производится сканирование схватом рабочей плоскости (например, строчное сканирование в квадрате с центром в точке вывода). Схват при этом ориентируется торцами пальцев перпендикулярно рабочей плоскости. При обнаружении сенсорами предмета схват останавливается над ним и ориентируется так, чтобы его пальцы находились симметрично относительно середины предмета с двух сторон от его наиболее узкой части. После этого схват опускается, и по сигналу датчика наличия предмета между пальцами пальцы схвата сводятся, захватывая предмет. В случае неудачи манипулятор останавливается и сигнал об этом поступает на пульт оператора. Однако обычно до этого производится повторение операции.

Проблемой данного метода является:

1. Схват должен быть оснащен датчиками в виде дальномеров и датчиками наличия предмета между пальцами
2. Затрачивается время на сканирование схватом рабочей плоскости
3. Нет возможности определения всех пространственных координат объекта

Поэтому в настоящее время для решения подобных задач используются системы интеллектуального управления.

Современные робототехнические комплексы активно интегрируют методы искусственного интеллекта для реализации следующих ключевых функций.

- Интеллектуальная обработка данных сенсоров. Алгоритмы машинного обучения обеспечивают интерпретацию сигналов от разнородных датчиков, включая визуальные, тактильные и инерциальные системы [19].
- Когнитивные функции принятия решений. Интеллектуальные системы анализа ситуации и стратегического планирования обеспечивают адаптивное поведение робота в динамической среде [20].
- Интерактивные человеко-машинные системы. Естественные интерфейсы взаимодействия на основе ИИ обеспечивают эффективную коммуникацию между оператором и робототехническим комплексом [21].

Основные области применения интеллектуальных роботов – это именно выполнения таких сложных операций, которые требуют интеллектуального подхода при решении вышеперечисленных задач.

Согласно данным Международной федерации робототехники [4] за 2023 год 68% промышленных роботов с интеллектуальными функциями оснащены системами машинного зрения. После сбора данных, первым шагом в рабочем процессе таких систем является обнаружение объектов или присутствия объектов или людей. Для некоторых сценариев, таких как мониторинг зон безопасности при сотрудничестве человека и робота [22], достаточно обнаружить только присутствие объекта или человека, чтобы продолжить, приостановить или остановить выполнение задачи. В большинстве сценариев этого недостаточно, и всю необходимую информацию об обнаруженных объектах необходимо извлечь и учесть на этапе планирования. В этих сценариях типичными действиями после обнаружения объекта являются оценка положения объекта, определение положение захвата, планирование движения и управление в соответствии с этим [23].

При интеллектуальном управлении промышленным манипулятором выделяется ряд задач, которые отличаются по сложности [24]:

1. работа с неперекрывающимися объектами;
2. работа с разнородными объектами, перекрывающимися друг друга и находящимися в заранее неопределенной позиции.

Первую задачу можно считать несложной с точки зрения компьютерного зрения, поскольку рабочая область манипулятора включает объекты, которые не заслонены другими. Подобная конфигурация встречается часто, но большинство применимых случаев связаны с оценкой положения детали и более точным извлечением признаков для дальнейшей обработки, такой как генерация пути при автономной окраске [25], сварке [26], полировке и т. д.

Вторая задача наиболее актуальна для современных промышленных роботизированных систем. Она предполагает работу с множеством объектов, которые могут частично перекрывать друг друга или иметь сложную форму, что

требует специализированных алгоритмов захвата. Типичным примером такой задачи является операция извлечения деталей из контейнера [27] – классическая проблема промышленной робототехники, не теряющая актуальности на протяжении десятилетий. Суть задачи заключается в том, что роботу необходимо последовательно извлекать случайно расположенные и накладывающиеся друг на друга объекты, точно позиционируя их в заданном месте. Эта операция считается одной из наиболее сложных в автоматизации производства из-за огромного разнообразия возможных пространственных конфигураций объектов.

В отдельных случаях проблему можно упростить, организовав упорядоченное расположение деталей вдоль вертикальной оси. Такое структурированное размещение позволяет использовать более простые алгоритмы компьютерного зрения [28]. Однако на практике это решение часто оказывается неприменимым – оно требует существенной перестройки производственных линий с установкой громоздких разделительных конструкций [29]. Подобные изменения не только занимают производственное пространство, но и снижают гибкость предприятия, затрудняя адаптацию к выпуску новой продукции. В результате многие производства вынуждены сохранять ручной труд для подобных операций.

## **1.2. Анализ систем управления промышленными роботами**

Промышленные роботы – это самый распространенный класс в робототехнике, основную часть которого составляют манипуляционные роботы. Данный подкласс характеризуется наличием манипулятора с конечным эффектором (рабочим органом), что позволяет им выполнять ключевые для автоматизации производства функции: захват, перемещение, позиционирование и обработку деталей и инструментов. Именно манипуляционные роботы являются основой гибких производственных ячеек, сборочных линий и комплексов для выполнения таких операций, как сварка, окраска, паллетизация и монтаж [30]. Системы управления таких роботов, как правило, имеют схожую структуру, которая в обобщенном виде представлена на рис. 1.2 и предполагает ручное программирование манипулятора.

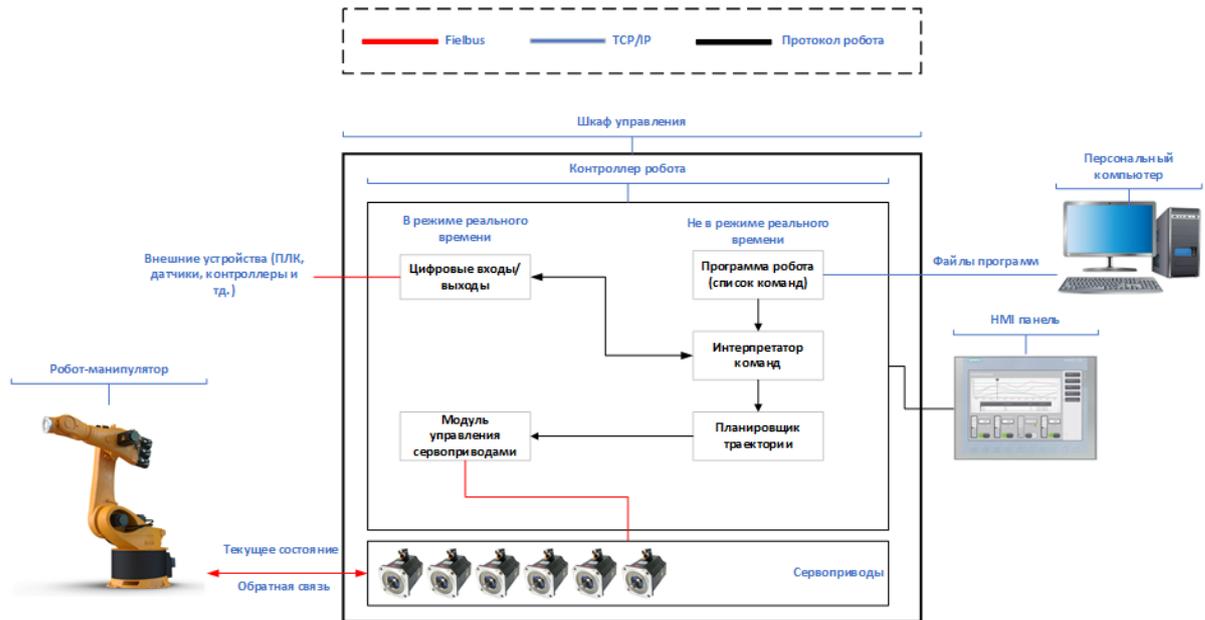


Рисунок 1.2 - Структурная схема системы управления манипуляционного промышленного робота [31]

Промышленные сервоприводы управляют разными степенями подвижности через исполнительные регуляторы, которые подключаются к устройству управления. Это устройство может быть построено на базе программируемого логического контроллера или обычного персонального компьютера. Его главная задача заключается в формировании оптимальных параметров скорости и положения для каждого сервопривода, учитывая кинематику и динамику робота. Для связи между устройством управления, сервоприводами используется одна из промышленных шин, на основе протоколов FieldBus, TCP/IP, работающая в реальном времени.

Для выполнения движения рабочего органа робота и управления его дополнительным оборудованием используются команды, которые поступают через внешние сетевые интерфейсы, а также специальные программы на языках программирования, таких как ARPS, RAPID, KRL и др., или в виде G-кодов. В большинстве случаев на таком устройстве используется операционная система реального времени, такая как VxWorks [32], Linux с патчем реального времени [33] или операционные системы на базе Windows.

При использовании такого подхода система управления промышленного робота выполняет последовательность операций и движений, которые должны быть выполнены в рамках запрограммированного рабочего цикла, определяемого пользователем с помощью набора инструкций, закодированных на языке программирования, специфичном для конкретного поставщика, и непосредственно загружен на систему управления робота. При планировании пути рассчитывается возможный путь, который соединяет начальную точку с конечной точкой в пространстве фланца робота-манипулятора, тогда как генератор траектории объединяет геометрическую и временную информацию и предоставляет в качестве результата обратной кинематической задачи движение суставов робота в пространстве. Затем вычисленные заданные значения положения циклически передаются на сервоприводы, которые модулируют и подают электрический ток на серводвигатели посредством управления с обратной связью по положению серводвигателей, т. е. углового энкодера, установленного на каждом серводвигателе.

Программирование логики управления и задач робота с помощью онлайн программирования — старейшая и наиболее распространенная практика в промышленности. Такой подход приводят к следующим практическим проблемам:

- Потеря времени. Любое запрошенное изменение в роботизированной задаче/последовательности/цикле требует прерывания работы установки, ручной модификации программы робота и ее последующей загрузки в контроллер робота. Затем, прежде чем вернуть робота в полную работоспособность, контроллеру требуется дополнительное время для повторной инициализации системы и выполнения первого движения на очень низкой скорости [34-36], что неизбежно приводит к задержкам в производственной системе.

- Дублирование кода [37]. Современная производственная линия может включать сотни роботов, каждый из которых имеет собственный код, исполняемый на локальном уровне. В большинстве случаев аналогичные задачи (т.е. процедуры в рабочем цикле робота) программируются несколько раз в разных сценариях и загружаются на каждый контроллер робота, что неизбежно увеличивает

количество программ робота, нуждающихся в обслуживании. Дублирование кода может сказываться на количестве сбоев и увеличивает время, необходимое для любой реконфигурации, поскольку операторы, отвечающие за обновление кода на многих автономных контроллерах, не обмениваются информацией.

- Ограничения языков программирования. В отличие от программирования станков с ЧПУ, основным язык в которых G-код, в роботах манипуляторах отсутствует универсальный, независимый от производителя язык программирования, что усложняет интеграцию различных робототехнических технологий на одном производственном предприятии. Этот подход затрудняет поддержку, обновление или добавление новых функций в зависимости от текущих производственных потребностей [38], а также усложняет взаимодействие между роботами разных производителей или другими внешними устройствами, такими как камеры технического зрения, различные датчики и исполнительные механизмы, в рамках одного производственного процесса. В результате сложно настроить взаимодействие и обмен данными между различными робототехническими системами, что является препятствием для совместной работы [39]. Обобщенные инструменты автономного программирования (например, RoboDK, Siemens Process Simulate) используют специальные постпроцессоры для перевода команд и функций 3D-моделирования в готовые к использованию коды роботов, содержащие инструкции конкретного поставщика. Однако они по-прежнему не охватывают все библиотеки функций каждого коммерческого языка программирования робота и не могут заменить оператора, когда необходимы более сложные процедуры программирования. Также разработка стандартизированного языка программирования не поддерживается поставщиками роботов и они не предоставляют подробную информацию о своих схемах управления низкого уровня [40].

- Контроллеры, не способные выполнять сложные вычислительные операции. Аппаратная основа современных промышленных контроллеров часто не соответствует требованиям сложных алгоритмов, поскольку может включать

низкопроизводительные процессоры (например, одноядерный Intel Pentium 4) [31] и использовать устаревшие операционные системы (например, Windows XP). Дополнительным ограничением является крайне малый объём доступной памяти (в некоторых случаях не более 25 МБ), что делает невозможным развёртывание современных программных комплексов и требует постоянного контроля за размером всех загружаемых компонентов.

Для расширения функциональности и быстрой интеграции в уже существующие контроллеры робота, в настоящее время применяются новые подходы, такие как.

1. Задание команд управления с внешнего устройства [41]. Команды управления роботу отправляются с внешнего устройства (ПК или ПЛК), в формате программы, написанной на высокоуровневом языке программирования. На контроллере робота программное обеспечение-интерпретатор запускается циклически для преобразования полученных команд на язык, специфичный для робота. Затем преобразованные команды обрабатываются и на контроллере робота осуществляется планирование траектории, как в традиционном подходе. Структурная схема системы управления, при задании команд управления с внешнего устройства приведена на рис.1.3.

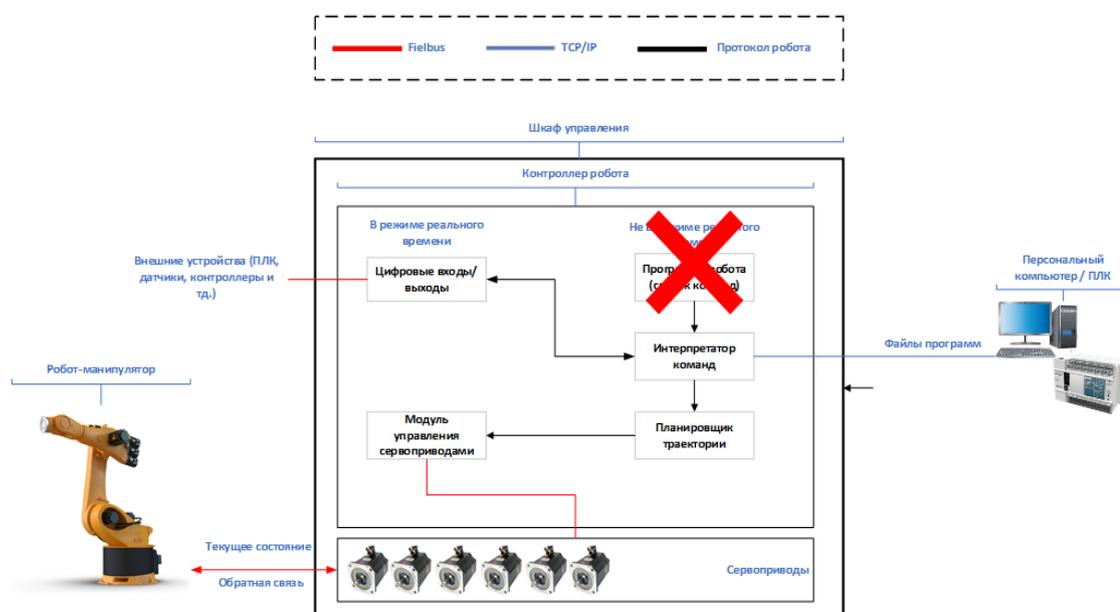


Рисунок 1.3 - Структурная схема системы управления, при задании команд управления с внешнего устройства [31]

2. Внешний планировщик траектории [42]. Планирование траектории выполняется с помощью внешнего устройства, которым может быть либо ПК, оснащенный платой реального времени, либо ПЛК. Быстрое вычисление траектории движения и обратная связь с роботом осуществляется через модуль ввода-вывода робота. Модуль контроллера, которые не работает в режиме реального времени, не участвует в планировании движения, хотя его все еще можно использовать для обработки других типов задач и логических событий. Структурная схема системы управления, при использовании планировщика траектории приведена на рис.1.4.

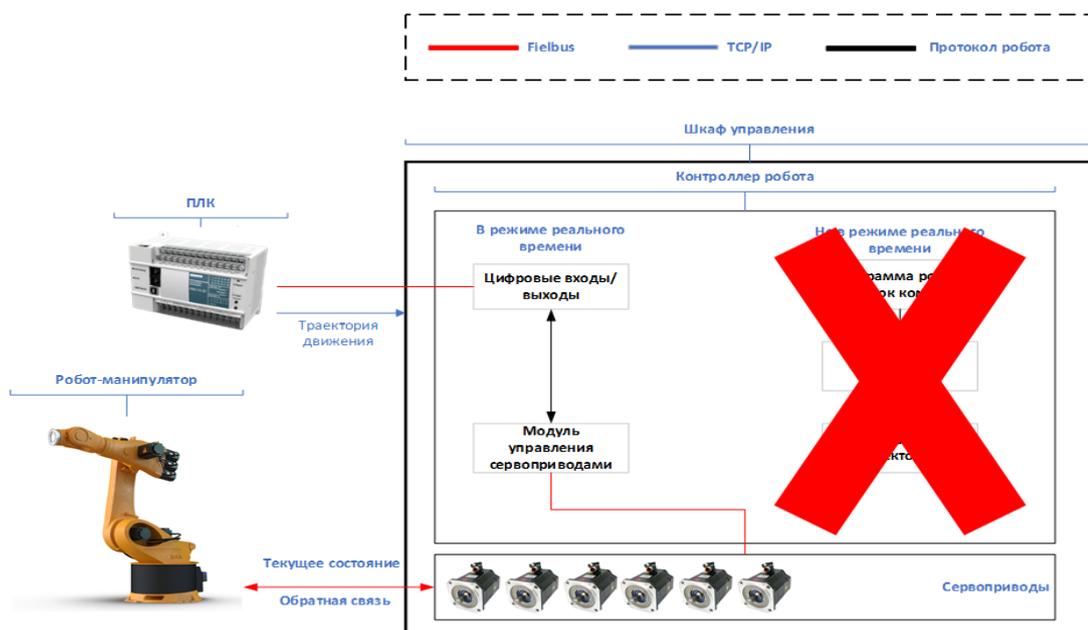


Рисунок 1.4 - Структурная схема системы управления, при использовании планировщика траектории [31]

В случае задания команд управления с внешнего устройства оператор программирует робота в среде более высокого уровня (например, на ПЛК или языке программирования Python), а разработанные инструкции последовательно передаются в контроллер робота, где реализуется интерпретатор, написанный на его родном языке [40]. Для этого используется ПК поскольку для передачи инструкций и чтения/записи переменных, структур и регистров связь с роботом в реальном времени не требуется. В настоящее время разработаны пакеты для передачи данных и команд между роботами и внешними устройствами, которые

можно разделить на коммерческие пакеты (уже выпущенные на рынок) и пакеты, ориентированные на исследования (все еще находящиеся в стадии разработки/доработки).

Одним из примеров коммерческого пакета является ПК Fanuc (PCDK) [43], который обеспечивает передачу информации и инструкций между внешним ПК и контроллером робота Fanuc (R-J3 и R-J3iB). Основные действия, которые позволяет реализовать пакет PCDK — это чтение/запись переменных, установка и настройка входов и выходов, чтение/запись позиций и загрузка/сохранение программ. Похожий пакет есть у компании ABB [44], который представляет собой специализированное приложение запускающейся на внешнем ПК и связывающееся с контроллером робота по сети.

Пакеты, которые ориентированы на исследования, строятся по аналогичной схеме на базе ПК, с возможностью использования большего количества высокоуровневых языков программирования. Например в [45] представлен набор инструментов MATLAB для программирования KUKA и Kawasaki IR. Обмен данными между программой MATLAB и интерпретатором на контроллере робота осуществляется через каналы передачи данных (т. е. последовательный порт или TCP/IP). Еще один инструмент с открытым исходным кодом JOpenShowVar, представленный в [46], позволяет читать и записывать все собственные переменные в контроллере KUKA без необходимости использования дополнительных проприетарных пакетов. Система имеет архитектуру клиент/сервер: JOpenShowVar выступает в качестве клиента на удаленном ПК, а пакет KUKAVARPROXY — в качестве сервера в контроллере робота. Связь между клиентом и сервером осуществляется через беспроводной канал передачи данных, поэтому доступ к данным робота в режиме реального времени не гарантируется и нет возможности отправки нескольких последовательных точек для движения робота.

В таблице 1.1 автором представлены основные пакеты для задания команд управления с внешнего устройства.

Таблица 1.1 – Пакеты для задания команд управления с внешнего устройства

Название пакета	Устройство ввода	Совместимость с контроллерами робота	Свободно распространяемое ПО	Плюсы и минусы
N.D. [47]	ПК	KUKA/Comau/Adept	ПО не распространяется	Ограниченное количество команд управления
RAP [48]	ПК (Robotstudio)	ABB	Платное ПО	Работает только с роботами компании ABB
ABB PC SDK [44]	ПК	ABB	Свободно распространяемое	Возможность редактирования кода, нет возможности потоковой передачи команд, работает только с роботами компании ABB
Fanuc PCDK [43]	ПК	Fanuc	Свободно распространяемое	Возможность редактирования кода, потоковая передача команд (например, запуск программы), работает только с роботами компании Fanuc, язык программирования Visual Basic
JOpenShow Var [46]	ПК	KUKA	Свободно распространяемое	Возможность чтения и записи команд, нет возможности потоковой передачи данных, работает только с роботами компании Kuka
KCT [49]	ПК (Matlab)	KUKA (RSI)	Свободно распространяемое	Возможность планирования/управления движением, нет возможности потоковой передачи данных, работает только с роботами компании Kuka, для работы необходимо платное ПО Matlab

При использовании коммерческих продуктов основной акцент делается на возможность полностью контролировать робота с высоким уровнем надежности и безопасности, а пакеты направленные на исследовательские цели часто ограничены в своем функционале и не позволяют использовать все команды

управления роботом. Практически все ПО связано с конкретным производителем и не позволяет использовать его для роботов других компаний.

В робототехнике могут возникать ситуации, когда требуется точное управление движением, например, для сварки, когда требуется введение/контроль определенных геометрических траекторий и/или законов движения фланца робота. В таких случаях задание команд с внешнего устройства не подходит, т.к. манипулятор использует внутренний модуль планирования траектории движения робота, который ограничен по количеству вариантов. Например, в роботах компании Kuka [34] поддерживаются следующие типы движения PTP (точка-точка), LIN (линейное движение), CIRC (круговое движение), Spline (подходит для особо сложных криволинейных траекторий). Для решения данной проблемы часто используется внешний генератор траектории, но для его использования контроллер робота должен быть оснащен интерфейсами реального времени, позволяющими управлять движением робота с помощью аналоговых и цифровых выходов контроллера. На практике эти интерфейсы должны обеспечивать быструю и надежную циклическую связь с внешним устройством (например ПЛК), отвечающим за расчет траектории.

В таком случае необходимо знать данные о положении робота, которые вычисляются с помощью дополнительных датчиков с обратной связью, например датчик положения, энкодер, силомоментные датчики, которые циклически передают данные в контроллер робота с определенной скоростью. При использовании такого подхода робот движется без использования контроллера робота, что может привести к ошибочному определению сигнала управления, в результате некорректной работы оборудования, например неисправности датчика. Для уменьшения риска происшествия следует провести соответствующие проверки безопасности и следовать процедурам безопасности.

Для применения подхода к управлению промышленными манипуляторами, использующего внутренний планировщик траекторий, существуют дополнительные пакеты, которые позволяют управлять роботом в режиме реального времени, например KUKA.RobotSensorInterface (RSI) [34], ABB

Externally-guided motion (EGM) [35] и FANUC Dynamic Path Modifier (DPM) [36]. С помощью RSI обмен данными может осуществляться либо через Ethernet (протокол UDP/IP), либо через систему ввода-вывода контроллера робота (полевая шина). Система RSI поддерживает две скорости передачи 12 и 4 мс. Информация о положении должна отправляться в модуль RSI в течение каждого цикла, а полученные сигналы влияют на движение робота. ABB EGM предлагает те же функции, но разделен на два отдельных режима работы:

- управление положением EGM;
- коррекция пути EGM.

Первый способ перемещения робота основан на информации о его местоположении, предоставляемой внешним источником и передаваемой каждые 4 мс с некоторой задержкой управления до 20 мс. Второй метод позволяет корректировать траекторию робота в реальном времени с использованием информации от датчиков. Каждые 48 мс необходимо вносить коррекции в систему координат робота. Fanuc предлагает два режима работы DPM: модальный и встроенный. Модальный DPM подходит для постоянной модификации по заданному пути с частотой цикла 8 мс и уменьшенной задержкой управления. Встроенный DPM корректирует каждый сегмент движения только в конечном пункте назначения. Все описанные пакеты являются платными и используются в исследовательских целях для генерации траектории движения робота.

Данный способ имеет преимущества, по сравнению с планировщиками траекторий, которые используются в контроллерах роботов, т.к. он учитывает потребности каждой задачи и позволяет использовать различные датчики. Оператор при использовании такого планировщика траектории несет ответственность за управление всеми параметрами, такими как скорость, ускорение и положение робота. Помимо этого, существуют проблемы, связанные с ограниченным набором функции ввода-вывода и большими экономическими затратами на приобретение пакетов.

Для решения задач классификации и адаптивного захвата произвольно расположенных объектов необходима модульная архитектура управления. В такой

архитектуре функции генерации управляющих команд возлагаются на внешний вычислительный узел (например, ПК), что обусловлено требованием к использованию высокоуровневых языков программирования и вычислительным ресурсам для обработки видеопотока и работы алгоритмов компьютерного зрения.

### **1.3. Анализ исследований в области распознавания объектов в промышленной робототехнике**

В современных промышленных системах автоматизации технологии распознавания объектов играют ключевую роль, особенно в задачах сортировки и упаковки продукции. Выбор оптимального метода и технических средств зависит от специфики решаемой задачи, характеристик объектов и сложности их описания. Основу составляют системы технического зрения, включающие RGB-D камеры, стереоскопические датчики и лидары, которые обеспечивают получение как двумерных изображений, так и трехмерных данных о рабочей сцене. Эти устройства позволяют точно определять положение, ориентацию и геометрические параметры объектов даже в условиях динамично изменяющейся производственной среды. Для обработки визуальной информации применяются современные методы машинного обучения, в частности глубокие нейронные сети. Сверточные архитектуры и трансформеры демонстрируют высокую эффективность при детекции и классификации объектов различной формы и текстуры. Особое внимание уделяется алгоритмам, способным работать в реальном времени и сохранять точность при частичном перекрытии объектов или изменении освещенности.

В исследовании [50] для расширения функций коллаборативного робота Franka Emika Panda используется камера Realsense D435i, установленная на исполнительном органе. Вместо использования метода машинного обучения на основе большого набора данных авторы предложили метод ручного программирования робота на основе анализа единого эталонного изображения. Эта роботизированная система может обнаруживать различные объекты, независимо от их положения и ориентации, достигая среднего уровня успеха более 90% менее

чем за 5 минут обучения, используя Ubuntu 16.04, работающий на процессоре Intel(R) Core(TM). ЦП i5-2400 (3,10 ГГц) и графический процессор NVIDIA Titan X.

Другой подход к захвату случайно расположенных объектов представлен в [27], в котором авторы предложили набор метрик производительности, которые позволили сравнить четыре роботизированных системы для сбора мусора в соревнованиях Amazon Robotics Challenge 2017. Результаты оценки показали, что наиболее перспективными, для решения такой задачи, являются RGB-D камеры и сверточные нейронные сети.

Подобные задачи по локализации и сортировке объектов возникают практически в каждом производственном подразделении. В [51] предложен экспериментальный метод с использованием пневматического манипулятора для выделения объектов из набора по их цвету. Объект движется по конвейеру, и когда датчик цвета определяет указанный цвет, конвейер останавливается, а пневматический робот захватывает объект и помещает его на следующий конвейер. Если датчик цвета не обнаруживает, робот остается в режиме ожидания до конца операции. Описанная система сортировки работает по алгоритму обработки изображений в программе MATLAB.

Более совершенные методы распознавания объектов, основанные на одновременном определении цвета и высоты, представлены в статье [52], в которой робот-сортировщик на основе компьютерного зрения, одновременно обнаруживает и сортирует объекты по их цветам и высоте. Программно-аппаратный комплекс оснащен роботизированной рукой с 6 степенями свободы, с помощью которой он поднимает и затем размещает объекты в соответствии с их цветом и высотой в заранее определенное место в соответствии с требованиями производственной системы. Камера с программным обеспечением компьютерного зрения распознает различные цвета и высоты, а для сортировки объектов использовался алгоритм на основании каскадов Хаара. В предлагаемой системе эффективность сортировки по цвету и высоте составляет около 90%, что доказывает эффективность системы.

Еще один пример распознавания и определения дефектов на объекте представлено в [53], в котором три различные модели нейронных сетей (CNN, Fast R-CNN и Faster R-CNN) сравниваются при сортировке объектов простой формы и определение дефектов на них. В результате максимальная точность распознавания дефектов достигается с помощью нейронной сети Faster RCNN и равняется 78,8%.

Несмотря на определенный прогресс в области робототехники, задача надежного захвата произвольно расположенных объектов в сложных сценах остается ключевой проблемой. В исследовании [54] авторы предложили, основанный на использовании RGB-D камеры метод обнаружения твердых мусорных отходов. Система сортировки отходов состоит из сервера, технического зрения, промышленных роботов и конвейера. Эксперименты, проведенные по анализу изображений твердых отходов, показали среднее значение точности 49,1%, что является недопустимым при использовании подобных систем на производстве.

Другой сложной проблемой распознавания объектов и манипулирования ими являются объекты неопределенной формы, загрязненные пылью или более мелкими частицами, такими как минералы или уголь. Довольно часто такая задача требует не только распознавания объекта, но и определения положения центра масс объекта. В работе [55] предложили метод сортировки угля и пустой породы на основе обработки изображений. Особенности оттенков серого и текстуры образцов анализируются при освещении 300 люкс. Три классификатора обучаются с использованием функции оттенков серого, текстуры и функции соединения, сочетающей асимметрию с контрастом соответственно. Результаты экспериментов показывают, что классификатор, использующий совместный признак, превосходит другие. Кроме того, в статье авторы разрабатывают систему позиционирования и идентификации и дополнительно оценивают систему, используя изображения цели и пустой породы, которые случайным образом выбираются из производственной среды. Результаты показывают, что средние ошибки координат в направлениях  $x$  и  $y$  составляют 2,73% и 2,72%, точность идентификации проб угля и пустой породы составляет 88,3% и 90,0% соответственно, а сумма времени идентификации, позиционирования для одного отсчета в среднем составляет 0,130 с.

В робототехнике задача распознавания объектов часто включает в себя не только распознавание и определение координат, но и играет существенную роль при создании программы управления роботом. Целью статьи [56] является создание недорогой системы идентификации форм для программирования промышленных роботов на базе шестиосевого робота «ABB IRB 140» для процесса сварки в 2D. Вся система состояла из нескольких элементов, разрабатывавшихся на отдельных этапах. Авторами написана компьютерная программа, которая обеспечивает связь с цифровой камерой, захватывающую и обрабатывающую изображения. Кроме того, задачей программы являлось распознавание геометрических фигур, нарисованных человеком. В этом исследовании также представлены исследования метода бинаризации и распознавания контуров для этой задачи. Благодаря этому робот способен сваривать одни и те же контуры на 2D-плоскости.

В результате анализа можно сделать вывод, что современные промышленные РТК требуют многоуровневого подхода к распознаванию объектов, включающего аппаратные сенсоры, алгоритмы предобработки данных и интеллектуальные методы анализа. Оптимальное решение зависит от условий эксплуатации (освещенность, динамика сцены) и требуемой точности.

#### **1.4. Методы определения пространственных координат объекта**

При работе систем компьютерного зрения информацию об объекте, получаемую с камеры необходимо преобразовать в систему координат робота, для его движения. Помимо определения координат в 2D пространстве важным аспектом является значение глубины – расстояние до объекта по вертикальной оси.

Методы определения глубины объекта разделяются на три типа: метод определения глубины объекта с использованием одной камеры, метод определения глубины объекта с использованием геометрии бинокулярного зрения, метод определения глубины объекта с использованием RGB-D камеры.

*Метод определения глубины объекта с использованием одной камеры [57] работает следующим образом. Используются два изображения, которые делаются*

в двух положениях камеры при ее перемещении вдоль оптической оси. Предполагается, что расстояние между двумя положениями камеры известно. Расстояние от камеры до объекта вычисляется путем расчета соотношения размеров объекта, проецируемого на два изображения. Недостаток этого метода состоит в том, что для вычисления глубины объекта от камеры, необходимо отметить объекты определенного размера в изображении (например, с помощью QR-кода).

Устройства, состоящие из двух датчиков изображения (камер), разделенных известным расстоянием, называются стереосистемами. Принцип стереозрения основан на способности человеческого мозга оценивать глубину объектов, присутствующих на изображениях, снятых глазами [58].

При стереозрении две камеры расположены близко друг к другу на одной линии. Центры камер  $C_L$  и  $C_R$  разделены между собой базовой линией  $B$ . При этом обе камеры имеют одинаковое фокусное расстояние  $f$ . Некая точка  $P$  будет проецироваться в разные положение для камер  $p_L$  и  $p_R$  соответственно с координатами  $(x_L, y_L)$  для левой камеры и  $(x_R, y_R)$  для правой камеры. Эта разница в положении известна как диспаратность и описывается как диспаратность  $= x_L - x_R$  и используется для расчета расстояния по Оси  $Z$  (формула 1.1.) посредством математического соотношения [59], как показано на рис.1.5.

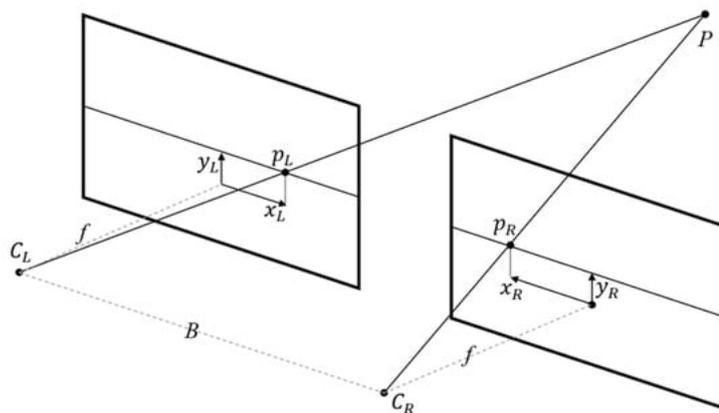


Рисунок 1.5 - Геометрическое соотношение конфигурации стереокамеры

$$Z = \frac{B * f}{x_L - x_R}, \quad (1.1)$$

Стереокамера передает изображение глубины, другими словами, изображения, значения интенсивности которых представляют глубину точки  $(x, y)$  на сцене в виде облака точек — это структура данных, используемая для представления точек в трех измерениях  $(X, Y, Z)$ , где глубина представлена координатой  $Z$  [60]. Как только изображения глубины станут доступны, можно получить облако точек, используя внутренние значения камеры, с помощью которой получена информация. Этот процесс известен как депроекция; точка  $P$  с координатами  $(X, Y, Z)$  может быть получена согласно (1.2-1.4) из информации о глубине  $D_{x,y}$ , представляющей собой значение карты глубины в точке с координатой  $(x_0, y_0)$  (в миллиметрах), где переменные  $C_x$  — координата центра изображения по оси  $X$  (в пикселях),  $C_y$  — координата центра изображения по оси  $Y$  (в пикселях),  $f_x, f_y$  — внутренние параметры оптической системы камеры, используемой для получения изображения (фокусные расстояния по осям  $X$  и  $Y$ ),  $(x_0, y_0)$  — координата центра обнаруженного объекта на изображении в пикселях [61].

$$X(mm) = \frac{D_{x,y}(C_x - x_0(\text{пиксели}))}{f_x}, \quad (1.2)$$

$$Y(mm) = \frac{D_{x,y}(C_y - y_0(\text{пиксели}))}{f_y}, \quad (1.3)$$

$$Z(mm) = D_{x,y}, \quad (1.4)$$

Метод определения глубины объекта с использованием RGB-D камеры основан на использовании особенностей конструкции RGB-D камеры, которая состоит из объектива IR-камеры и объектива RGB-камеры. Определение глубины объекта от камеры обеспечивается с использованием изображений глубины, получаемых с помощью объектива IR-камеры. Этот метод определения глубины способен достаточно точно вычислять глубину объекта без дополнительных вычислений. В настоящее время выделяется несколько применений RGB-D камер [62-66]. Информация о глубине, полученная с помощью RGB-D устройств, может быть использована для создания 3D-моделей, измерения расстояний,

распознавания объектов и многих других приложений, требующих пространственной информации.

Одной из важных характеристик датчиков RGB-D является разрешение глубины. Оно определяет наименьшее расстояние, на котором датчик способен измерить точную глубину. Более высокое разрешение глубины позволяет более точно измерять относительно близкие объекты.

Еще одна важная характеристика - это точность глубины или погрешность измерения. Она указывает, насколько близко измеренные значения глубины приближаются к реальным значениям. Чем ниже погрешность, тем точнее и достовернее будут измерения.

Кроме того, угол обзора датчика также важен при работе на близком расстоянии. Он определяет, насколько широкую область объектов датчик способен охватить и измерить. Большой угол обзора обеспечивает большую область видимости, что может быть полезным при сканировании больших объектов.

В рассмотренных основных методах определения глубины объекта для систем компьютерного зрения в робототехнике: монокулярный (с одной камерой), стереоскопический (бинокулярное зрение) и RGB-D-сенсоры ключевыми характеристиками для выбора метода являются разрешение и точность глубины, угол обзора и устойчивость к внешним условиям. RGB-D камеры демонстрируют преимущество в задачах, требующих быстрого и надежного получения 3D-данных, таких как навигация, манипуляция и 3D-реконструкция.

### **Выводы по главе**

1. В проанализированных работах выделено, что внедрение роботов сталкивается с проблемами негибкости, связанными с зависимостью от проприетарных решений и сложностью перепрофилирования. Адаптивное управление, основанное на применении технологий компьютерного зрения, расширяет функциональность роботов, позволяя им распознавать и проводить классификацию объектов, производить их локализацию в пространстве и работать в недетерминированных условиях.

2. Традиционные системы с ручным программированием обладают существенными недостатками, такими как временные затраты на перенастройку, дублирование кода и ограниченная совместимость. Альтернативные подходы, включая задание команд с внешних устройств и использование внешних планировщиков траектории, позволяют повысить гибкость и эффективность управления, но требуют специализированного ПО и аппаратного обеспечения.

3. Исследования, связанные с распознаванием объектов в промышленной робототехнике, показали, что основным направлением является использование нейросетевых методов для классификации и детектирования объектов и RGB-D сенсоров для определения глубины и построения 3D моделей объектов. При этом в рассмотренных исследованиях сохраняются проблемы с точностью распознавания в сложных сценах и невозможностью работы с перекрывающимися объектами.

4. Методы определения пространственных координат объектов имеют ряд ограничений. Традиционные методы с одной камерой требуют известного смещения камеры и маркировки объектов, что ограничивает его применение в динамических средах. Стереозрение, основанное на диспаратности, позволяет вычислять глубину через геометрические соотношения, но чувствительно к калибровке и освещению. Наиболее точным является использование RGB-D камер, которые напрямую предоставляют данные о глубине за счет комбинации RGB- и IR-сенсоров, обеспечивая высокую точность без сложных вычислений.

## ГЛАВА 2. РАЗРАБОТКА ПОДСИСТЕМЫ ДИСТАНЦИОННОГО УПРАВЛЕНИЯ ПРОМЫШЛЕННЫМ МАНИПУЛЯТОРОМ

В главе представлена разработка архитектуры подсистемы управления промышленным манипулятором с возможностью задания управляющих программ с внешнего вычислительного устройства. Архитектура подсистемы, включает: программный модуль на внешнем вычислительном устройстве с интерфейсом связи с контроллером робота и программный модуль на контроллере робота, содержащий программный пакет C3 Bridge Interface, позволяющий удаленным клиентам выполнять запросы к подсистеме KUKA Cross 3 и программа на языке KRL, обеспечивающая непосредственное управление движением манипулятора.

### 2.1. Архитектура подсистемы дистанционного управления промышленным манипулятором

Согласно опросу [67] наиболее распространёнными среди крупных промышленных предприятий в России являются промышленные роботы KUKA, которые занимают 44% рынка. Второе место занимают промышленные роботы Fanuc 23%, третье место с 12% рынка занимают роботы компании ABB. Многие крупные компании закупают устройства одновременно у нескольких поставщиков (18%). Статистика основных поставщиков представлена на рисунке 2.1.

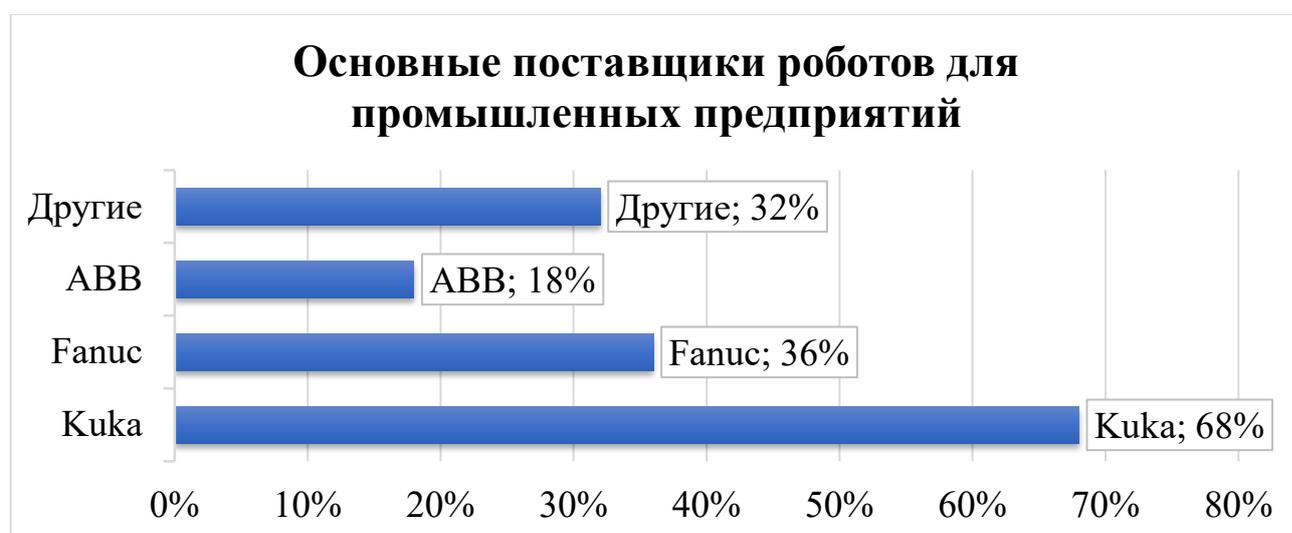


Рисунок 2.1 – Основные поставщики роботов для промышленных предприятий

Анализ, проведенный в первой главе, выявил, что современные системы управления промышленными роботами носят проприетарный характер и жестко привязаны к оборудованию конкретных производителей, что исключает их кросс-платформенное применение. В рамках диссертационной работы разрабатываемая подсистема дистанционного управления, будет реализована для манипуляторов компании KUKA, в связи с тем они являются основными поставщиками промышленных манипуляторов в РФ [67]. В качестве манипулятора используется робот KUKA KR4 R600 [68] и контроллер KUKA KR C5 micro [69]. Технические характеристики робота-манипулятора приведены в таблице 2.1.

Таблица 2.1 – Технические характеристики робота манипулятора KUKA KR 4 R600

Рабочий диапазон, мм	600	Грузоподъемность, кг	4
Количество осей	6	Повторяемость, мм	$\pm 0.015$
Угол поворота по осям, °		Максимальная скорость по осям, °/s	
J1	-1	J1	336
J2	-4,875	J2	336
J3	-0,766666667	J3	488
J4	-1	J4	600
J5	-1	J5	529
J6	-1	J6	800

Рабочая зона манипулятора KUKA KR 4 R600 представлена на рис 2.2.

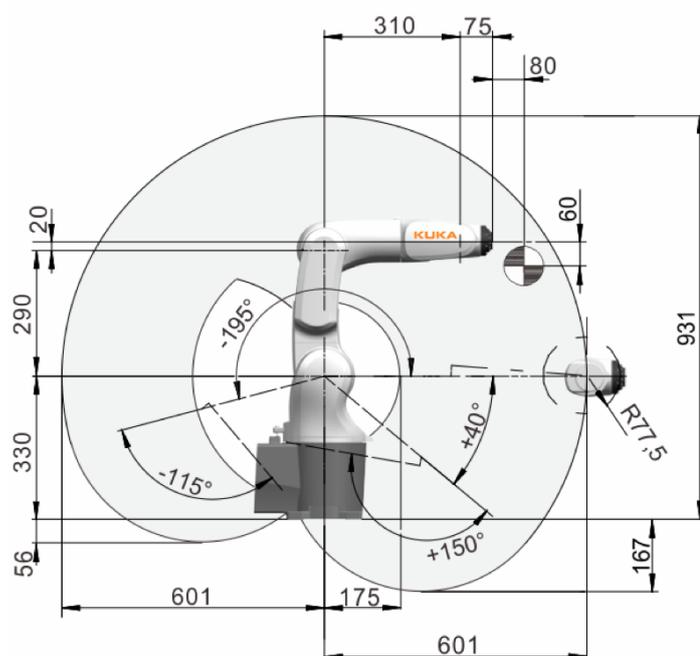


Рисунок 2.2 - Рабочая зона робота манипулятора Kuka KR4 R600

Ручное управление манипулятором Kuka осуществляется с помощью пульта управления KUKA SmartPAD, внешний вид которого представлен на рисунке 2.3.

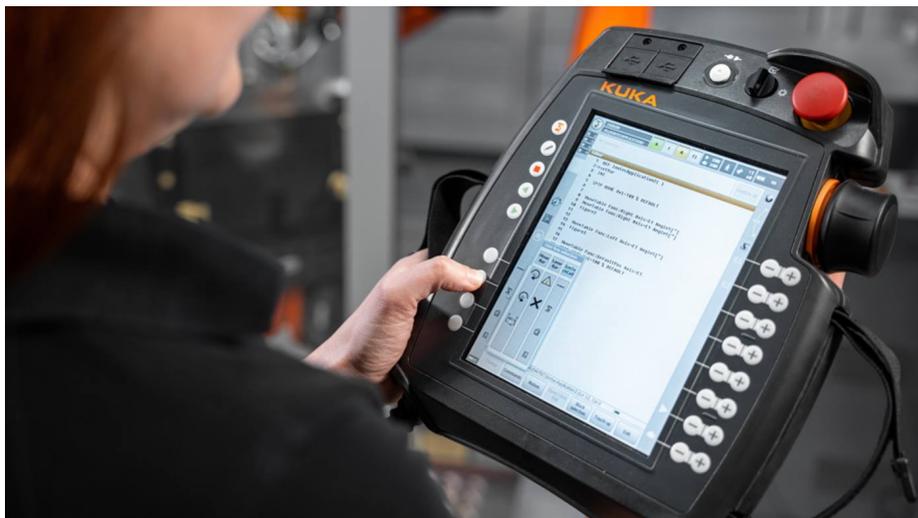


Рисунок 2.3 – Kuka SmartPad

Устройство KUKA SmartPAD осуществляет двусторонний обмен данными с контроллером робота (KRC) через внутреннюю сеть. Аппаратные элементы управления — 6D-мышь, клавиши и энкодеры — реагируют на нажатие клавиш и передают результирующие команды в контроллер манипулятора. Полученные данные обрабатываются планировщиком траекторий KRC, который формирует безопасное и оптимальное движение манипулятора, после чего генерируются управляющие сигналы для двигателей каждой оси.

Одновременно реализуется замкнутый контур обратной связи: энкодеры на двигателях передают данные о фактическом положении осей, что позволяет корректировать движение в реальном времени. Дополнительно на дисплее визуализируется состояние системы — координаты, ошибки и статус программ. Важным элементом является модуль безопасности, который мониторит состояние аварийной кнопки и 3-позиционного переключателя, имея приоритетное право на остановку робота при нарушении условий безопасной эксплуатации.

При этом при использовании Kuka SmartPad существует ряд недостатков.

Эргономика и интерфейс - интерфейс требует специального обучения, малый резистивный сенсорный экран не подходит для решения сложных задач.

Функциональность - жесткая привязка к проприетарной ОС и программному обеспечению KUKA (KSS). Ограниченные возможности для интеграции с внешним ПО (например, системами технического зрения) напрямую.

Безопасность - процесс обучения и ручного управления полностью зависит от человеческого фактора, что повышает риск ошибок и травм при недостаточной квалификации.

Гибкость - робот не может адаптироваться к изменениям в окружающей среде (сдвиг детали, появление нового объекта) без вмешательства оператора.

Для реализации современных задач, требующих адаптивности, интеграции с сенсорами и работы в неструктурированной среде (например, захват произвольно расположенных деталей), необходимы внешние системы управления, которые используют SmartPAD и контроллер KUKA как низкоуровневый исполнительный механизм, получая команды от внешнего компьютера.

Для решения данной проблемы предложена подсистема дистанционного управления промышленным манипулятором Kuka, основанная на языке программирования Python и стеке протоколов - TCP/IP.

Архитектура подсистемы дистанционного управления промышленным манипулятором представлена на рисунке 2.4.

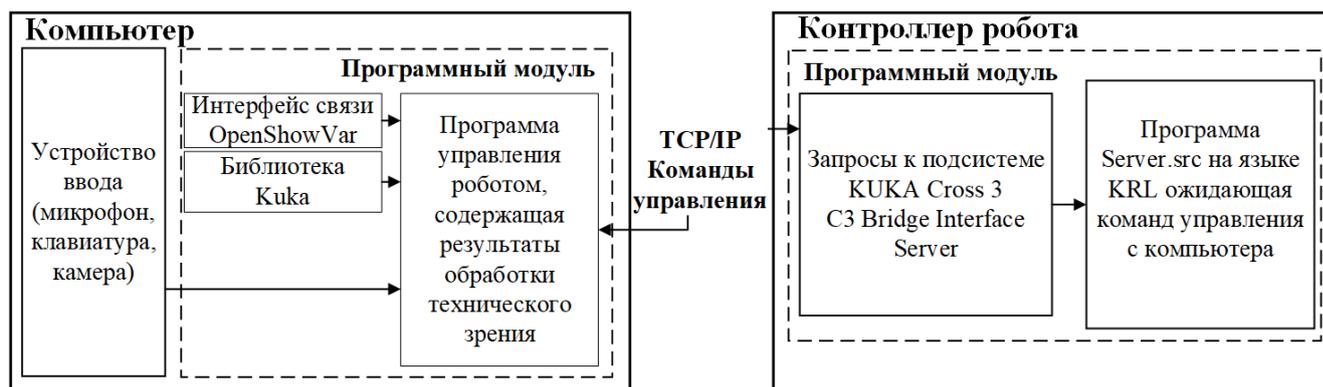


Рисунок 2.4 - Архитектура системы дистанционного управления промышленным манипулятором

В качестве устройства ввода может выступать:

- микрофон, при организации диалоговой системы управления;

- клавиатура, в случае удаленного ручного управления роботом манипулятором;
- камера, при реализации систем управления на основе технического зрения.

С помощью модели протоколов TCP/IP на промышленный манипулятор Kuka передаются команды управления через C3 Bridge Interface Server, который позволяет удаленным клиентам выполнять запросы к подсистеме KUKA Cross 3 и возвращать ответы. Контроллер робота Kuka C5 micro оснащен двумя операционными системами: «ОС VxWorks» для управления роботом в реальном времени и «ОС Windows» для визуализации. Эти две операционные системы должны взаимодействовать друг с другом. Например, для отображения текущих координат объекта «POS\_ACT» на KUKA smartPAD требуется запросить у ОС реального времени текущее значение \$POS\_ACT. Эта связь осуществляется через перекрестный интерфейс KUKA Cross 3. Server.src – программа на языке KRL ожидающая команд управления с внешнего вычислительного устройства и обеспечивающая непосредственное управление движением манипулятора

Для решения задачи дистанционной передачи команд роботу через стек протоколов TCP/IP используют различные программные пакеты, расширяющие возможности взаимодействия с манипулятором:

- KUKA Ethernet KRL
- KUKA RobotSensorInterface
- KUKAVARPROXY

Пакет Ethernet KRL представляет собой программное решение для организации сетевого взаимодействия между KRL-программой, выполняемой на контроллере робота, и внешними устройствами, такими как персональные компьютеры. Функциональность пакета не является автономной и всецело определяется алгоритмом пользовательской KRL-программы. Программист должен явно интегрировать в код специализированные инструкции, обеспечивающие инициализацию сетевого соединения, формирование исходящих пакетов данных, а

также обработку входящих запросов. Таким образом, обмен данными инициируется и управляется непосредственно логикой KRL-приложения, что делает процесс последовательным, но требует ручного описания всех сетевых операций. Технология обмена данными с помощью Ethernet KRL в общем случае не предоставляет способа контроля робота и выхода из аварийных ситуаций. При возникновении каких-либо проблем (столкновений, перегрузки и т.д.), оператор робота будет вынужден использовать пульт управления smartPAD для их решения.

Схожими возможностями обладает фирменный пакет RobotSensorInterface. Его основное назначение – корректировка траектории в процессе движения робота. Как и Ethernet KRL, работа данного программного пакета опирается на выполняющуюся KRL-программу.

Третий продукт – KUKAVARPROXY, встраивается в обмен между двумя операционными системами контроллера робота и позволяет проводить чтение и запись глобальных переменных языка KRL вне зависимости от того, выполняется ли в данный момент KRL программа или нет. Программа KUKAVARPROXY имеет ряд функциональных ограничений и написана на языке Visual Basic образца 1998 года, что требует от оператора системы управления навыков по установке и регистрации специальных программных компонентов. Кроме того, среди значимых недостатков программы можно отметить отсутствие функций контроля выполнения KRL-программы и информирования удаленной системы о причинах возникновения ошибок.

Для устранения недостатков KUKAVARPROXY, в рамках подсистемы используется программный продукт – С3 Bridge Interface [70]. Разработанная в Научно-исследовательском технологическом институте имени С.П. Капицы Ульяновского государственного университета программа обладает следующими преимуществами:

- обладает совместимостью с KUKAVARPROXY, что позволяет использовать его с уже имеющимися сторонними программными решениями;
- позволяет отправлять команды управления на контроллер манипулятора;

- обладает высоким быстродействием за счет реализации на Visual C++ и не требует установки дополнительных программных библиотек в систему управления;
- не имеет ограничений на количество одновременных сетевых подключений и время подключения в режиме простоя.

В документации C3Bridge описаны форматы, в которых должны присылаться сообщения и форматы ответов на них. Для разрабатываемого программного комплекса необходимо знать формат сообщения на чтение координат и формат сообщения на запись координат.

Формат сообщения/ответа на запрос данных с KUKA, представленный в документации C3Bridge, представлен в таблице 2.2.

Таблица 2.2 – Формат сообщения/ответ на данные.

Offset (bytes)	Size (bytes)	Type	Meaning	Structure
REQUEST				
0	2	UINT16	Tag ID	Header
2	2	UINT16	Message Length	
4	1	UINT8	Message Type Value:0	
5	1	UINT16	LVN Length of Variable Name	Payload
7	2	STRING	Variable Name	
RESPONSE				
0	2	UINT16	Tag ID	Header
2	2	UINT16	Message Length	
4	1	UINT8	Message Type Value:0	
5	2	UINT16	LVN Length of Variable Name	Payload
7	LVV	STRING	Variable Name	
Variable	2	UINT16	Error code	FOOTER
Variable	1	BOOL	Success Flag	

Формат сообщений/ответа на запись данных похож на формат сообщений на получение данных (таблица 2.3), в нём добавляется только поле для новых значений.

Таблица 2.3 – Формат сообщения/ ответа на запись данных.

Offset (bytes)	Size (bytes)	Type	Meaning	Structure
REQUEST				
0	2	UINT16	Tag ID	Header
2	2	UINT16	Message Length	
4	1	UINT8	Message Type Value:1	
5	2	UINT16	LVN Length of Variable Name	Payload
7	LVV	STRING	Variable Name	
Variable	2	UINT16	LVV Length of Variable Name	FOOTER
Variable	LVV	STRING	Variable Name	
RESPONSE				
0	2	UINT16	Tag ID	Header
2	2	UINT16	Message Length	
4	1	UINT8	Message Type Value:1	
5	2	UINT16	LVV Length of Variable Name	Payload
7	LVV	STRING	Variable Name	
Variable	2	UINT16	Error code	FOOTER
Variable	1	BOOL	Success Flag	

Все взаимодействия с роботом будут осуществляться через сервер C3Bridge, который должен находиться в одной локальной сети с роботом и удаленным устройством. Подключаться программный комплекс к нему будет по IP-адресу и порту.

Для осуществления подключения в приложении должен быть создан сокет с указанным адресом и отправлен запрос на подключение.

У сокета есть два параметра для создания: домен и тип.

Домен – указывает на то, локальным или удалённым будет сокет. (AF\_INET [71] – удалённый по протоколу IPv4)

Тип – способ взаимодействия с протоколом (SOCK\_STREAM [72] – поддерживает надежное взаимодействие с байтовым потоком, ориентированным на подключение).

## 2.2. Программные модули на внешнем вычислительном устройстве и контроллере робота

В программный модуль на внешнем вычислительном устройстве входит библиотека для управления манипулятором ru-openshowvar и класс функций для работы с роботом на python.

Библиотека OpenShowVar [73] содержит три основные функции:

- `__init__` - для связи с роботом;
- `Read` – для чтения внутренних переменных робота (например: `$POS_ACT` – которая позволяет считывать текущие координаты робота);
- `Write` – для отправки и записи новых значений переменных.

Для удобства считывания и преобразования команд создан класс Kuka [74], в котором прописаны функции для работы с зарезервированными переменными робота, такими как:

- `$BASE` – координаты базы робота, в которой он работает сейчас;
- `$POS_ACT` – текущие координаты робота;
- `$TOOL` - координаты инструмента робота, с которым он работает сейчас;
- `$IN` и `$OUT` – переменные определяющие цифровые входы и выходы.

При внешнем управлении манипулятором KUKA доступ к системным переменным, обозначенным символом «\$», ограничен. Их модификация возможна только через предварительное объявление пользовательских переменных в файле конфигурации `Config.dat`, расположенном в системном каталоге контроллера по пути `C:\KRC\Roboter\KRC\R1\System`. Для дальнейшего использования системных переменных требуется их программное переопределение на языке KRL, где объявленным переменным присваиваются значения соответствующих системных параметров. Таким образом, взаимодействие с защищенными переменными осуществляется опосредованно через пользовательские переменные, что обеспечивает корректную работу системы при внешнем управлении.

В рамках разработанного программного модуля используются следующие пользовательские переменные:

COUNTER\_VARIABLE – необходима для использования пользовательских переменных.

COM\_CASEVAR – осуществляет выбор Case, в котором прописанные определенные операции (например выбор Базы, Инструмента и тд.)

COM\_FRAME – используется для выбора Базы, Инструмента.

COM\_VALUE – используется для выбора Скорости.

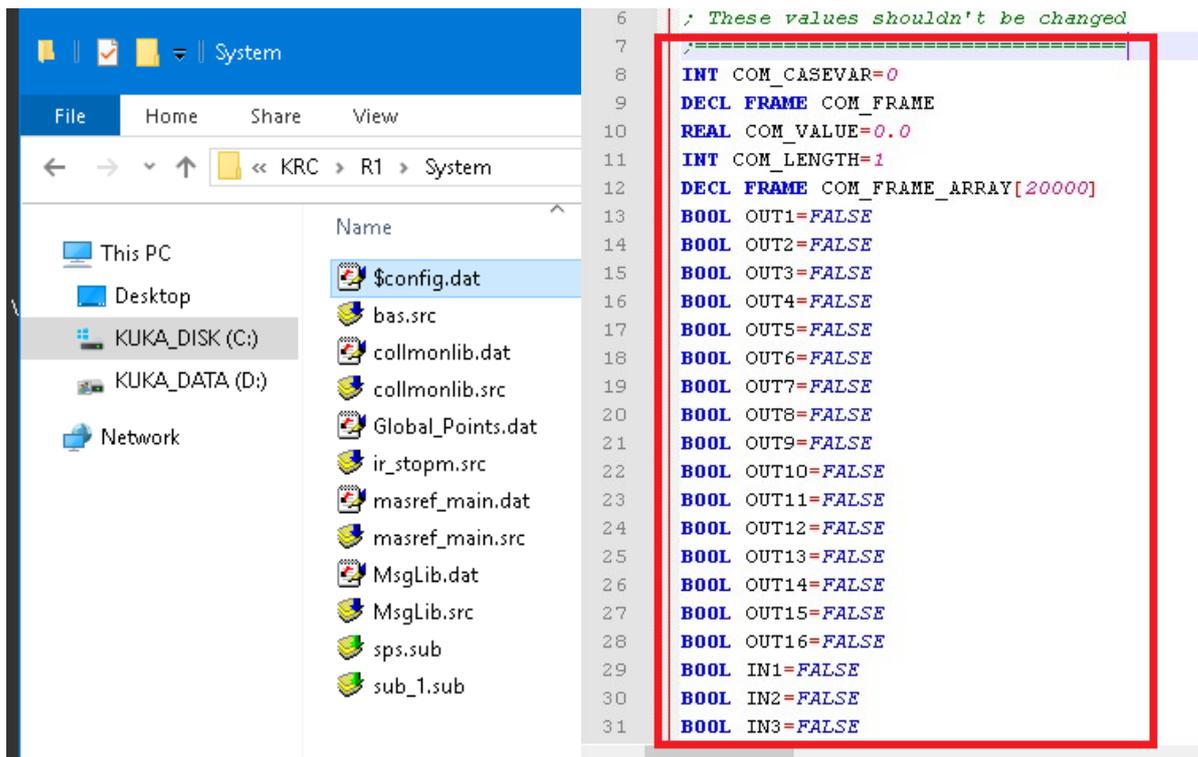
COM\_LENGTH – используется для определения количества переданных точек для движения манипулятора.

COM\_FRAME\_ARRAY – используется для передачи массива точек для движения манипулятора.

OUT1-16 – используется для управления цифровыми выходами.

IN1-16 – используется для считывания информации с цифровых входов.

Переменные, добавленные в файл отображены на рис.2.5.



```

6 ; These values shouldn't be changed
7
8
9 INT COM_CASEVAR=0
10 DECL FRAME COM_FRAME
11 REAL COM_VALUE=0.0
12 INT COM_LENGTH=1
13 DECL FRAME COM_FRAME_ARRAY[20000]
14
15 BOOL OUT1=FALSE
16 BOOL OUT2=FALSE
17 BOOL OUT3=FALSE
18 BOOL OUT4=FALSE
19 BOOL OUT5=FALSE
20 BOOL OUT6=FALSE
21 BOOL OUT7=FALSE
22 BOOL OUT8=FALSE
23 BOOL OUT9=FALSE
24 BOOL OUT10=FALSE
25 BOOL OUT11=FALSE
26 BOOL OUT12=FALSE
27 BOOL OUT13=FALSE
28 BOOL OUT14=FALSE
29 BOOL OUT15=FALSE
30 BOOL OUT16=FALSE
31
32 BOOL IN1=FALSE
33 BOOL IN2=FALSE
34 BOOL IN3=FALSE

```

Рисунок 2.5 – Добавление переменных в файл config.dat

Server.src находится в режиме ожидания до отправки команды с управляющими воздействиями с удаленного устройства. Алгоритм программы Server.src представлен на рис.2.6.

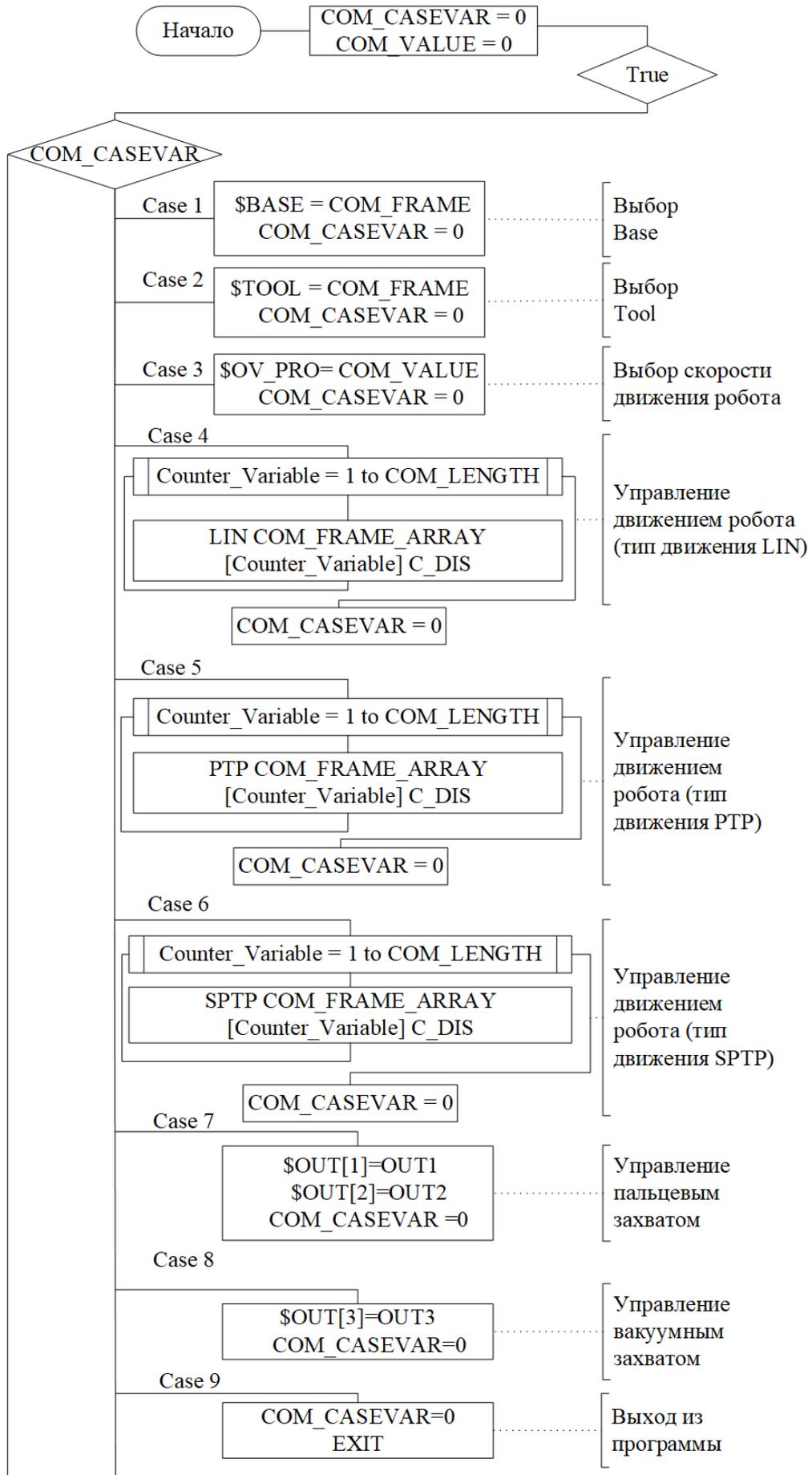


Рисунок 2.6 – Алгоритм программы робота на языке KRL

Согласно алгоритму 2.6 взаимодействие с промышленным роботом KUKA реализованного через механизм switch-case. Данный подход позволяет гибко выбирать требуемые операции в зависимости от передаваемой команды.

Case 1: Выбор системы координат BASE.

Система координат BASE представляет собой декартову систему, описывающую рабочую зону робота или объекта в рабочей зоне. Она определяется относительно системы координат WORLD и по умолчанию BASE[0] идентична ей.

Пользователь может определять до 32 систем координат BASE в зависимости от конфигурации контроллера. Изменение активной системы координат выполняется присваиванием переменной BASE значения из массива \$BASE\_DATA[№], где № — индекс требуемой базы. Это позволяет автоматически трансформировать всю траекторию движения без пересчета отдельных точек.

Информация о координатах BASE в контроллере робота хранятся в виде FRAME. FRAME – это структура данных, в которой содержится шесть параметров: {X, Y, Z, A, B, C}. Эти параметры соответствуют шести параметрам в представлении системы угловых координат: X, Y, Z,  $\alpha$ ,  $\beta$ ,  $\gamma$ , где  $\alpha = A$ ,  $\beta = B$ ,  $\gamma = C$ .

Пример системы координат основания робота (\$ROBROOT), основной системы координат (\$BASE), системы координат фланца (\$NULLFRAME (FLANGE)) и системы координат инструмента (\$TOOL) приведен на рис.2.7.

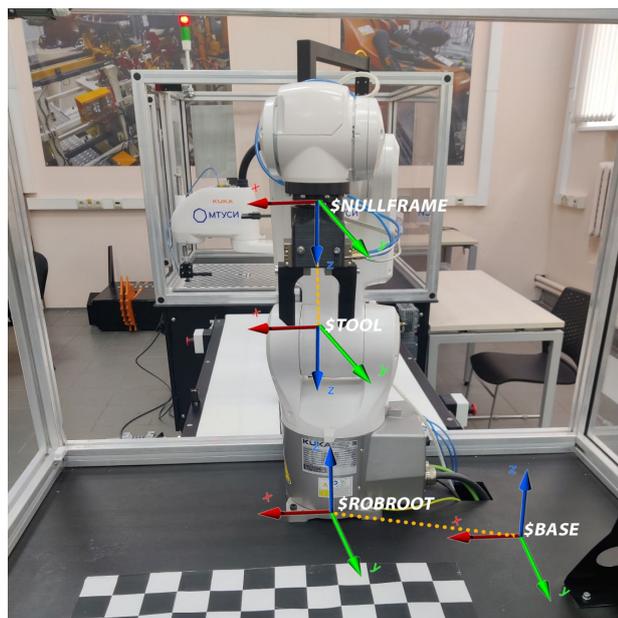


Рисунок 2.7 - Выбор BASE и TOOL

**\$ROBROOT**

- Зафиксировано в основании робота;
- Это начало координат робота

**\$BASE**

- Свободно определяемая система координат

**\$NULLFRAME**

- Система координат FLANGE зафиксирована на фланце робота;
- Начало координат – середина фланца робота;
- Это отсчетная точка для системы координат TOOL

**\$TOOL**

- Свободно определяемая система координат;
- Начало системы координат TOOL, обозначается как TCP (Tool Center Point)

**Case 2: Выбор инструмента (TOOL).**

Аналогично выбору базы, в Case 2 происходит установка инструмента путем присваивания переменной TOOL значения из массива \$TOOL\_DATA[№]. Система координат инструмента (TOOL) определяет положение и ориентацию TCP (Tool Center Point) относительно фланца робота.

**Case 3: Установка скорости движения.**

Скорость движения робота задается в процентах от максимальной через переменную \$OV\_PRO [75]. Для регулирования скорости движения робота передается значение скорости в %, где максимальное значение равняется 100 для Kuka KR4 R600.

**Case 4–6: Движение по заданным траекториям.**

Для отправки координат для движения робота необходимо отправить структуру данных FRAME в виде:

{FRAME: X значения, Y значения, Z значения, A значения, B значения, C значения},

где  $X$ ,  $Y$ ,  $Z$  – координаты относительно выбранной базы,  $A$ ,  $B$ ,  $C$  – углы поворота,  $A$  – угол, вращающийся вокруг оси  $Z$ ,  $B$  – угол, вращающийся вокруг оси  $X$ ,  $C$  – угол, вращающийся вокруг оси  $Y$ .

Значения в робота могут быть переданы как в формате `float`, так и в формате `integer`.

Case 4: Последовательная отправка массива с координатами точек для движения манипулятора типом движения «линейное движение» (LIN).

Case 5: Последовательная отправка массива с координатами точек для движения манипулятора типом движения «от точки к точке» PTP.

Case 6: Последовательная отправка массива с координатами точек для движения манипулятора типом движения SPTP.

Case 7–8: Управление захватными устройствами.

В манипуляторах Kuka для управления захватными устройствами используются цифровые выходы и входы (I/O interfaces). Например, в контроллере Kuka C5 Micro для этого используется интерфейс XG12.

Интерфейс ввода-вывода XG12 имеет 16 входов и 16 выходов:

- 16 входов, которые могут работать с датчиками;
- 16 выходов, которые могут работать с исполнительными устройствами.

В зависимости от захватного устройства управление им осуществляется с помощью одного или нескольких цифровых выходов, которые принимают значение `True` или `False`. Например, параллельный пальцевый пневматический захват может быть подключен к пневмораспределителю, который в свою очередь подключен к интерфейсу XG12 в `OUT[1]` и `OUT[2]`, тогда в Case должно быть прописано управление этими выходами. Данные переменные являются системными и запрещены для редактирования в случае управления с внешнего устройства, но существует возможность их переопределения через другие переменные, которые ранее добавлены в файл `config.dat`.

Таким образом предполагается:

Case 7: Управление пальцевым пневматическим захватом через выходы `OUT[1]` и `OUT[2]`.

Case 8: Управление вакуумным захватом через выход OUT[3].

В случае добавления новых захватных устройств необходимо добавить новый Case, который будет отправлять сигнал на соответствующие цифровые выходы.

Case 9: Использование подсистемы дистанционного управления совместно с ПЛК в автоматическом режиме.

Для управления передаваемыми от ПЛК номерами программ используется организационная программа **Cell.src**. Она всегда находится в папке **R1**. Как и любую обыкновенную программу, программу Cell также можно индивидуально настроить, однако при этом структура программы должна сохраняться.

В программе **Cell.src** используется аналогичная структура switch-case, где за выбор конкретного Case отвечает ПЛК. Следовательно, необходимо создать Case в программе **Cell.src**, в котором будет содержаться вызов программы **Server.src**.

После выполнения операции манипулятором необходимо вызвать Case 9, который реализует алгоритм выхода из программы **Server.src**. Case 9 позволяет передать в ПЛК информацию о том что манипулятор выполнил задачу. Например, с помощью системы дистанционного управления манипулятор захватил объект и поставил его на конвейер, тогда выход из программы может стать сигналом для запуска движения конвейера.

### 2.3. Дистанционное управление промышленным манипулятором

Особенностью систем управления промышленными манипуляторами является последовательная обработка команд в рамках выполняемой программы. В случае асинхронного получения нескольких управляющих воздействий от внешнего вычислительного устройства, контроллер обрабатывает их в порядке поступления, однако результатом будет немедленный переход к исполнению последней полученной команды. Это поведение обусловлено однопоточной моделью выполнения и отсутствием встроенного буферизации команд в стандартных конфигурациях, что критично учитывать при проектировании систем дистанционного управления. В рамках представленной реализации в классе Кука

для каждой функции сделан процесс проверки значения переменной COM\_CASEVAR. Если переменная COM\_CASEVAR равна нулю, то робот не выполняет никаких операций и следовательно он не находится ни в одном из Case.

Пример функции Lin\_continuous представлен на рисунке 2.8. Временная задержка вначале функции установлена для последовательного приема команд, если ее не будет, то роботу отправятся все команды управления одновременно, что приведет к их некорректной обработке. На вход функции lin\_continuous поступает значение FRAME в виде массива данных (arg), в каждой строке которого записаны координаты точек движения робота. В переменную COM\_LENGTH записывается количество строк полученного массива, а в переменную COM\_CASEVAR записывается номер CASE, в который должна перейти программа на работе (Server.src) для выполнения движения робота. Далее производится проверка значения переменной COM\_CASEVAR и как только она станет равна 0 роботу могут быть отправлены новые инструкции.

---

**Algorithm 1** lin continuous

---

**Require:** *DataArrayFrame*

**Ensure:** *RobotMovement*

*delay* = 0.1

*SendFrameArray(DataArrayFrame)*

*OpenShowVar.Write(COMLENGTH, Length(DataArrayFrame))*      ▷

Passing the length of an array to the COMLENGTH

*OpenShowVar.Write(COMCASEVAR)*      ▷ Selecting a Case

**while** *OpenShowVar.Read(COMCASEVAR) != 0* **do** *continue*

**end while**

---

Рисунок 2.8 – Алгоритм функции lin\_continuous

Тестирование проводилось на роботах KUKA KR4 R600 и KUKA SCARA KR6 R500 Z200 на базе контроллеров KUKA KR C5 micro [76].

На рисунке 2.9 показан пример отправки команд: выбор BASE 8, TOOL 10, установки скорости движения 30% и перемещение робота в точку с координатами X=282.73, Y=122.48, Z=451.23, A=90,57, B=0, C=180.

**Algorithm 2** Testing the remote control subsystem

**Require:** *ip, port, Base, Tool, Speed, Coordinat* ▷ IP address and port of the robot, Base and Tool number, movement speed and coordinates

**Ensure:** *SetBase, SetTool, RobotMovement, OpenCloseGripper*

*openshowvar(ip, port)*

▷ Connecting to the robot

*SetBase*

*SetTool*

*SetSpeed*

*ReadCartesian*

▷ Reading current coordinates robot

*TrajectoryArr(np.array(x, y, z, A, B, C))*

▷ Forming an array of coordinates

*LinContinuous(TrajectoryArr)*

▷ Sending a trajectory to the robot

*OpenGrip*

*CloseGrip*

Рисунок 2.9 – Тестирование системы дистанционного управления

В результате выполнения программы производится выбор системы координат BASE, смещение системы координат World относительно установленного инструмента TOOL и установка скорости выполнения программы роботом.

Следующим этапом происходит движение робота в заданные координаты и после достижения конечной точки открывается и закрывается пальцевый захват. Результаты выполнения представлены на рис. 2.10.



Рисунок 2.10 - Результат выполнения программы

При использовании системы технического зрения команды управления могут быть переданы либо в виде массива координат, которые робот будет выполнять последовательно, либо в процессе организуется потоковая передача, при параллельной работе робота и системы компьютерного зрения.

Подсистема дистанционного управления предполагает работу в ручном и автоматическом режиме, для каждого из которых предусмотрены функции безопасности.

При работе в ручном режиме, необходимо постоянно удерживать клавишу пуска, при ее отпуске робот прекратит выполнение процедуры, а при повторном нажатии продолжит выполнение с момента ОСТАНОВА.

При работе в автоматическом режиме в алгоритме предусмотрены две основные возможности останова тех.процесса:

Движение робота и обработка информации, поступающая с внешнего вычислительного устройства распараллелены. Для этого создан дочерний процесс, в котором происходит обработка и отправка команд непосредственно роботу.

В процессе работы создается очередь, которая очищается если робот не движется или если он дошел до определенной точки. Как только очередь пуста в процесс, отвечающий за движение робота, отправляются новые значения, полученные в результате работы алгоритма.

Параллельные вычисления позволяют останавливать выполнение программы, с помощью функции библиотеки Кука. В процессе работы можно в любой момент времени установить скорость выполнения 0%, что позволит остановить технологический процесс, а по завершению процесса устранения неполадок запустить его заново, установив соответствующую скорость выполнения.

Прервать технологический процесс можно с помощью устройства АВАРИЙНОГО ОСТАНОВА РТК. Для продолжения работы следуют деблокировать кнопку АВАРИЙНОГО ОСТАНОВА посредством вращения, квитировать появившееся после этого сообщение и перезапустить тех.процесс.

### **Выводы по главе**

1. Во второй главе описана разработка подсистемы дистанционного управления промышленным манипулятором Кука, обеспечивающая возможность задания команд управления с внешнего вычислительного устройства. Модульная

архитектура включает: внешнее вычислительное устройство, с которого осуществляется контроль и управление основными функциями манипулятора и контроллером робота, с расположенном на нем программном обеспечении, включающее C3 Bridge Interface и программа на языке KRL, обеспечивающая непосредственное управление движением манипулятора.

2. Ключевыми функциональными возможностями подсистемы дистанционного управления является: выбор системы координат (BASE) и инструмента (TOOL); управление скоростью движения робота; передачу координат для линейного (LIN) и точечного (PTP) перемещения; контроль цифровых выходов для управления захватными устройствами; обратная связь, включая чтение текущих координат и статуса выполнения команд.

3. Для передачи данных с внешнего вычислительного устройства на контроллер робота использован протокол TCP/IP, обеспечивающий надежную связь и минимальные задержки. Для обеспечения высокоуровневого взаимодействия с переменными KRL-программ контроллера KUKA напрямую из Python-окружения реализована библиотека, упрощающая интеграцию с внешними системами. Многопоточность, реализованная в подсистеме с помощью высокоуровневых языков программирования, обеспечивает возможность управлять манипулятором: получать информацию о положении захвата, скорости движения, напряжении на приводах и т.д. в процессе перемещения фланца манипулятора, что позволяет в критических ситуациях остановить манипулятор и не допустить повреждения фланца, захватного устройства и объекта захвата.

4. Главными отличиями разработанного решения является: возможность адаптации под различные контроллеры KUKA и интеграция с системами компьютерного зрения, поддержка проводного и беспроводного каналов связи, что расширяет сферы применения (например, для голосового или жестового управления, использование открытого ПО (Python, C3 Bridge) позволяет исключить использование проприетарных решений.

### **ГЛАВА 3. РАЗРАБОТКА МЕТОДА КЛАССИФИКАЦИИ И ЗАХВАТА МАНИПУЛЯТОРОМ ПРОИЗВОЛЬНО РАСПОЛОЖЕННЫХ ФЕРРОМАГНИТНЫХ ДЕТАЛЕЙ ИЗ НАКОПИТЕЛЯ**

В главе представлена разработка метода классификации с помощью нейронной сети и последующего захвата манипулятором объекта, произвольно расположенного в транспортно-накопительном устройстве. Метод обеспечивает определение границ накопителя в рабочей зоне манипулятора, сегментацию ферромагнитных объектов, используя алгоритмы компьютерного зрения на основе сверточных нейронных сетей, выбор объекта захвата, определение пространственных координат объекта захвата, определение объекта возле стенок накопителя, перемещение фланца манипулятора и объекта магнитным пневматическим захватным устройством.

#### **3.1. Этапы метода классификации и захвата манипулятором произвольно расположенных ферромагнитных деталей из накопителя**

Предлагаемый в работе метод сортировки ферромагнитных объектов содержит шесть основных этапов, приведенных на рисунке 3.1.

##### **1. Этап определения рабочей зоны манипулятора.**

Для предотвращения столкновения с ограждениями робототехнической ячейки необходимо исключить из рабочей зоны области, в которых манипулятор не должен находиться. К таким зонам относится пространство за пределами накопителя с заготовками и границы самой ячейки.

##### **2. Этап классификации объектов в накопителе.**

На данном этапе с помощью нейросетевой модели Yolov8 выполняется семантическая сегментация изображения, поступающего с камеры. Модель выделяет все ферромагнитные объекты, находящиеся в зоне видимости, отделяя их от фона и друг друга.

##### **3. Этап определения объекта захвата.**

Для захвата объекта роботом необходимо удовлетворить условию, чтобы поверх интересующего объекта не находились другие, иначе робот не сможет захватить объект или захватит объект, но в процессе затронет другие объекты, что может повредить фланец робота или захват. Следовательно, необходимо определять объект для захвата, который не перекрыт другими и захват которого не приведет к повреждению фланца или ЗУ.

#### 4. Этап определения объекта возле стенок накопителя.

Для решения проблемы, связанной с расположением предметов возле стенок накопителя, в рамках диссертации, используется дополнительно дообученная ИНС YOLOv8 производящая определение границ накопителя. Специальный алгоритм сравнивает координаты ограничивающей рамки объекта с границами накопителя. Если объект признается находящимся у стенки, для его захвата может применяться специальная стратегия или он может быть исключен из кандидатов.

#### 5. Этап определения пространственных координат объекта.

В рамках этого этапа выполняется определение координат центра объекта захвата через фокусное расстояние камеры и расстояние до объекта по вертикальной оси и определение углов наклона объекта, с помощью алгоритма, основанного на соответствии между данными, получаемыми с RGB-D сенсора и 3D моделью существующего объекта.

#### 6. Этап захвата и перемещение объекта захвата.

Полученные координаты и углы ориентации передаются в контроллер робота через разработанную подсистему дистанционного управления. После позиционирования фланца над объектом осуществляется его захват с помощью магнитного пневматического захватного устройства и последующее перемещение в целевую точку.

Согласно алгоритму, приведенному на рисунке 3.1 этапы 2, 3, 4 повторяются дважды, для минимизации погрешностей вычислений координат центра объекта захвата и его углов наклона относительно фланца манипулятора.

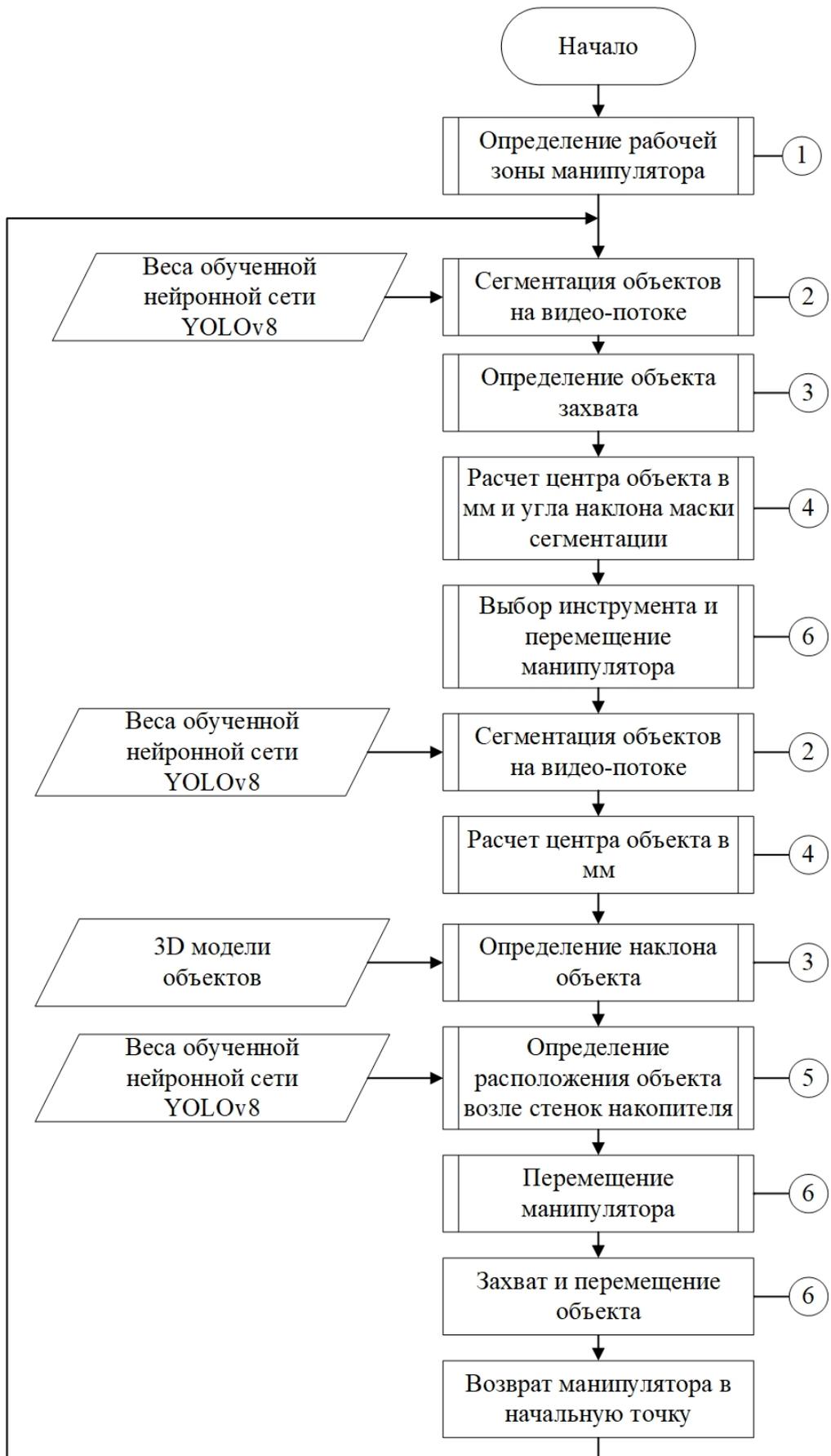


Рисунок 3.1 – Основные этапы метода классификации и захвата манипулятором произвольно расположенных ферромагнитных деталей в накопителе

Таким образом, для решения проблемы сортировки ферромагнитных деталей, произвольно расположенных в накопителе, предложена модульная архитектура системы управления промышленным роботом, отличающаяся от структуры приведенной на рисунке 1.2 двумя ключевыми подсистемами: подсистемой удаленного управления в которой реализованы методы двусторонней связи и параллельной обработки команд, а также подсистемой компьютерного 3D-зрения, обеспечивающей определение пространственного положения объектов за счет комбинации RGB-D сенсоров и нейросетевых алгоритмов, специально оптимизированных для задач сегментации ферромагнитных деталей в условиях производственной среды. Это позволило реализовать гибкое беспрограммное управление, основанное на анализе сенсорной информации. Модульная структура системы управления представлена на рисунке 3.2.

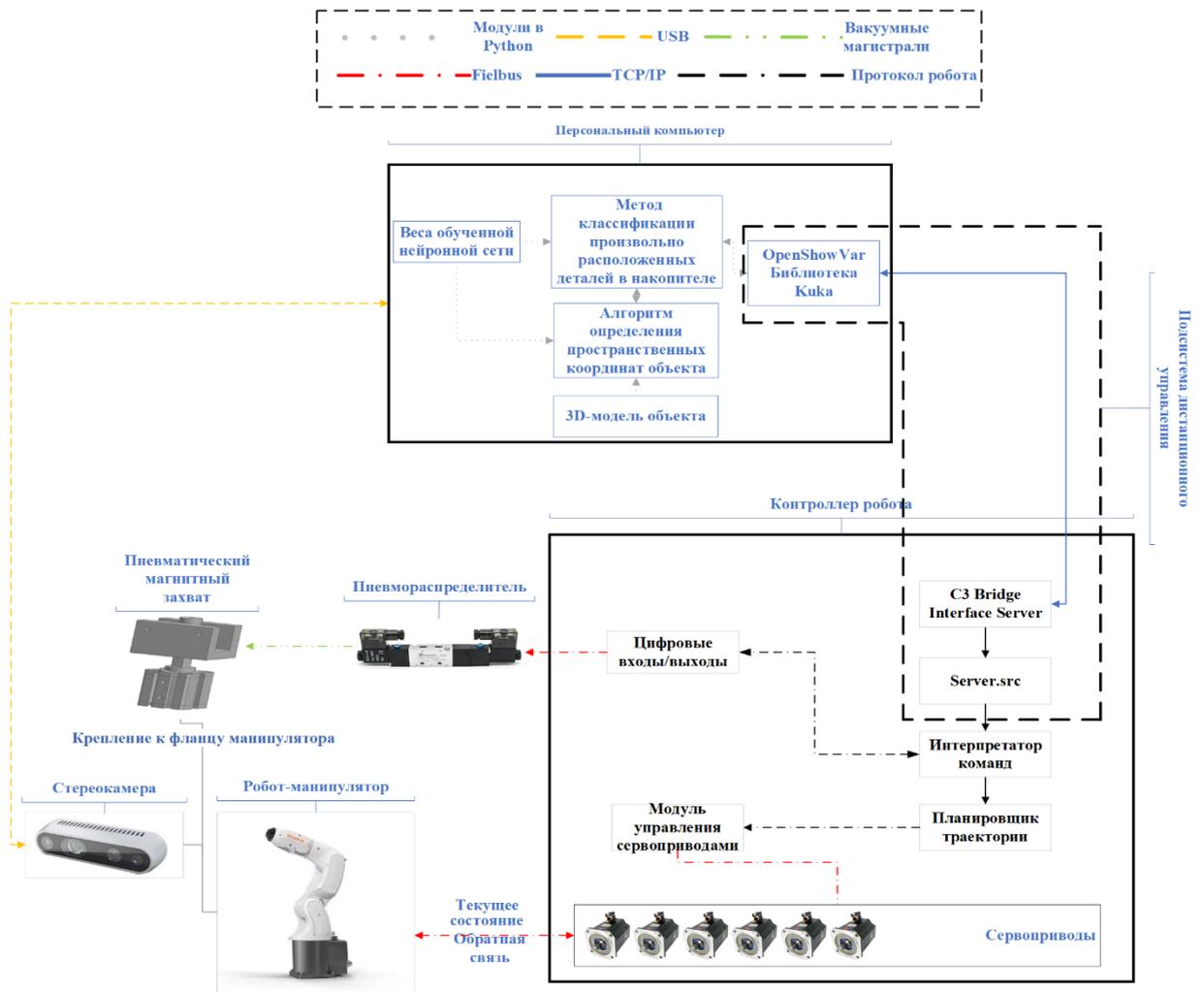


Рисунок 3.2 – Модульная архитектура системы управления промышленным манипулятором для сортировки ферромагнитных деталей

### 3.2. Выбор камеры бинокулярного зрения и ее размещение в рабочей зоне робота

Разрабатываемый метод предполагает классификацию и определение пространственных координат объекта с помощью технологий компьютерного зрения в видеопотоке, получаемом с помощью RGB-D камеры. При этом важным техническим решением, обеспечивающим необходимое качество видеопотока, является выбор и установка камеры при реализации процесса сортировки.

В работе [77] произведено исследование характеристики самых распространённых RGB-D камер Microsoft Kinect v2, Orbbec Astra S и серии Intel D400. Для каждого из этих устройств проведено два типа экспериментов чтобы оценить, точность камер на разных расстояниях и качество восстановления геометрии объекта. Для оценки качества реконструкции геометрии объекта измерялось распределение расстояний между полученными точками и реальными геометрическими моделями, в ходе этих испытаний сравнивались плоскости, цилиндры и сферы. В результате автор пришел к выводу, что наилучшие характеристики камер глубины с точки зрения попиксельной точности и точности восстановления геометрии объекта находятся в диапазоне до 1000 миллиметров. Сводная таблица полученных автором характеристик приведена в таблице 3.1.

Таблица 3.1 – Сводная таблица характеристик RGB-D сенсоров [77].

	Расстояние от камеры до объекта 1000 мм		Расстояние от камеры до объекта 2500 мм	
	Случайная ошибка	Систематическая ошибка	Случайная ошибка	Систематическая ошибка
Kinect V2	1.5 мм	5 мм	2 мм	10 мм
Orbbec Astra S	1 мм	8 мм	5 мм	96 мм
D415	1.5 мм	2 мм	15 мм	25 мм

В работе [78] производится оценка точности работы Intel RealSense D415. В частности, проводятся испытания по оценке работоспособности устройства в ближнем диапазоне (т. е. 100–1000 мм). Испытания проводятся в соответствии со стандартом VDI/VDE 2634 Часть 2 («Оптические 3D-измерительные системы») [79]. В результате автор получает результаты, что средняя ошибка смещения по

вертикальной оси от реальной плоскости равняется 2,52 мм в диапазоне от 500 мм до 1000 мм, ошибка в диапазоне от 150 мм до 500 мм составляет в среднем 0,99 мм.

В рамках исследования использовалась камеры Intel RealSense D415, которая оснащена цветной камерой и системой, позволяющей вычислять глубину, включающая две ИК-камеры и ИК-проектор. Проектор улучшает работу стереосистемы, подсвечивая сцену структурированным инфракрасным светом, что повышает точность построения карты глубины.

Характеристики Intel RealSense D415 представлены в таблице 3.2.

Таблица 3.2 – Характеристики Intel RealSense D415

Параметр	D415
Определение глубины	Двойной ИК-луч
Размер пиксела	1.4 x 1.4 мкм
Размер поля глубины	63.4° x 40.4°
Разрешение поля глубины	1280 x 720
Параметры RGB-потока	1080p @ 30 FPS
Диапазон глубины	0.16 – 10 м
Размер камеры (Ш x Г x В)	99 x 20 x 23 мм

На рис.3.3. представлено поле зрения камеры Intel RealSense в диапазоне 500–1500 миллиметров.

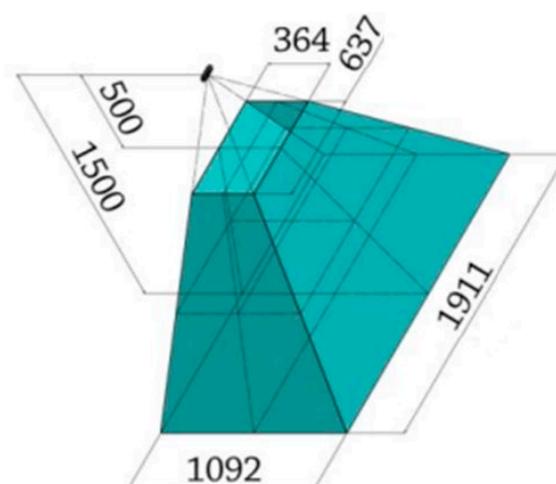


Рисунок 3.3 - Поле зрения камеры Intel RealSense в диапазоне 500–1500 миллиметров

В исследовании используется разрешение видео потока 640×480 пикселей и частота кадров 30 кадров в секунду, которые являются стандартным для библиотеки OpenCV (Open Source Computer Vision Library) [81].

Размещение камеры в рабочей зоне робота.

Существуют два разных подхода к использованию камеры в задаче захвата объекта манипулятором, которые могут быть названы как «eye in hand» [82], размещение камеры непосредственно на манипуляторе робота, и «eye to hand», камера находится вне манипулятора, например, на статической платформе или на стене.

Конфигурация «eye in hand» обеспечивает высокую точность позиционирования непосредственно в зоне захвата и не имеет зон, закрытых от обзора самим манипулятором, что идеально подходит для задач с требованием высокой точности. Однако её поле зрения ограничено и постоянно меняется при движении робота. Конфигурация «eye to hand» предоставляет стабильное и неизменное поле обзора всей рабочей зоны, что упрощает калибровку и планирование движений. Её ключевой недостаток — наличие мёртвых зон, которые могут быть закрыты манипулятором, а также потенциально меньшая точность из-за фиксированного угла обзора и большего расстояния до объекта.

В рамках диссертационного исследования выбран подход «eye in hand» позволяющий достичь лучшей точности и контроль за манипулятором, так как камера находится на самом инструменте, исключая возможность погрешностей, связанных с позиционированием камеры отдельно от робота. Крепление для камеры, изготовленное аддитивным способом и установленное на фланце манипулятора, представлено на рис. 3.4.



Рисунок 3.4 – Закрепление камеры на фланце манипулятора

При таком типе крепления камера расположена параллельно основанию, на котором расположены объекты захвата, следовательно нет необходимости производить расчеты поворота камеры относительно основания.

### 3.3. Определение границ накопителя в рабочей зоне манипулятора

Для решения поставленной задачи необходимо производить захват объектов, расположенных в накопителе. В связи с этим требуется ограничить область обработки данных системой компьютерного зрения границами накопителя. Для этого выполняется детектирование его контуров на видеопотоке с последующим наложением бинарной маски, исключающей из анализа фоновые области. В качестве накопителя рассматривается пластиковый контейнер с габаритами 600×400 мм. После определения его границ на изображении производится маскирование внешних областей видеопотока, что обеспечивает минимизацию ложных срабатываний и повышение эффективности алгоритма.

Алгоритм определения границ накопителя представлен на рис.3.5.



Рисунок 3.5 - Алгоритм определения границ накопителя

Для поиска автоматического выделения границ накопителя применяется алгоритм обнаружения краев Canny, который приведен на рис.3.6.

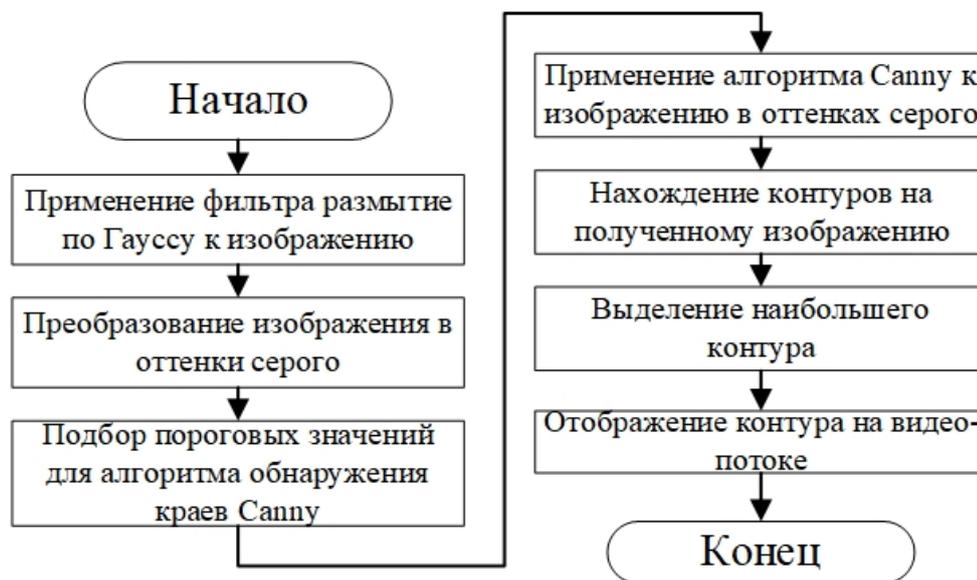


Рисунок 3.6 - Алгоритм выделения границ накопителя

Алгоритм Canny (Canny Edge Detection) [83] является одним из наиболее эффективных методов выделения границ, обеспечивающим высокую точность и устойчивость к шуму. Метод реализуется в четыре этапа: сглаживание изображения для подавления шума, вычисление градиентов яркости, подавление немаксимумов (Non-Maximum Suppression) и пороговую обработку с гистерезисом (Hysteresis Thresholding).

- Сглаживание изображения (Гауссово размытие). Исходное изображение фильтруется гауссовым фильтром [84] для подавления шумов, которые могут вызывать ложные срабатывания при обнаружении границ.

- Нахождение градиента интенсивности изображения (оператор Собеля). Для каждого пикселя вычисляются градиенты по осям X и Y с помощью операторов Собеля или Превитта [85]. На основе этих значений определяется величина и направление градиента, которые указывают на резкие изменения яркости — потенциальные границы.

- Подавление немаксимумов (Non-Maximum Suppression). Чтобы получить тонкие и четкие границы, алгоритм оставляет только пиксели с

локальными максимумами градиента в направлении границы. Остальные пиксели подавляются, что устраняет размытие и утолщение границ.

- Пороговая обработка с гистерезисом (Hysteresis Thresholding). Используются два порога: высокий ( $\maxVal$ ) и низкий ( $\minVal$ ). Пиксели с градиентом выше  $\maxVal$  считаются сильными границами, ниже  $\minVal$  - отбрасываются. Пиксели между порогами сохраняются только если они связаны с сильными границами, что обеспечивает непрерывность контуров и отсеивает случайный шум.

Полученное, в результате применения метода Canny, бинарное изображение передается на вход алгоритма, реализующего метод поиска контуров FindContours, выделяющего максимальный контур в рабочей зоне робота. Для данного контура вычисляются и возвращаются следующие параметры: минимальное значение по оси X, ширину контура, минимальное значение по оси Y, длину контура.

В результате выполнения алгоритма 3.6. на видео потоке будет выделен накопитель, как показано на рис.3.7.

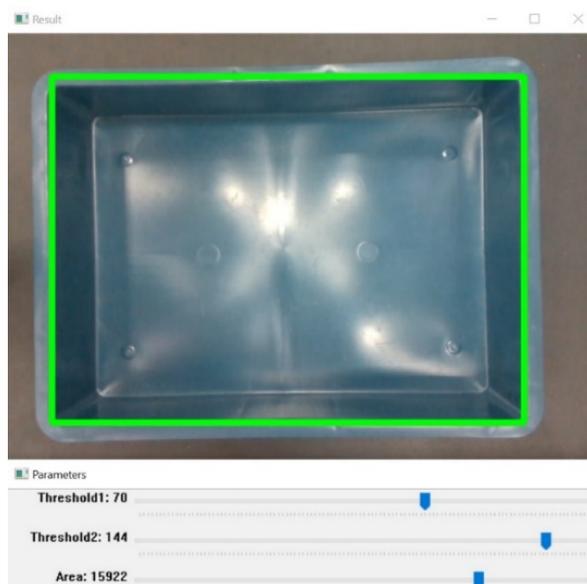


Рисунок 3.7 - Выделение накопителя в рабочей зоне робота

В случае частичного нахождения накопителя в кадре происходит автоматическая подстройка манипулятора для центрирования камеры, что позволяет не допустить дополнительных ложных срабатываний, согласно алгоритму 3.8.

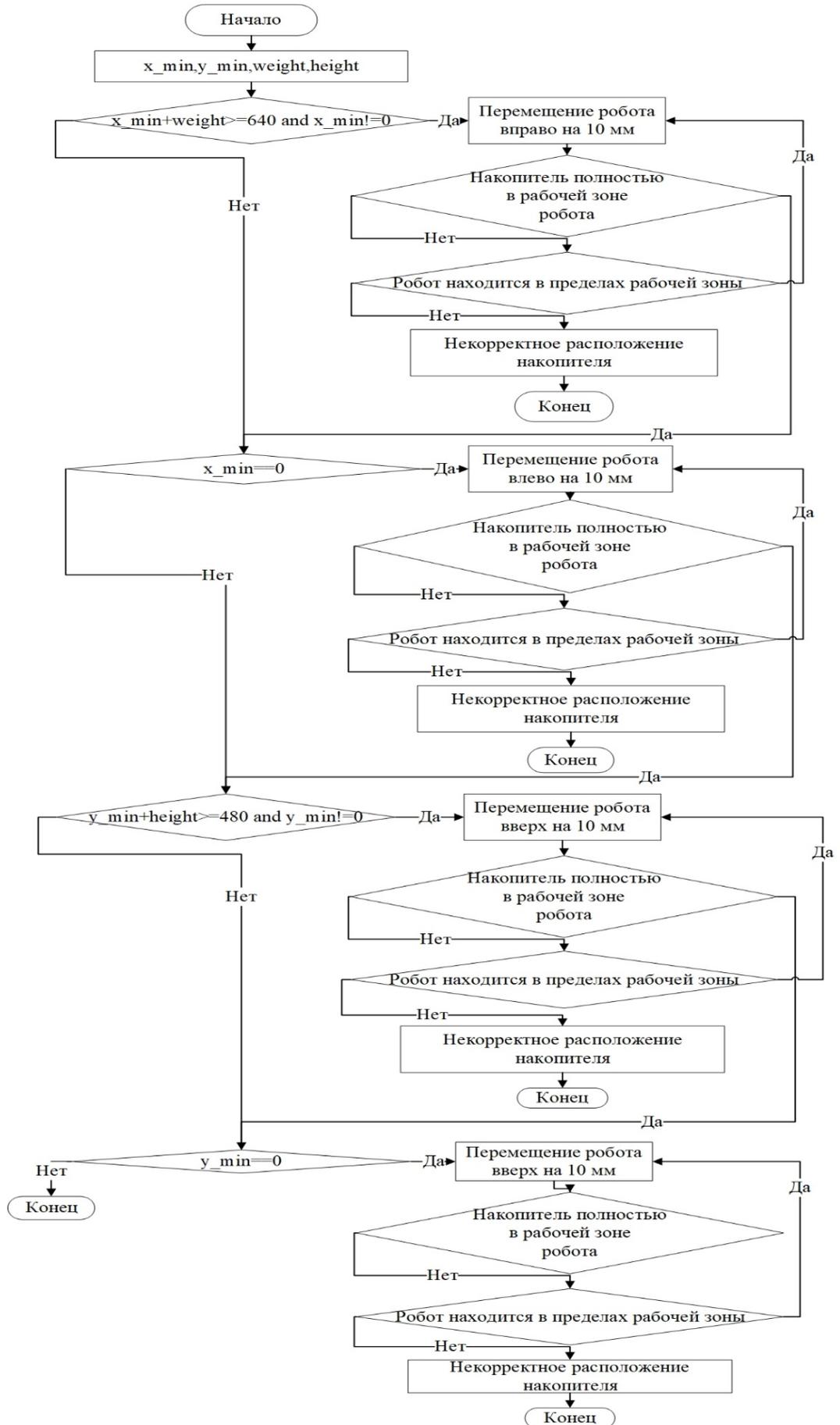


Рисунок 3.8 - Алгоритм подстройки рабочей зоны робота под накопитель

После выделения границ накопителя на экран оператору выводится изображение с отрисованным контуром и запрашивается подтверждение корректности выделения. В случае некорректного выделения накопителя оператор может вручную двигать робота клавишами с клавиатуры по 10 миллиметров пока накопитель не будет полностью в кадре, получаемого с камеры. Далее на видеопоток накладывается бинарная маска, исключая из анализа фоновые области.

### **3.4. Нейросетевая классификация объектов в накопителе**

Для классификации объектов в современной промышленной робототехнике используются нейросетевые алгоритмы компьютерного зрения. Сверточные нейронные сети (CNN) составляют основу для большинства решений, обеспечивая высокоточное распознавание и классификацию объектов по их визуальным признакам.

Самыми популярными архитектурами сверточных нейронных сетей, для задач компьютерного зрения являются:

- Region Proposals (с использованием различных региональных сверточных нейронных сетей: R-CNN, Fast R-CNN, Faster R-CNN, cascade R-CNN) [86];
- Single-Shot MultiBox Detector [87];
- You Only Look Once [88];

Нейросети Faster R-CNN достаточно популярны в решении задач детекции объектов. Принцип работы Faster R-CNN построен на разделении исходного изображения на области, в которых возможно нахождение целевых объектов. Эти области подаются на вход сверточным слоям, ответом которых являются векторы признаков для каждой из областей. С помощью алгоритма опорных векторов происходит классификация векторов признаков, в результате которой возможно обнаружение целевого объекта.

Нейросети SSD имеют большую скорость работы по сравнению с Fast R-CNN, поскольку в их архитектуре отсутствует шаг деления изображений на области, в которых могут находиться целевые объекты. Принцип работы нейросети

SSD следующий: на первом этапе формируются карты признаков для различных масштабов с помощью сверточных слоев, на втором этапе применяется сверточный фильтр, ответом которого являются рамки, описывающие целевые объекты, на третьем этапе оценивается вероятность присутствия объекта в рамках [89].

Среди современных архитектур сверточных нейронных сетей модель YOLO (You Only Look Once) является оптимальной для решения задач в реальном времени. Алгоритм реализует принцип единовременного детектирования, при котором входное изображение делится на размерностью  $S \times S$ . Каждая ячейка сетки отвечает за предсказание  $B$  bounding boxes и вероятностей принадлежности к  $C$  классам. Выходной тензор сети имеет размерность  $S \times S \times (B \times 5 + C)$ . Для устранения избыточных детекций применяется алгоритм non-maximum suppression (NMS), который фильтрует перекрывающиеся предсказания на основе пороговых значений confidence score и intersection over union (IoU).

В исследовании [90] сравниваются 3 основных алгоритма обработки изображений: Single Shot Detection, Faster R-CNN, YOLO. В этом сравнительном анализе с использованием набора данных Microsoft COCO (Common Object in Context) оценивается производительность этих трех алгоритмов, а также анализируются их сильные стороны и ограничения на основе таких параметров, как точность (accuracy), precision и F1 score. По результатам анализа авторы сделали выводы, что качество работы любого из алгоритмов по сравнению с двумя другими в значительной степени определяется вариантами использования, в которых они применяются.

В исследовании [91] производится сегментация объектов в сельском хозяйстве для таких задач, как автоматизированный или роботизированный сбор урожая. В статье авторы сравнивают YOLOv8 и Mask R-CNN для сегментации объектов в различных условиях сада в двух наборах данных. Результаты показали, что YOLOv8 работает лучше, чем Mask R-CNN, достигая хорошей точности (precision) и почти идеальной полноты (recall) в обоих наборах данных при доверительном пороге 0,5. В частности, для набора данных 1 YOLOv8 достиг точности 0,90 и полноты 0,95 для всех классов. Для сравнения, Mask R-CNN

продемонстрировал точность 0,81 и полноту 0,81 для того же набора данных. С набором данных 2 YOLOv8 достиг точности 0,93 и полноты 0,97. Маска R-CNN в этом одноклассовом сценарии достигла точности 0,85 и полноты 0,88. Кроме того, время сегментации для YOLOv8 составило 10,9 мс для сегментации нескольких классов (набор данных 1) и 7,8 мс для сегментации одного класса (набор данных 2) по сравнению с 15,6 мс и 12,8 мс, достигнутыми с помощью Mask R-CNN, соответственно.

В результате можно сделать выводы, что алгоритмы R-CNN и SSD требуют больше вычислительных ресурсов и времени для анализа и обработки данных. По сравнению с ними алгоритм Yolo имеет несколько преимуществ:

- **Высокая скорость работы:** YOLO способен обрабатывать изображения в реальном времени благодаря однократному анализу всего изображения, в отличие от других алгоритмов, которые обрабатывают определенные регионы изображения.
- **Высокая точность:** Набор данных COCO (Common Objects in Context) [92] — это крупномасштабный набор данных для обнаружения, сегментации объектов. Он предназначен для исследований широкого спектра категорий объектов и обычно используется для сравнительного анализа моделей компьютерного зрения. Это важный набор данных для исследователей и разработчиков, работающих над задачами обнаружения, сегментации и оценки объектов.

В таблице 3.3 показана точность восьмой версии YOLO на COCO с использованием данных, собранных командой Ultralytics [93] и опубликованных в [94]:

Таблица 3.3 - Характеристики моделей YOLOv8

Модель	Размер изображения (px)	mAP (50-95)	Кол-во параметров	FLOPs
YOLOv8n	640	37,3	3,2	8,7
YOLOv8s	640	44,9	11,2	28,6
YOLOv8m	640	50,2	25,9	78,9
YOLOv8l	640	52,9	43,7	165,2
YOLOv8x	640	53,9	68,2	257,8

Архитектура YOLOv8 поддерживает 3 моделей, преобученных по задачам: классификация, детектирование, сегментация (рисунок 3.9). YOLOv8 использует ту же основу, что и YOLOv5, с некоторыми изменениями в CSPLayer, который теперь называется модулем C2f. Модуль C2f сочетает в себе функции высокого уровня с контекстной информацией для повышения точности обнаружения.

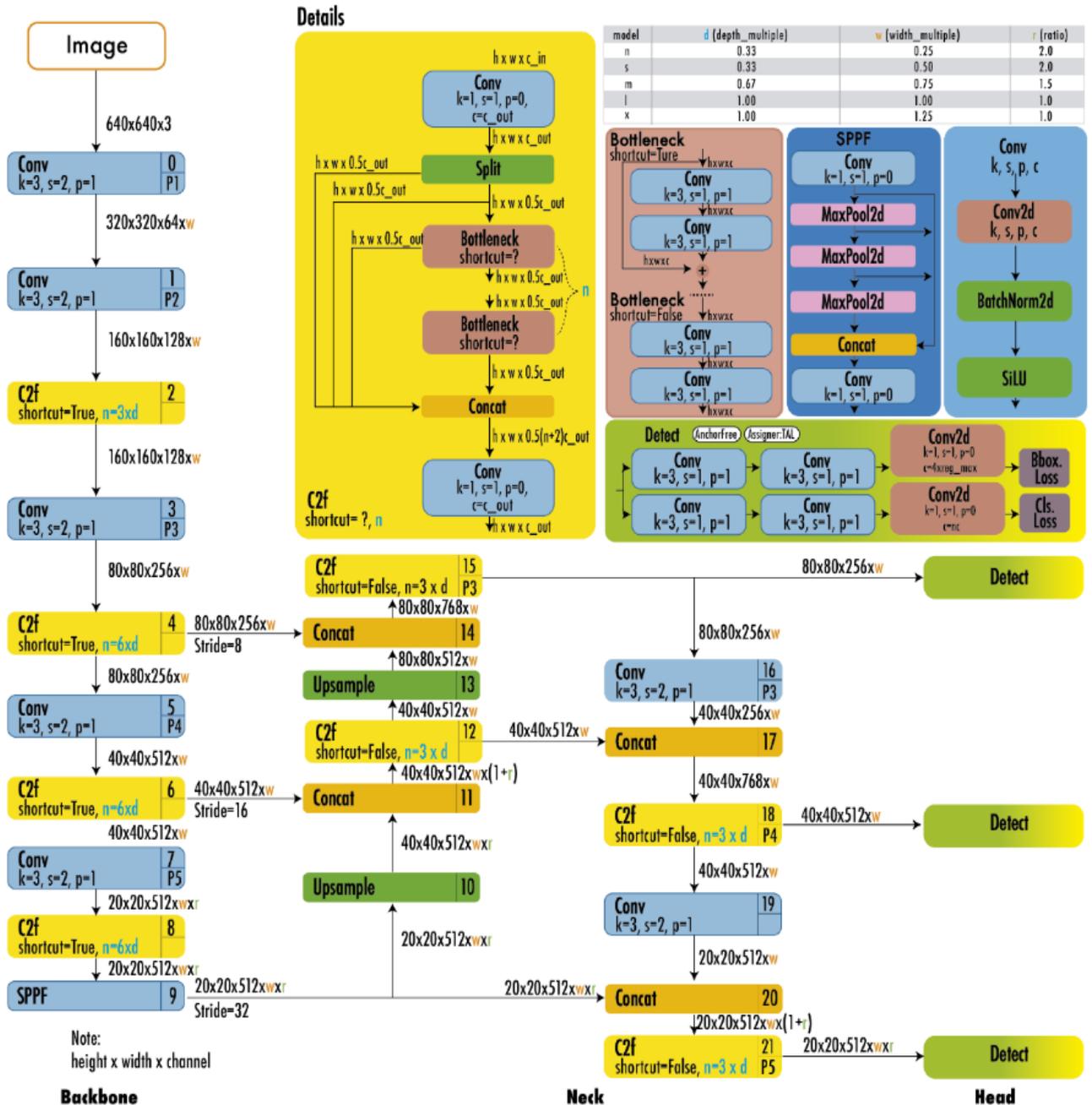


Рисунок 3.9 – Архитектура YOLOv8 [95]

В выходном слое YOLOv8 используется сигмоидальная функция в качестве функции активации для оценки объектности, представляющей вероятность того,

что ограничивающая рамка содержит объект. Для вероятностей классов, представляющую вероятность объектов, принадлежащих каждому возможному классу, используется функция softmax. YOLOv8 использует функции потерь CIoU [96] и DFL [97].

YOLOv8 также предоставляет модель семантической сегментации, называемую моделью YOLOv8-Seg. Основой является экстрактор функций CSPDarknet53, за которым следует модуль C2f вместо традиционной архитектуры YOLO. За модулем C2f следуют две головки сегментации, которые учатся прогнозировать семантические маски сегментации для входного изображения. Модель имеет головки обнаружения, аналогичные YOLOv8, состоящие из пяти модулей обнаружения и уровня прогнозирования. Модель YOLOv8-Seg достигла самых современных результатов в различных тестах обнаружения объектов и семантической сегментации, сохраняя при этом высокую скорость и эффективность [98].

На основании проведенного сравнительного анализа алгоритмов компьютерного зрения, в рамках исследования, будет использоваться YOLOv8 для решения задачи семантической сегментации объектов в пределах рабочей зоны накопителя. Особенностью рассматриваемой технологической задачи является заранее неизвестная ориентация объектов захвата, что требует определения как их координат относительно базовой системы координат робота-манипулятора, так и угловых параметров ориентации в трехмерном пространстве.

Выбор YOLOv8-Seg обоснован его способностью одновременно решать задачи детекции объектов и построения точных масок сегментации, что является необходимым условием для последующего расчета пространственной ориентации методом сопоставления с эталонной 3D-моделью.

#### **3.4.1. Способ автоматического формирования дообучающего набора данных**

Стандартная предобученная модель YOLOv8, обученная на датасете COCO (Common Objects in Context), демонстрирует высокую эффективность при распознавании 80 классов объектов. Этот набор данных не включает

специализированные промышленные объекты, поэтому необходимо дообучение модели на собственном наборе данных.

Структура пользовательского набора данных для обучения YOLOv8 строго регламентирована и включает в себя два основных компонента: изображения и текстовые файлы аннотаций, организованные в определённую структуру.

Файлы аннотаций представляются собой текстовый документ, где каждому изображению соответствует одноимённый текстовый файл со следующим форматом каждой строки в файле: `class_id` – целочисленный идентификатор класса объекта (начинается с 0). `center_x`, `center_y` – координаты центра ограничивающей рамки (`bounding box`) в нормализованных относительно ширины и высоты изображения значениях (диапазон  $[0, 1]$ ). `width`, `height` – ширина и высота ограничивающей рамки в нормализованных значениях (диапазон  $[0, 1]$ ).

Такой формат аннотаций, как показано в работе [99], обеспечивает совместимость с архитектурой YOLOv8 и позволяет эффективно решать задачи сегментации. Для формирования обучающего набора данных необходимо сделать как можно больше фотографий объектов при различных условиях:

- разные ракурсы, съёмка объекта должна быть выполнена с поворотами вокруг всех трех осей (X, Y, Z). Недостаточно менять только угол обзора сверху;
- разное освещение, съёмка при разной интенсивности света — от затемненных условий до яркого, почти засвечивающего освещения. Источники света должны располагаться с разных сторон (спереди, сбоку, сзади, сверху), чтобы создать многообразные паттерны теней и бликов;
- рабочая среда, съёмка должна производиться на том же типе фона, на котором объект будет обнаружен в реальной задаче (например, конвейерная лента, пластиковый поддон, сетка транспортировочного стеллажа).

Ручная сборка набора данных достаточно трудоемкий процесс, т.к. человеку необходимо сделать большое количество фотографий объектов в разных условиях, что занимает большое количество времени. Например, набор данных металлических объектов с соревнований RoboCup содержит 18 классов и 1518 фотографий [100] и на сборку такого набора потребуется около 5 рабочих дней, при

затрате на одну фотографию около 2 минут с учетом подготовки и настройки оборудования, смены фона, освещения, углов съемки, перерывов, усталость, а также отбраковки некачественных снимков.

Для сокращения времени, затрачиваемого на сборку набора данных и повышения его качества, разработан способ автоматического формирования дообучающего набора данных, позволяющий автоматически в циклическом режиме получать фотографии объектов, с помощью поворота камеры, закрепленной на фланце. Программа управления манипулятором реализует обход рабочей зоны с шагом задаваемым оператором и получение фотографии в точках остановки.

Алгоритм прохода рабочей зоны роботом Kuka KR4 R600 приведен на рис.3.10.

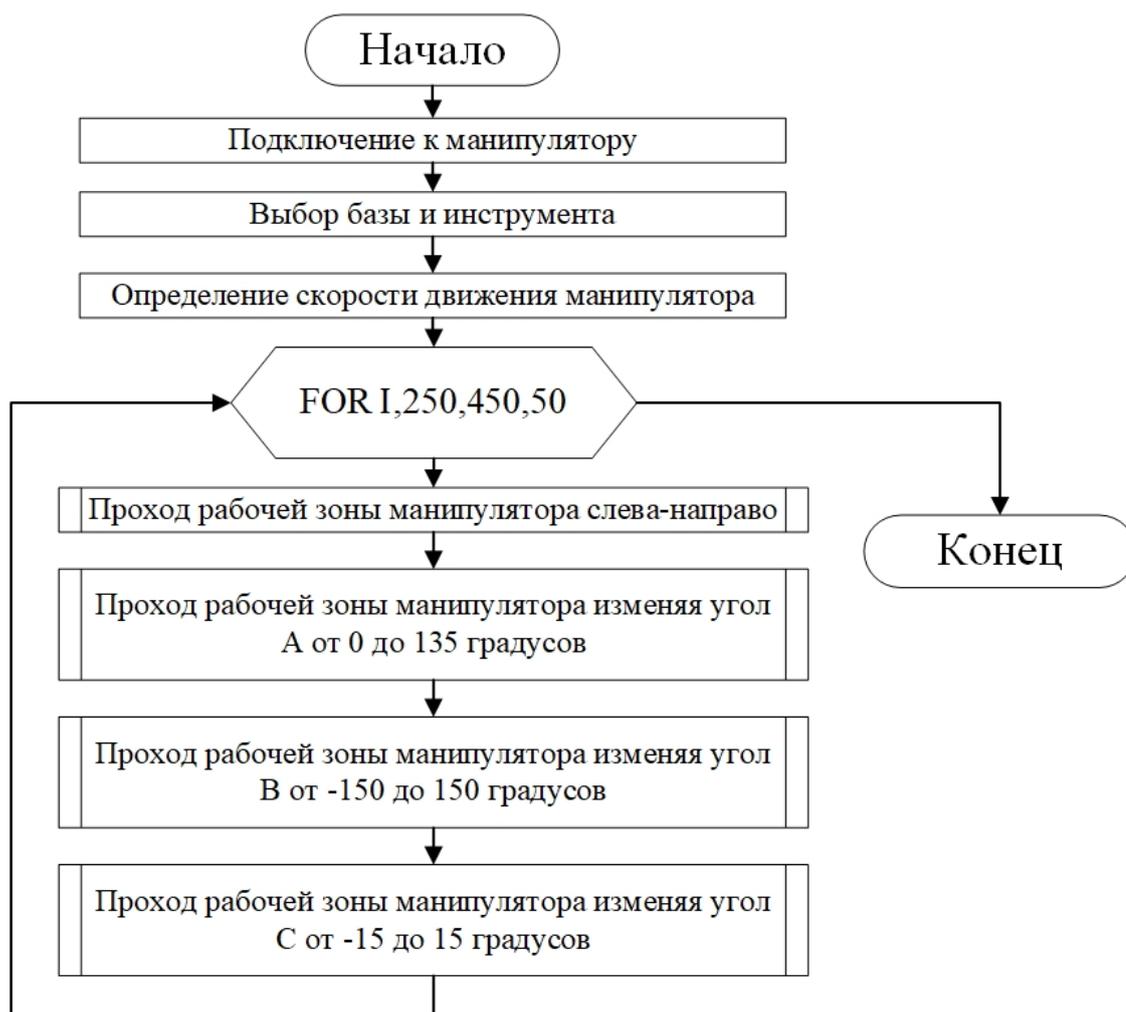


Рисунок 3.10 – Алгоритм роботизированной сборки дообучающего набора данных Kuka KR4 R600

Алгоритм представленный на рис.3.10 предполагает использование системы дистанционного управления, представленного в главе 2 диссертации и написан с использованием языка программирования Python. В результате выполнения алгоритма робот в автоматическом режиме проходит рабочую зону с шагом, который задается в зависимости от размера объектов, которые будут использоваться в наборе данных. Чем больше размер объектов, тем больше будет шаг, т.к. при малом изменении шага, изображение не будет подвергаться сильному изменению. В процессе прохождения рабочей зоны роботом изменяются углы по трем осям: угол А изменяется в диапазоне от 45 до 135 градусов, угол В изменяется в диапазоне от -150 до 150 градусов, т.к. при изменении угла в большем диапазоне может произойти блокировка робота, угол С изменяется в диапазоне от -15 до 15 градусов, т.к. при большем изменении угла происходит блокировка робота.

В результате выполнения алгоритма с шагом 100 миллиметров по оси X, с шагом 50 миллиметров по оси Y и с шагом 100 миллиметров по оси Z камера делает 336 фотографий размером 640x640 пикселей. Время выполнения при скорости 50% (1 м/с) равняется 10 минут 31 секунду. Большую скорость не рекомендуется использовать, т.к. камера не фокусируется на объекте. Пример полученных фотографий, снятых камерой Intel RealSense D415i, закрепленной на роботе, приведен на рис.3.11.



Рисунок 3.11 – Пример фотографий, собранных предложенным способом

После сбора изображений выполняется их аннотирование посредством ручного выделения целевых объектов с последующим присвоением соответствующих классов. Полученный таким образом набор данных разделяется на обучающую, валидационную и тестовую выборки в стандартных пропорциях.

Для повышения репрезентативности набора данных применяются методы аугментации (повороты, наложение размытия и шума на изображение).

### 3.4.2. Выбор объекта для безопасного захвата манипулятором

В процессе сегментации на видео потоке выделяются все объекты, которые распознаны. Пример приведен на рис.3.12.



Рисунок 3.12 – Выделение объектов на видео потоке

В случае, приведенном на рисунке 3.12 объект №1 не перекрыт и не перекрывает другие объекты, объект №2 не перекрыт другими объектами, но при этом он закрывает объект №3, который в свою очередь перекрывает объект №4.

При таком расположении объектов роботу-манипулятору необходимо выбрать тот объект, захватив который он не затронет другие. В приведенном случае объектом к захвату может быть объект №1 или объект №2, но объект №1 находится дальше по оси  $Z$  относительно системы координат камеры, чем объект №2. При попытке его захвата может произойти столкновение с другими объектами, что нежелательно, т.к. может повредить захватное устройство. Пример возможного столкновения с другими объектами при захвате объекта приведен на рис. 3.13.



Рисунок 3.13 – Пример захвата объекта №1

На основании этого можно сделать вывод, что захватывать необходимо объекты, которые находятся ближе всего к фланцу робота по вертикальной оси. Для определения расстояния до центров распознанных объектов по вертикальной оси используется RGB-D камера IntelRealSense D415i и библиотека pyrealsense2 [101]. На рис.3.14 произведена сегментация объектов, определение их центра (отмечены зеленой точкой) и расстояния по вертикальной оси.

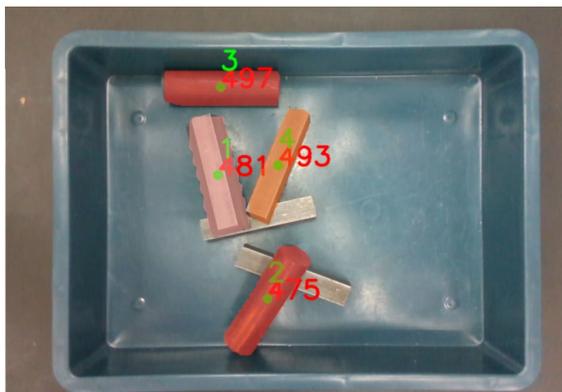


Рисунок 3.14 - Сегментация объектов с помощью RGB-D камеры IntelRealSense D415i

Согласно рисунку 3.14 для захвата будет выбран объект №2, т.к. расстояние до него по вертикальной оси минимальное.

### 3.5. Методика определения объекта возле стенок накопителя

В работах [102] описаны основные причины неудачного схвата объектов промышленным манипулятором в процессе сортировки при их расположении в накопителе:

- Изменениями окружающей среды, которые происходят после оценки пространственных координат объекта. В случае использования пальцевого захвата, когда он приближается к объекту, его пальцы часто соприкасаются с окружающими объектами.
- Расположение предметов возле стенок накопителя.

Автором [103] в процессе экспериментальных исследований установлено, что решением проблемы, связанной с затрагиванием других объектов в процессе захвата, является использование магнитного захвата, который захватывает объекты

сверху и не соприкасается с другими, тем самым не происходит изменения расположения заготовок в накопителе.

Для решения проблемы, связанной с расположением предметов возле стенок накопителя, используется дополнительно дообученная ИНС YOLOv8 производящая определение границ накопителя и алгоритм, сравнивающий координаты ограничивающей рамки объекта и полученные значения границ накопителя. При приблизительном совпадении значений крайней координаты ограничивающей рамки объекта и координат стенок накопителя, манипулятор смещается от стенки. Общий алгоритм способа определения расположения объекта возле стенок накопителя, для магнитного захвата приведен на рис.3.15.

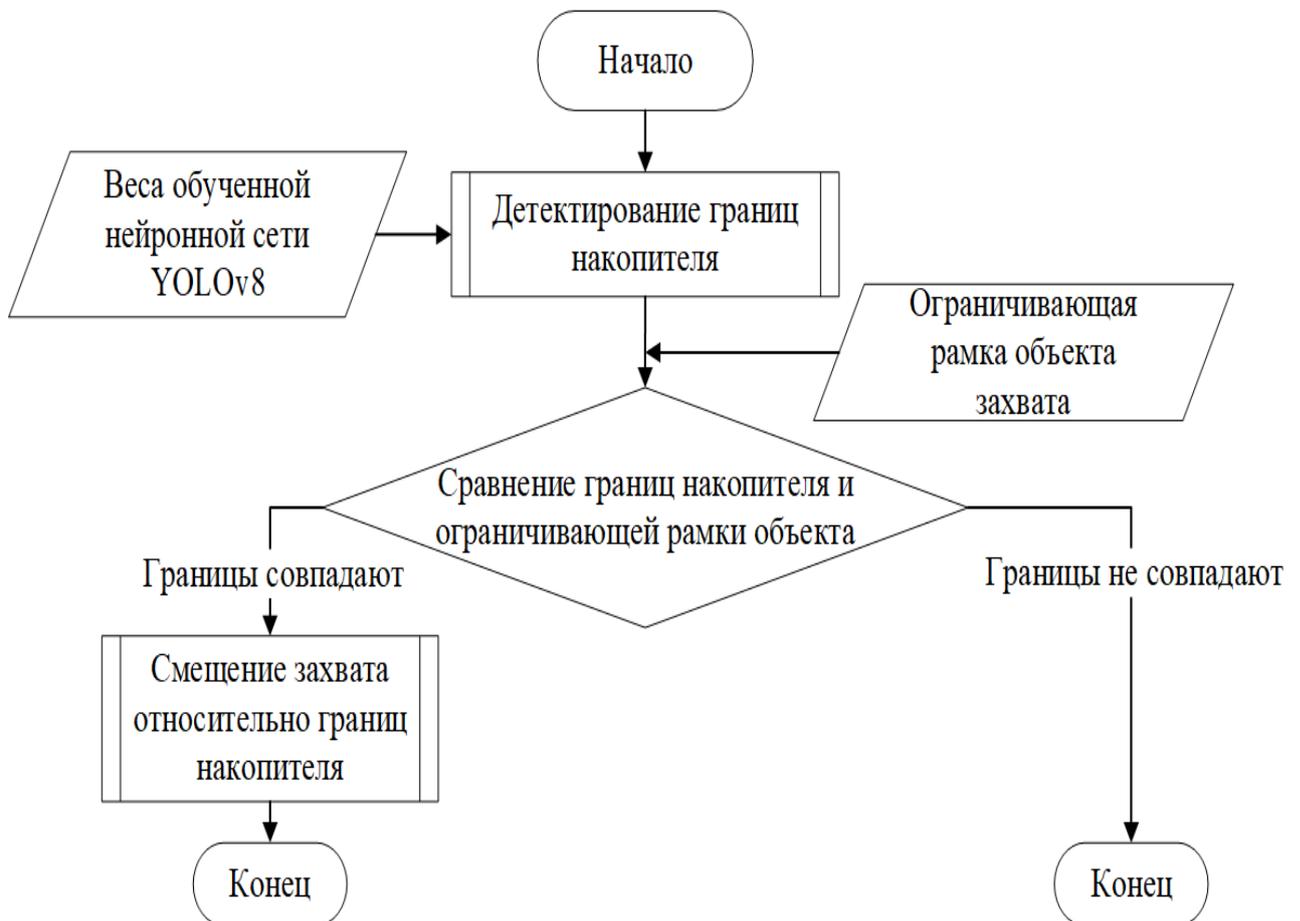


Рисунок 3.15 - Алгоритм методики определения расположения объекта возле стенок накопителя

При использовании алгоритма, приведенного на рис.3.15, происходит захват объекта со смещением относительно его центра, что необходимо учитывать и использовать захватное устройство с достаточным удерживающим усилием.

### 3.6. Алгоритм определения пространственных координат объекта

После определения объекта для захвата необходимо рассчитать его пространственные координаты согласно алгоритму, состоящего из двух подэтапов: определение координат центра объекта захвата и определение углов наклона объекта.

Для определения координат центра объекта захвата, в рамках диссертационного исследования проанализированы два способа:

- Перевод полученных значения центра объекта из пикселей в миллиметры, с помощью вычисления соотношения реальных размеров рабочей области в миллиметрах и количества пикселей, по каждой из осей, которые ей соответствуют.

- Определение насколько сдвинут центр объекта относительно центра фланца робота через фокусное расстояние камеры и расстояние до объекта по вертикальной оси.

Для реализации первого способа необходимо откалибровать камеру. В рамках работы для этого используется алгоритм `findChessboardCorners`, который предназначен для автоматического обнаружения внутренних углов шахматного паттерна на изображении и позволяет определить внутренние параметры (фокусное расстояние, оптические искажения) и внешнюю ориентацию камеры [104].

Пример расположения шахматной доски в рабочей зоне робота приведен на рис.3.16.

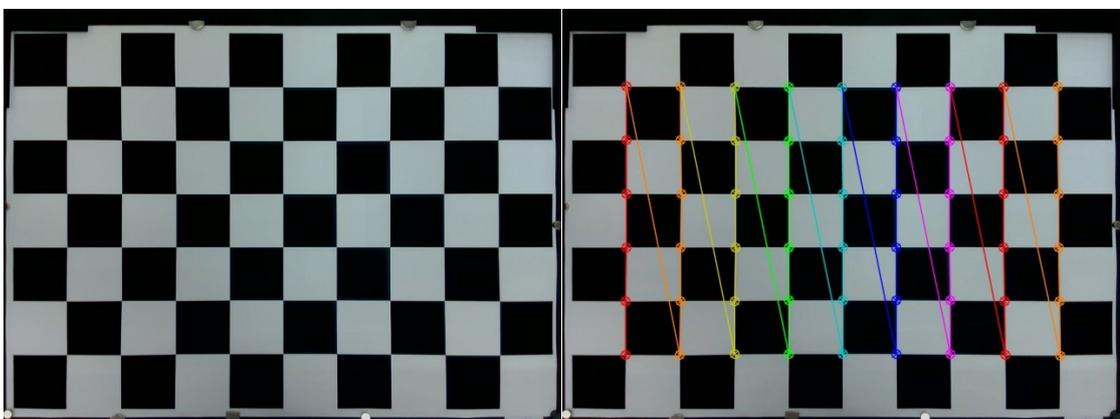


Рисунок 0.16 – Расположение шахматной доски и калибровка камеры

В результате выполнения процедуры калибровки камеры определяются коэффициенты дисторсии, количественно описывающие отклонения реальной оптической системы от идеальной модели точечной проекции. При этом анализируются два типа искажений: радиальная дисторсия (возникает из-за неидеальности формы линзы) и тангенциальная дисторсия (обусловлена несовпадением оптического центра с геометрическим) [105]. Калибровка камеры позволяет убрать искажения, особенно на краях рабочей зоны, что позволяет снизить погрешность, возникающую при пересчете координат центра объекта из пикселей в миллиметры.

Также помимо исправления искажений объекта алгоритм `findChessboardCorners` позволяет определить размер шахматной клетки в пикселях, соотнеся который с реальным размером клетки шахматной доски в миллиметрах, можно в автоматическом режиме найти ширину и длину рабочей зоны камеры в миллиметрах и обрезать получаемый видеопоток согласно расположенной шахматной доске.

Далее на основании полученного соотношения, согласно формулам 3.1 и 3.2 рассчитываются координаты центра объекта в миллиметрах:

$$cX_{mm} = \frac{A}{W} * cX, \quad (0.1)$$

$$cY_{mm} = \frac{B}{H} * (H - cY), \quad (0.2)$$

где  $cX$  и  $cY$  – координаты центра объекта в пикселях,  $W$  – размеры получаемого изображения с камеры по длине, с учетом что полученное изображение обрезаются по краям шахматной доски,  $H$  – размеры получаемого изображения с камеры по высоте, с учетом что полученное изображение обрезаются по краям шахматной доски,  $A$  – размер рабочего поля по Оси  $X$  в мм,  $B$  – натуральный размер рабочего поля по Оси  $Y$  в мм.

В `OpenCV` начальная точка с координатами  $(0,0)$  находится в левом верхнем углу, в связи с этим в формуле 3.2 во втором слагаемом необходимо вычесть общее число пикселей по высоте и координаты центра распознанного объекта.

Для корректного переведение пикселей в миллиметры необходимо откалибровать базу робота, так чтобы точка с координатами (0,0) робота и камеры совпадали.

Общий алгоритм, позволяющий перевести значения центра объекта из пикселей в миллиметры приведен на рисунке 3.17.

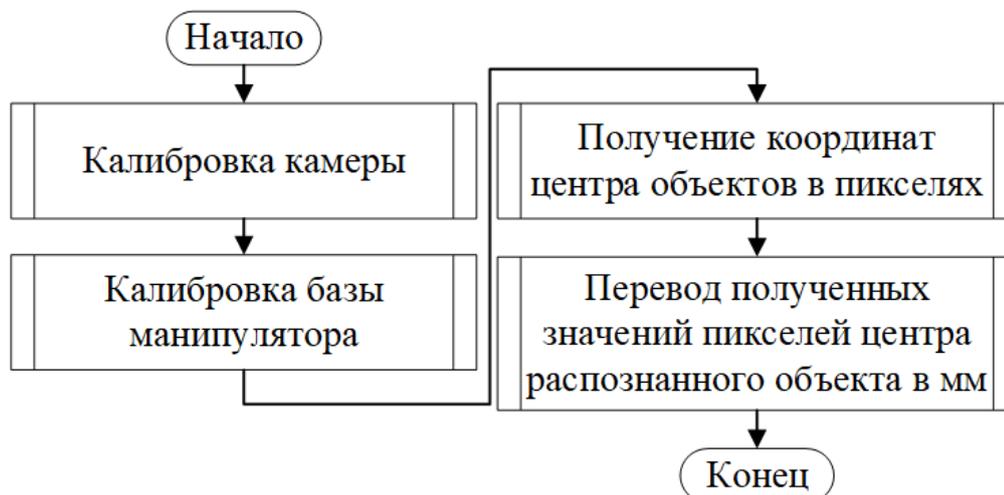


Рисунок 3.17 – Алгоритм способа перевода пикселей в миллиметры

На основании алгоритма, приведенного на рис. 3.17, создан модуль на языке Python, который позволяет автоматически определять пересечения клеток шахматной доски и выводить их координаты в пикселях и используя формулы 3.1 и 3.2 производится пересчет значений пикселей в реальные размеры в миллиметрах.

В связи с закреплением камеры Intel RealSense D415i, RGB линза которой смещена относительно центра фланца робота в формулы необходимо добавить смещение:

$$cXmm_{center} = cXmm - 31.9, \quad (3.3)$$

$$cYmm_{center} = cYmm - 64.6, \quad (3.4)$$

Самой большой проблемой данного способа является большое количество настроек, таких как калибровка базы робота, установка шахматной доски при первичной настройке, ограничение рабочей зоны камеры для соотнесения нулевой точки. При этом корректно способ будет работать только на одном исходном

расстоянии до объекта по Оси  $Z$ , т.к. при другом расстоянии соотношение, представленное в формулах 3.1 и 3.2 изменится.

Существует другой способ предполагает не определения координат центра объекта в миллиметрах, а определение того насколько сдвинут центр объекта относительно центра фланца робота через фокусное расстояние камеры и расстояние до объекта по вертикальной оси.

Основные соотношения, используемые для этого способа представлены в пункте 1.4 диссертации и основываются на исследовании [61]:

$$X(mm) = \frac{D_{x,y}(C_x - x_0(\text{пиксели}))}{f_x}, \quad (3.5)$$

$$Y(mm) = \frac{D_{x,y}(C_y - y_0(\text{пиксели}))}{f_y}, \quad (3.6)$$

где  $D_{x,y}$  — расстояние до центра объекта по Оси  $Z$  (в миллиметрах),  $C_x$  — координата центра изображения по оси  $X$  (в пикселях),  $C_y$  — координата центра изображения по оси  $Y$  (в пикселях),  $f_x, f_y$  — внутренние параметры оптической системы камеры, используемой для получения изображения (фокусные расстояния по осям  $X$  и  $Y$ ),  $(x_0, y_0)$  — координата центра обнаруженного объекта на изображении в пикселях.

Для использования второго способа необходимо знать расстояние до объекта по вертикальной оси, вычисления которого осуществляется с помощью RGB-D камеры Intel RealSense D415i. Размер изображения, получаемого с камеры, равняется 640x480 пикселей и фокусное расстояние, для данного разрешения, получаемое с помощью функции `get_intrinsics` библиотеки `pyrealsense2` [106] равняется  $f_x = 604.602, f_y = 604.162$ .

Для определения координат центра распознанного объекта в миллиметрах необходимо из текущих координат робота по Оси  $X$  вычесть полученные значения смещения, а по Оси  $Y$  прибавить, т.к. Ось  $Y$  в библиотеки OpenCV начинается в левом верхнем углу.

Главным минусом второго способа является использование датчика, позволяющего вычислять расстояние до объекта по вертикальной оси, но, в рамках

диссертационной работы предполагалось использование подобного датчика, для определения расстояния до объекта и захвата его роботом.

Способ для перевода пикселей центра объекта в миллиметры должен быть универсальным для любой высоты до объекта и не требовать дополнительной первоначальной настройки. В связи с этим выбран второй способ, но при его использовании объекты, которые находятся под центром фланца робота имеют нулевую погрешность, а при удалении от центра погрешность возрастает. К аналогичному выводу пришли исследователи в [106] ошибка у которых получилась в среднем около 7–12 миллиметров и увеличивалась по мере удаления объектов от камеры, при использовании Intel RealSense D415i.

Для минимизации погрешности, возникающей при удалении объекта от центра предложена модификация способа, которая предполагает сегментацию объекта два раза в процессе захвата. На первой итерации происходит сегментация всех объектов, выделение объекта для захвата, с помощью способа, приведённого в пункте 3.3.2, и перевод координат его центра из пикселей в миллиметры, через фокусное расстояние камеры, далее робот перемещается над объектом, и камера производит второй раз процесс сегментации. В таком случае удастся избежать погрешностей, т.к. объект находится всегда практически в центре кадра.

Для минимизации ложных срабатываний, возникающих на второй итерации сегментации, в видеопоток вводится бинарная маска черного цвета. Данная маска применяется ко всем кадрам, получаемым с видеокамеры, за исключением области, соответствующей объекту, выделенному в результате первичной сегментации. Это позволяет ограничивать обработку изображения только релевантной областью, снижая уровень шумов и артефактов на последующих этапах анализа.

Ограничивающая маска создается по следующим формулам:

$$x = \frac{points[2]-points[0]}{2}, \quad (3.7)$$

$$y = \frac{points[3]-points[1]}{2}, \quad (3.8)$$

где  $points [0]$ ,  $points [2]$  – значения крайних точек ограничивающей рамки, получаемой в результате сегментации, по Оси X,  $points [1]$ ,  $points [3]$  – значения

крайних точек ограничивающей рамки, получаемой в результате сегментации, по Оси  $Y$ .

После первой итерации сегментации робот перемещается и зависает над объектом, следовательно для создания маски можно использовать следующую формулу:

$$mask = (C_y - y : C_y + y, C_x - x : C_x + x), \quad (3.9)$$

где  $C_x$ — координата центра изображения по оси  $X$  (в пикселях),  $C_y$ — координата центра изображения по оси  $Y$  (в пикселях). Для разрешения 640x480 пикселей  $C_x = 320, C_y = 240$ . Порядок выделения объекта на видео потоке представлен на рис. 3.18.



Рисунок 3.18 – Порядок выделения объекта на видео потоке

На втором подэтапе решается основная проблема точного определения ориентации объекта захвата. Для этого могут использоваться несколько методов, разделенных по типам оценки:

- Методы, основанные на соответствии между данными, получаемыми с камеры, чаще всего которыми являются облака точек, в случае использования RGB-D сенсоров и 3D моделью существующего объекта.
- Методы, основанные на шаблонах, которые ищут места на изображении, наиболее похожие на него.
- Методы, основанные на нейронных сетях.

Методы, основанные на шаблонах, могут использоваться при работе с объектами, имеющими неоднородные участки. Например углы, так как там происходит резкое изменение интенсивности в двух разных направлениях. Однако они не могут, с высокой точностью, работать с объектами, которые перекрываются

между объектами, потому что, если объект закрыт, то оценка подобия шаблона будет низкой [107].

При использовании RGB изображений, в настоящее время, применяют машинное обучение для оценки ориентации в 6D формате [108]. В таком случае в качестве шаблона выступает нейросетевой классификатор, который находит объект на видеопотоке и производит его сегментацию. Примером таких нейросетевых классификаторов могут быть PoseCNN [109], SSD6D [110], Deep6DPose [111]. Главной проблемой методов, использующих нейронные сети для определения углов наклона объекта в пространстве, является большой объем обучающих данных и большое количество затрачиваемого времени на обучение сети. Одним из подобных наборов данных является DIMO (Dataset of Industrial Metal Objects) [112], который содержит 31 200 изображений 600 реальных сцен и 553 800 изображений 42 600 синтетических сцен, разделенных на шесть классов металлических объектов: квадратная, прямоугольная и круглая заготовка, а также трубка круглого сечения, квадратная и круглая емкость. При этом на его сборку авторы затратили 6 месяцев при съемке с трех видов стереокамер.

В большинстве работ определение ориентации объекта использует RGB-D камеру, обзор методов приведен в [113-114]. Информация о расстоянии до объекта по вертикальной оси позволяет устранить неоднозначность масштаба объекта, чего нельзя добиться, используя монокулярное зрение, но для работы методов, предполагающих использование RGB-D камеры, необходимо наличие 3D модели объекта.

В случае применения методов, с использованием RGB-D камеры, в промышленной робототехнике часто 3D-модель объекта существует еще до создания реального объекта или модель объекта достаточно простая для ее 3D моделирования, а использование RGB-D датчика предполагается изначально, т.к. необходимо для вычисления расстояния по вертикальной оси до захватного устройства робота-манипулятора.

Разрабатываемый метод должен иметь возможность перенастройки на различные виды выпускаемой продукции, поэтому в рамках работы для

определения углов наклона объектов используется метод, основанный на сравнении координат 3D модели объекта, сформированной в системе координат с началом в точке  $[0,0,0]$  и множеством координат облака точек реального объекта с системой координат начало которой находится в центре объекта. Алгоритм определения ориентации объекта захвата приведен на рис.3.19.

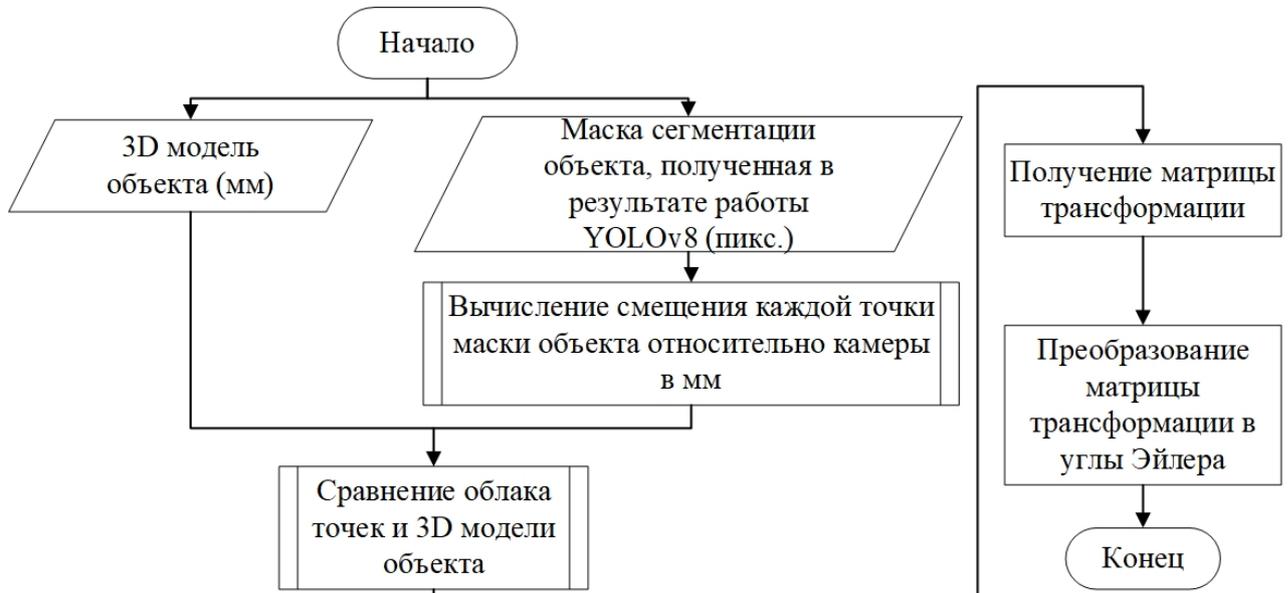


Рисунок 3.19 – Алгоритм вычисления ориентации объекта захвата

Полученная, в результате сегментации с помощью нейросетевого классификатора YOLOv8, маска объекта в пикселях передается в функцию, рассчитывающую смещение каждой точки маски относительно камеры по трем осям в миллиметрах. Полученные значения сравниваются с 3D моделью объекта, сформированной в САПР системе (Blender), с помощью аффинных преобразований.

Стандартным форматом 3D моделей для систем быстрого прототипирования и производства является .stl [115], но для соотношения облака точек с исходной 3D моделью необходимо использовать формат .ply, который позволяет описывать объект в виде списка плоских полигонов, которые образуют в свою очередь полигональную сетку. Для упрощения процесса построения модели в формате .ply можно использовать среду разработки Blender, в которую импортируется исходная модель в формате .stl. Количество вершин полигональной сетки, при

импортировании, может быть недостаточно для полноценного облака точек, пример показан на рис 3.20.

Для решения этой проблемы применяется подход создания новой топологии сетки (Remesh) с помощью соответствующего модификатора в редакторе трехмерной графики Blender. В результате работы данного инструмента новые вершины и ребра повторяют кривизну исходного объекта в виде правильных четырехугольников.

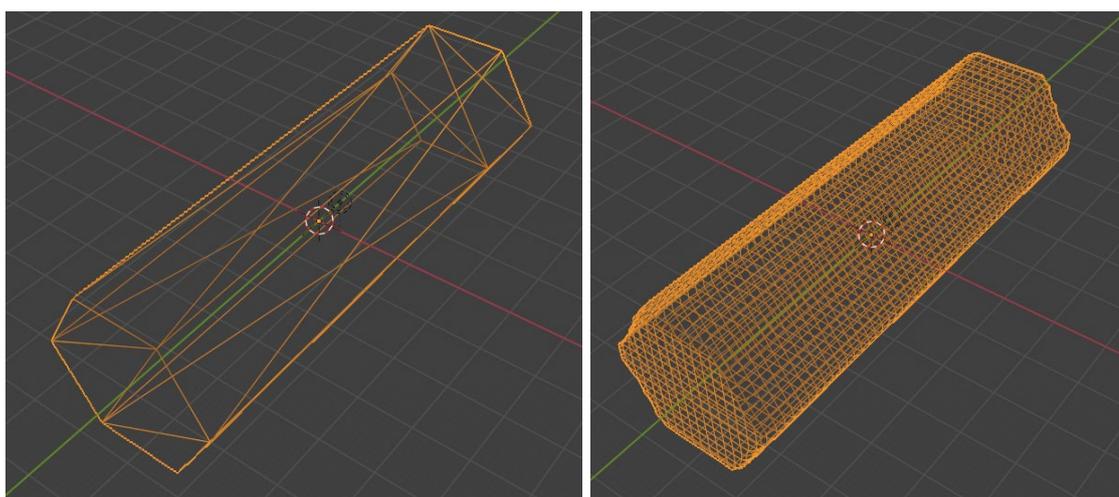


Рисунок 3.20 – Оригинальная модель детали в режим отображения ребер и исправленная с помощью новой топологии сетки

Сопоставление детали выполняется с помощью аффинных преобразований, которые включают в себя поворот и перемещение (параллельный перенос) детали к облаку точек [116].

В общем виде аффинное преобразование точки в трехмерном пространстве можно описать следующим выражением (3.10) **Ошибка! Источник ссылки не найден.:**

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix}, \quad (3.10)$$

где  $x'$ ,  $y'$ ,  $z'$  – координаты после преобразований, матрица  $R$  – матрица поворота,  $x$ ,  $y$ ,  $z$  – исходные координаты,  $t$  – вектор переноса.

В компьютерной графике аффинное преобразование выражается матрицей 4 на 4, что позволяет представить выражение (3.10) **Ошибка! Источник ссылки не найден.** в виде следующей формулы (3.11):

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} R_{11} & R_{12} & R_{13} & t_x \\ R_{21} & R_{22} & R_{23} & t_y \\ R_{31} & R_{32} & R_{33} & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}, \quad (3.11)$$

Матрица 4 на 4 в выражении (3.11) **Ошибка! Источник ссылки не найден.** является частным случаем матрицы преобразований, где нули в последней строке указывают на аффинность.

Для параллельного переноса матрица аффинного преобразования будет иметь следующий вид:

$$\begin{pmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad (3.12)$$

Для поворота вокруг оси Y, где координаты Y не меняются, данная матрица будет иметь следующий вид:

$$\begin{pmatrix} \cos(-\phi_y) & 0 & -\sin(-\phi_y) & t_x \\ 0 & 1 & 0 & t_y \\ \sin(-\phi_y) & 0 & \cos(-\phi_y) & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad (3.13)$$

Для поворота вокруг оси X, где координаты X не меняются матрица аффинного преобразования, будет иметь следующий вид:

$$\begin{pmatrix} 1 & 0 & 0 & t_x \\ 0 & \cos(\phi_x) & -\sin(\phi_x) & t_y \\ 0 & \sin(\phi_x) & \cos(\phi_x) & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad (3.14)$$

Для поворота вокруг оси Z, где координаты Z не меняются матрица аффинного преобразования, будет иметь следующий вид [117]:

$$\begin{pmatrix} \cos(\phi_z) & -\sin(\phi_z) & 0 & t_x \\ \sin(\phi_z) & \cos(\phi_z) & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad (3.15)$$

В работе решается задача, наложения САПР модели, созданной в формате плоских полигонов, и облака точек, то есть выравнивания двух частично перекрывающихся облаков точек таким образом, чтобы минимизировать расстояния между ними. Решение данной задачи затруднительно, поскольку неизвестна матрица преобразований трехмерной детали к облаку точек. Для нахождения этой матрицы принято решение дополнительно использовать алгоритм ICP (Iterative Closest Point) [118].

Алгоритм ICP является итеративным алгоритмом и состоит из 4-х шагов.

На первом шаге находятся ближайшие пары точек используя выражения (3.16) и (3.17):

$$S(\tau) = \|\tau_j - s_i\|_2, \quad (3.16)$$

$$(\tau) = \underset{s_j \in S, \tau_j \in T}{\operatorname{argmin}} S(\tau), \quad (3.17)$$

где  $S(\tau)$  – расстояние между ближайшими точками,  $\tau_j$  – точка из облака  $T$ , ближайшая к точке  $s_i$  из другого облака точек,  $\|\cdot\|_2$  – евклидова норма или евклидово расстояние.

На следующем шаге вычисляется так называемая матрица поворота в соответствии со следующими выражениями:

$$J(R, t) = \sum_{i=1}^N \|(Rs_i + t) - \tau_j\|_2^2, \quad (3.18)$$

$$(R, t) = \underset{R \in SO(3), t \in \mathbb{R}^3}{\operatorname{argmin}} J(R, t), \quad (3.19)$$

где  $R$  – матрица поворота,  $t$  – вектор трансформации,  $\tau_j$  – точка из облака точек, ближайшая к точке  $s_i$  из другого облака точек,  $SO(3)$  – ортогональная группа вращений,  $\|\cdot\|_2^2$  – квадрат евклидовой нормы.

На третьем шаге облако точек трансформируется на основании полученной матрицы поворота:

$$s_i = Rs_i + t, \quad (3.20)$$

где  $s_i$  – трансформируемая точка,  $Rs_i$  – матрица поворота,  $t$  – вектор трансформации.

Четвертый шаг предполагает повторение всех предыдущих этапов пока  $J(R,t) \geq \varepsilon$ , где  $\varepsilon$  – некоторый показатель сходимости, только при этом используется уже трансформированное облако точек.

Полученную матрицу поворота необходимо преобразовать в углы Эйлера, согласно формулам (3.21-3.23) [119]:

$$\alpha = \arctan\left(\frac{-R_{12}}{R_{11}}\right), \quad (3.21)$$

$$\beta = \arcsin(R_{13}), \quad (3.22)$$

$$\gamma = \arctan\left(\frac{-R_{23}}{R_{33}}\right), \quad (3.23)$$

где  $\alpha$  – угол поворота относительно Оси Z,  $\beta$  – угол поворота относительно Оси X,  $\gamma$  – угол поворота относительно Оси Y.

При анализе результатов экспериментов определено, что маска сегментации, получаемая в результате работы YOLOv8, выделяет не только точки, принадлежащие объекту, но и захватывает точки, которые находятся за его пределами, что является дополнительной проблемой и вносит значительную погрешность в результаты расчета углов наклона объекта относительно камеры, закрепленной на фланце, пример приведен на рис.3.21.



Рисунок 3.21 – Пример маски сегментации, при распознавании под углом

Для минимизации погрешности определения ориентации объекта необходимо производить расчет под прямым углом к маске сегментации, добиться чего возможно повернув фланец, предварительно определив поворот объекта по Оси Z. Для определения угла поворота используется метод FindContours,

позволяющий выделять контур объекта на двухбитном изображении, получаемом в результате сегментации.

Из полученного набора 2D точек определяется прямоугольник с минимальной площадью, охватывающий весь контур, а также угол поворота, который всегда лежит в пределах от  $-90$  до  $0$ , т.к. это угол между линией, соединяющей начальную (точка 0) и конечную точку (точка 3) и осью X. Четыре угловые точки располагаются по часовой стрелке, начиная с точки с наибольшим значением Y. Если две точки имеют одинаковую максимальную величину Y, то начальной точкой является самая правая точка. При этом в каждом квадранте начальная и конечная точка изменяются, как на рис.3.22.

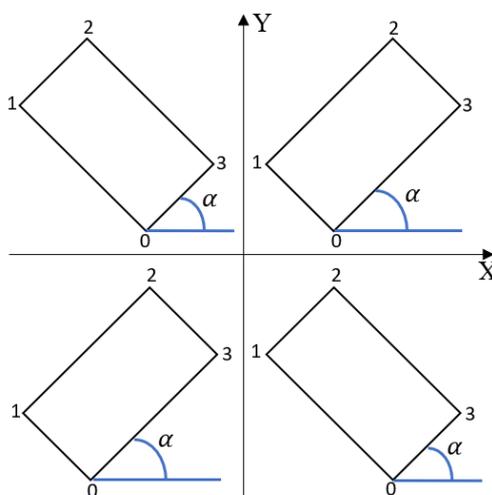


Рисунок 3.22 – Расчет угла относительно нормали

Чтобы рассчитывать угол положения камеры не от  $-90$  до  $0$ , а от  $0$  до  $180$  необходимо учитывать ширину и длину полученной рамки. Если ширина больше длины, то угол рассчитывается согласно следующей формуле:

$$\alpha = \text{angle}A - \text{angle}_{\text{contour}}, \quad (3.24)$$

где  $\text{angle}A$  – текущий угол поворота, фланца робота относительно Оси Z, его можно получить используя систему дистанционного управления и функцию `read_cartessian()`,  $\text{angle}_{\text{contour}}$  – рассчитанный угол, с помощью функции `cv2.minAreaRect()`.

Если длина, полученной рамки, больше ширины, то угол рассчитывается согласно следующей формуле:

$$\alpha = 90 + (\text{angle}A - \text{angle}_{\text{contour}}). \quad (3.25)$$

Применение алгоритма вычисления ориентации объекта захвата (рисунок 3.19), требует предварительного расчета угла поворота камеры относительно маски сегментации по методике расчета угла наклона маски сегментации. При этом RGB линза в стереокамере смещена на -32 мм по Оси X и на 65 мм по Оси Y, поэтому при повороте камеры относительно центра фланца робота, происходит смещение оптического центра камеры за границу размеров первоначально рассчитанной двухбитной маски, накладываемой на видео поток. В этой связи выполнена предварительная калибровка захвата, обеспечивающая совпадение точки отсчета центра фланца с оптическим центром линзы. В результате зона видимости камеры остается в пределах маски на протяжении всей операции позиционирования. В этом случае достигается перпендикулярное расположение камеры относительно плоскости объекта исчезают геометрические искажения и падение освещенности к краям кадра, что практически полностью устраняет шумы и ложные срабатывания алгоритма сегментации. Пример корректного «восприятия» камерой объекта расположения объекта в кадре приведен на рисунке 3.23.

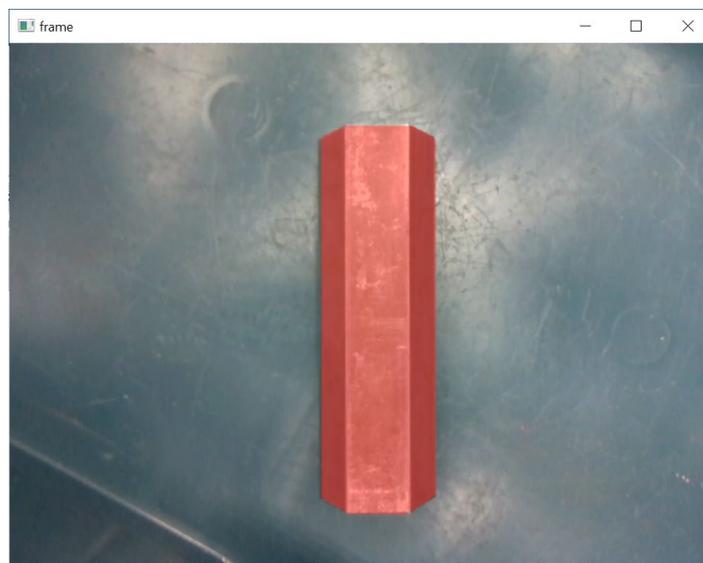


Рисунок 3.23 – Маска сегментации объекта, при расположении камеры под прямым углом к ней

Для верификации алгоритма, представленного на рис. 3.19, проведено тестирование в условиях, когда камера расположена под прямым углом

относительно маски сегментации. На первом этапе нейросетевая модель осуществляет детекцию объекта, находящегося в пределах накопителя. Затем с применением метода FindContours на двухбитной маске производится определение угла ориентации объекта в плоскости. Далее манипулятор, с закрепленной камерой, выполняет поворот на вычисленный угол, после чего осуществляется повторная итерация распознавания. Для полученной маски сегментации генерируется облако точек, на основе которого, в соответствии с алгоритмом ICP (Iterative Closest Point), вычисляется матрица поворота между 3D-моделью и облаком точек.

Результаты наложения продемонстрированы на рис. 3.24 для шестигранной ферромагнитной заготовке, размещенной на дне накопителя, а также на регулируемой подставке, обеспечивающей угол наклона  $-27^\circ$ , по одной из осей, относительно системы координат камеры.

В первом случае объект расположен параллельно основанию и углы наклона относительно камеры равняются  $\alpha = 0$ ,  $\beta = -1.15$ , т.к. основание немного изогнуто,  $\gamma = 0$ . Рассчитанные значения углов, приведенным выше способом, равняются  $\alpha = 0$ ,  $\beta = -1.11490694$ ,  $\gamma = 2.92540631$ .

Во втором случае объект расположен на подставке и находится под углом  $\gamma = -27.05$  по отношению к камере, углы  $\alpha = 0$ ,  $\beta = 1.7$ , связано это с неровностью основания. Рассчитанные значения углов, приведенным выше способом, равняются  $\alpha = 0$ ,  $\beta = 2.46865401$ ,  $\gamma = -28.221254919671175$ .

Погрешность при вычислении углов присутствует, но ее можно нивелировать, используя магнитное захватное устройство с достаточной удерживающей силой.

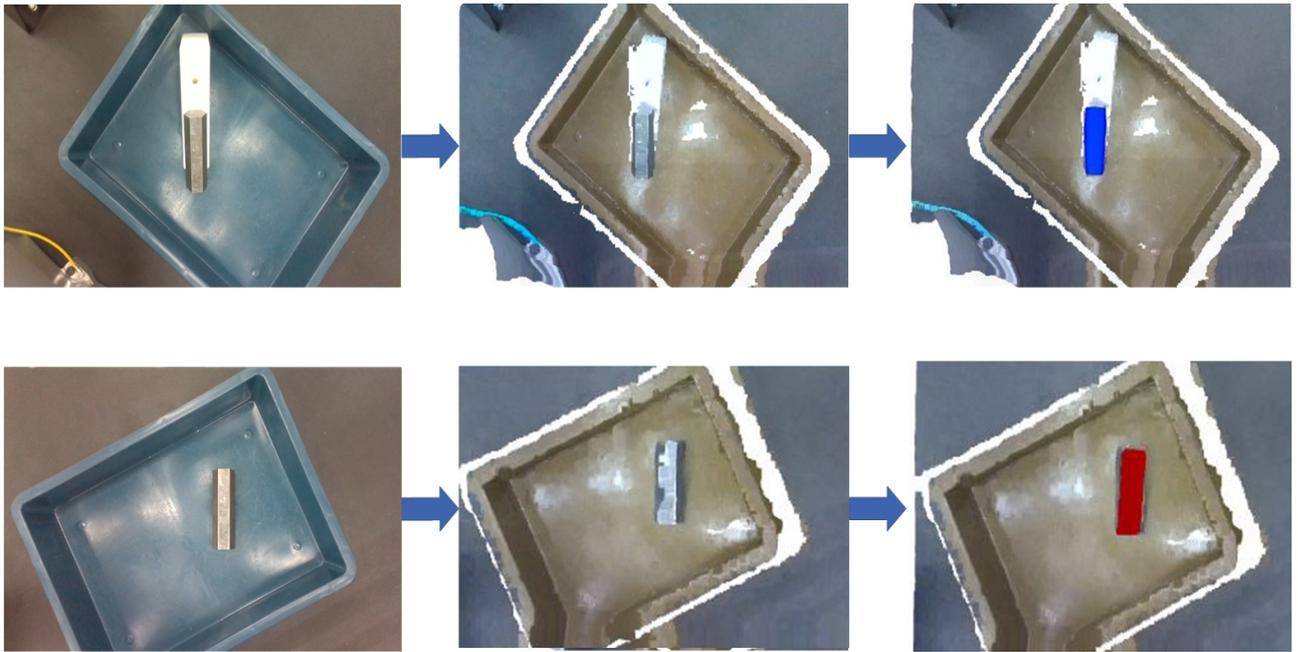


Рисунок 3.24 - Результирующее облако точек и наложенная 3D модель

### 3.7. Магнитный пневматический захват для промышленного манипулятора

После определения координат и ориентации объекта необходимо произвести его захват и транспортировку в необходимую точку, для чего используются захватные устройства (ЗУ).

Для захвата ферромагнитных деталей используют магнитные захватные устройства, которые используют силу магнитного поля для удержания объекта. Основной принцип работы заключается в использовании разного типа магнитов (ферритовые или неодимовые) и управления ими с помощью изменения направления магнитного поля или физического перемещения магнита внутри ЗУ [120].

В рамках диссертационного исследования принято решение о применении магнитного захватного устройства. Однако роботизированный комплекс Кука оборудован исключительно вакуумными магистралями, что технически исключает возможность использования захватного устройства на основе электромагнита [121].

Один из видов вакуумного магнитного захвата представлен в патенте [122]. Устройство состоит из корпуса с постоянным магнитом, отличающийся тем, что, с целью повышения надежности его работы, он снабжен установленным в корпусе с возможностью осевого перемещения поршнем и мембраной из диамагнитного материала, связанной с корпусом, причем корпус выполнен в виде цилиндра с отверстиями для подвода сжатого воздуха, а постоянный магнит установлен в поршне со стороны мембраны.

Недостатком устройства является низкий уровень отказоустойчивости при захвате изделия, габариты/ориентация которого отличны от установленных, а также отсутствие крепления к промышленному манипулятору

Еще одним магнитным захватом манипулятора является [123]. Согласно патенту, захват робота содержит корпус с установленным в нем магнитным захватным элементом, на боковой поверхности которого и коаксиально ему установлен другой захватный элемент, выполненный в виде вакуумной присоски, полость которой посредством каналов, выполненных в магнитном захватном элементе, соединена с вакуумной системой, снабженной пневмораспределителем, отличающимся тем, что, с целью повышения быстродействия, магнитный захватный элемент выполнен в виде постоянного магнита, а пневмораспределитель выполнен в виде пневмоцилиндра двустороннего действия, жестко закрепленного на неподвижном штоке, причем шток выполнен полым и имеет расширяющийся конус на выходе, а на цилиндрической части штока под углом 45- -50° к продольной оси выполнены две пары каналов, направленных в разные стороны и имеющих возможность попеременного перекрытия поршнем.

Недостатком устройства является сложность конструкции и ее габаритные размеры, что снижает грузоподъемность робота. К тому же, указанный прототип не оснащен дополнительными механизмами для увеличения погрешности при сближении с захватываемым изделием.

Недостатки представленных выше патентов не позволяют их использовать в разработанной системе компьютерного зрения с использованием RGB-D камеры, т.к. погрешность определения расстояния до объекта по оси Z составляет 2%. Чтобы не

допустить столкновения робота и объекта захвата необходимо учитывать полученную погрешность. Для этого ЗУ должно быть оснащено демпфером, позволяющим исключить погрешность определения расстояния по вертикальной оси.

ЗУ, которое полностью удовлетворяет поставленной задаче нет, поэтому автором разработан «Пневматический магнитный захват робота манипулятора» и произведена регистрация патента №226513. Отличительной особенностью разработанного устройства является наличие подпружиненно демпфера, который увеличивает допустимую погрешность сближения с захватываемой деталью и имеет вертикальное крепление к фланцу робота.

Такое решение позволяет предотвратить возможные аварии при захвате изделия, габариты/ориентация которого отличны от установленных, с сохранением исходной грузоподъемности робота.

На рисунке 3.25 представлен пневматический магнитный захват, проекционный вид и продольный разрез - сечение А-А.

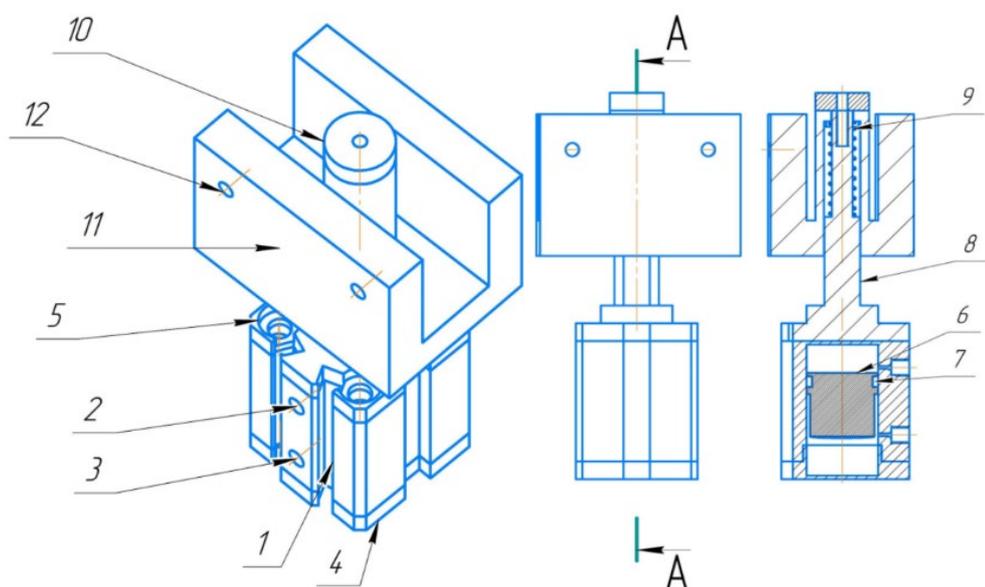


Рисунок 3.25 - Пневматический магнитный захват, проекционный вид и сечение А-А

Пневматический магнитный захват состоит из алюминиевого корпуса 1, отверстия для подвода сжатого воздуха через пневматические магистрали 2, 3,

защитной крышки 4, которая крепится через отверстие 5. Внутри корпуса 1 расположен поршень 6 с закрепленным неодимовым магнитом N38 и имеющий паз 7 под резиновый уплотнитель. Для перемещения корпуса 1 вертикально, относительно фланца робота, установлен шток 8 с пружиной 9 и заглушкой 10. Для крепления к роботу используется пластиковая платформа 11, с отверстиями для крепления 12.

Для корректной работы захвата, крепеж должен быть выполнен из ферромагнитных металлов.

Устройство работает следующим образом: Пневматический магнитный захват робота устанавливается на фланец робота посредством пластиковой платформы 11 с помощью четырех болтов М4 через отверстия 12. Перемещение поршня 6 с закрепленным на нём неодимового магнита N38 осуществляется посредством двух пневматических магистралей подключенных через быстросъемные цанговые штуцера, установленные в отверстия 2, 3. Для перемещения поршня 6 в паз 7 установлено уплотнительное кольцо. Скопление металлической стружки, на поверхности магнита позволяет избежать установлена защитная крышка 4, которая крепится четырьмя болтами М5 через отверстия 5 к алюминиевому корпусу 1. При перемещении к ферромагнитному изделию, во избежание столкновения, захват оснащен подпружиненным демпфером, состоящим из штока 8, пружины 9 и крышки 10. Крышка 10 прикручена к штоку 8 с помощью болта М3. При захвате ферромагнитного изделия по пневматической магистрали, подключенной к 2, подаётся сжатый воздух, перемещая поршень 6 к изделию. При отпуске изделия по пневматической магистрали, подключенной к 3 подаётся воздух и, одновременно, по магистрали, подключенной к 2 отбирается воздух, перемещая поршень 6 в положение "от изделия".

Вышеописанный захват позволяет роботу пневматического типа осуществлять захват ферромагнитных изделий, кроме того, увеличивается допустимая погрешность сближения с захватываемым изделием.

В «Пневматический магнитный захват робота манипулятора» используется магнит марки N38, где N (Normal) – магниты таких типов могут применяться при

нормальных температурах, то есть до 80 градусов Цельсия; 38 - указывает на Магнитную Энергию, измеряется в килоДжоуль на кубический метр. Этот критерий магнитов отвечает за мощность магнита или, так называемое, «усилие на отрыв», то есть сила, которую необходимо приложить к магниту, чтобы его «оторвать» от поверхности. Для магнита N38 Магнитная Энергия равняется 287-310 (36-39) килоДжоуль/м<sup>3</sup> (МегаГаусс-Эрстед).

Согласно техническим характеристикам магнита N38, диаметром 20 мм и толщиной 5 миллиметров при воздушном зазоре 1 миллиметр, образуемом за счет использования защитной крышки 4, удерживающая сила между магнитом и захватываемым изделием составляет 56 Н или 5.71041 кг.

Грузоподъемность робота Kuka KR4 R600 равняется 4 кг, дополнительно на фланце робота расположен захват, массой 162 грамм. Следовательно грузоподъемность робота снижена до 3838 грамм, что меньше максимальной грузоподъемности захвата.

Применение неодимового магнита позволяет значительно упростить систему за счет отсутствия токопровода на захватном механизме манипулятора, тем самым повысить надежность работы захвата.

Для захвата объектов под углом необходим магнит с большим удерживающим усилием. Для этого создано еще два магнитных захватных устройств с удерживающей силой 175 Н или 17.861 кг и 155 Н или 15.876 кг. Для увеличения удерживающего усилия в них используются неодимовые магниты из сплава N42, диаметром 36 миллиметров в первом случае и 32 миллиметра во втором случае, толщина магнита равняется 8 мм.

Захватное устройство с удерживающей силой 175 Н имеет воздушный зазор между магнитом и захватываемым изделием 2 миллиметра. Масса захватного устройства равняется 402 грамма.

Захватное устройство с удерживающей силой 155 Н имеет воздушный зазор между магнитом и захватываемым изделием 1 миллиметр. Масса захватного устройства равняется 344 грамма.

Разработанные захватные устройства выбираются исходя из задачи и могут быть заменены в течении нескольких минут.

Разработанные магнитные захватные устройства приведены на рис.3.26.

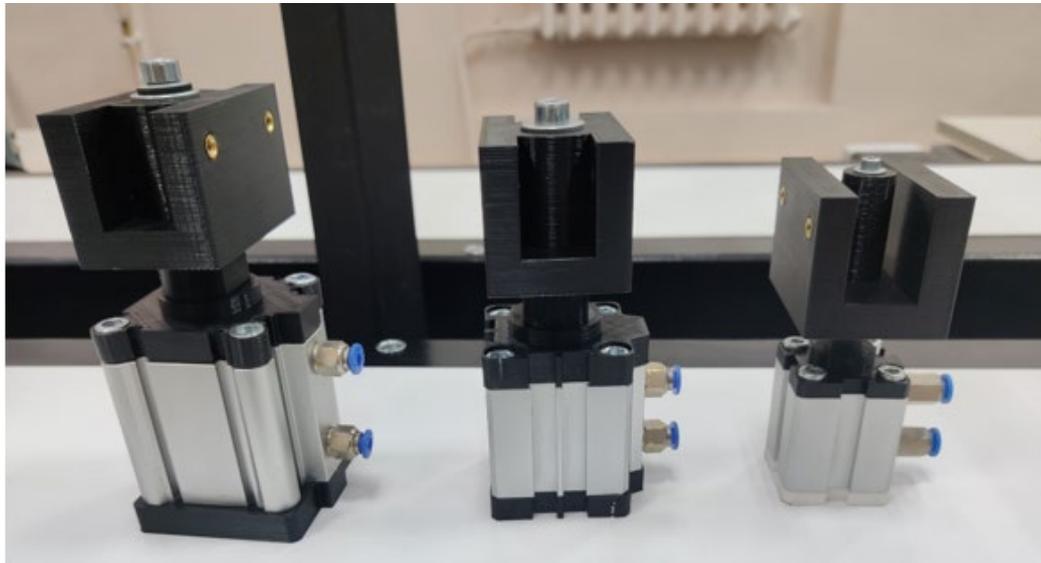


Рисунок 3.26 – Магнитные захватные устройства

### **Выводы по главе**

1. Разработан метод классификации и захвата манипулятором произвольно расположенных разнородных ферромагнитных деталей из накопителя с применением компьютерного зрения. Метод включает шесть этапов: определение границ накопителя, нейросетевую сегментацию объектов на основе дообученной модели YOLOv8, выбор объекта для захвата, определение пространственных координат объекта захвата и его нахождения возле стенок накопителя, перемещение объекта магнитным пневматическим захватным устройством.

2. Разработан модуль определения границ накопителя, обеспечивающий ограничение рабочей зоны манипулятора для исключения ложных срабатываний. Сегментация объектов в накопителе выполняется с помощью дообученной нейросетевой модели YOLOv8. Для сокращения времени затрачиваемого на сборку обучающего набора данных и повышения его качества, разработан способ автоматического формирования дообучающего набора данных, позволяющий автоматически в циклическом режиме получать фотографии объектов, с помощью поворота камеры, закрепленной на фланце. После чего осуществляется выбор

целевого объекта для захвата на основе минимального расстояния по вертикальной оси до камеры, измеряемого с использованием RGB-D камеры Intel RealSense D415i.

3. Разработан алгоритм определения пространственных координат объекта на основе интеграции нейросетевого метода и облака точек; на первом этапе алгоритма выполняется определение центра объекта захвата на основе фокусного расстояния камеры и расстояния до объекта по вертикальной оси, а на втором определении наклона объекта относительно манипулятора на основе сравнения 3D модели объекта в формате полигональной сетки и облака точек с применением алгоритма ICP (Iterative Closest Point). Для повышения точности расчет углов наклона объекта осуществляется при расположении камеры под прямым углом к маске сегментации, что уменьшает влияние ложных срабатываний.

4. Разработан «Пневматический магнитный захват робота манипулятора» и произведена регистрация патента №226513. Отличительной особенностью разработанного устройства является наличие подпружинено демпфера, который увеличивает допустимую погрешность сближения с захватываемой деталью и имеет вертикальное крепление к фланцу робота.

## **ГЛАВА 4. ЭКСПЕРИМЕНТАЛЬНЫЕ ИССЛЕДОВАНИЯ. ОБСУЖДЕНИЕ ПОЛУЧЕННЫХ РЕЗУЛЬТАТОВ**

В главе приведены результаты экспериментального применения методов и алгоритмов для решения задачи захвата промышленным манипулятором произвольно расположенных ферромагнитных объектов из накопителя. Детализируется технология формирования обучающего набора данных и дообучения нейросетевой модели. Проведено комплексное тестирование алгоритма определения пространственных координат объекта, анализ погрешностей позиционирования и оценка эффективности работы системы при различной конфигурации объектов в накопителе.

### **4.1. Автоматизированная сборка обучающего набора данных и дообучение нейронной сети**

В качестве накопителя, для тестирования метода классификации и захвата произвольно расположенных разнородных ферромагнитных деталей, выбран ящик размером 400x300x75 миллиметров. Такой размер и вид накопителя выбран исходя из того, что он:

- охватывает большую часть рабочей зоны манипулятора Kuka KR4 R600;
- позволяет хранить объекты в неориентированном виде;
- используется при партионной обработке на технологических операциях;
- легко и быстро заменяется в случае необходимости.

Основными видами заготовок для деталей, изготавливаемых из металлов и их сплавов, являются: сортовой материал, изготавливаемый прокатом, волочением и т.п. из черных металлов и цветных сплавов (прутки круглого, квадратного и шестигранного сечения, трубы) [124]. В связи с этим в качестве объектов захвата выбраны ферромагнитные заготовки трех видов: труба (tube), квадратная цельная заготовка (square), шестигранная цельная заготовка (hexagon).

Объекты захвата, имеют следующие размеры: квадратная цельная заготовка – 20x20x100 мм., шестигранная цельная заготовка 30x100 мм. и труба – 30x100 мм.

Для дообучения ИНС YOLOv8 на сегментацию выбранных видов заготовок с помощью способа автоматического формирования дообучающего набора данных собрано 700 фотографий и время, затрачиваемое на автоматическое его формирование, составило около 50 минут. При этом в наборе содержались фотографии с различными цветами фона ячейки и разным расположением объектов: объекта лежали по отдельности, объекта располагались навалом, перекрывая друг друга. По сравнению с традиционной ручной сборкой, предложенный подход позволил сократить временные затраты на 3 рабочих дня, обеспечивая при этом высокое разнообразие условий съемки и сохранение репрезентативности данных.

Для сегментации объектов в рамках накопителя произведено дообучение модели YOLOv8m (средний), в соответствии с таблицей 3.3., с различным количеством эпох и размером изображения 640 на 640 на подготовленном обучающем наборе данных.

Метрики обучения модели YOLOv8m, полученные в ходе 100 эпох, представлены на рисунке 4.1.

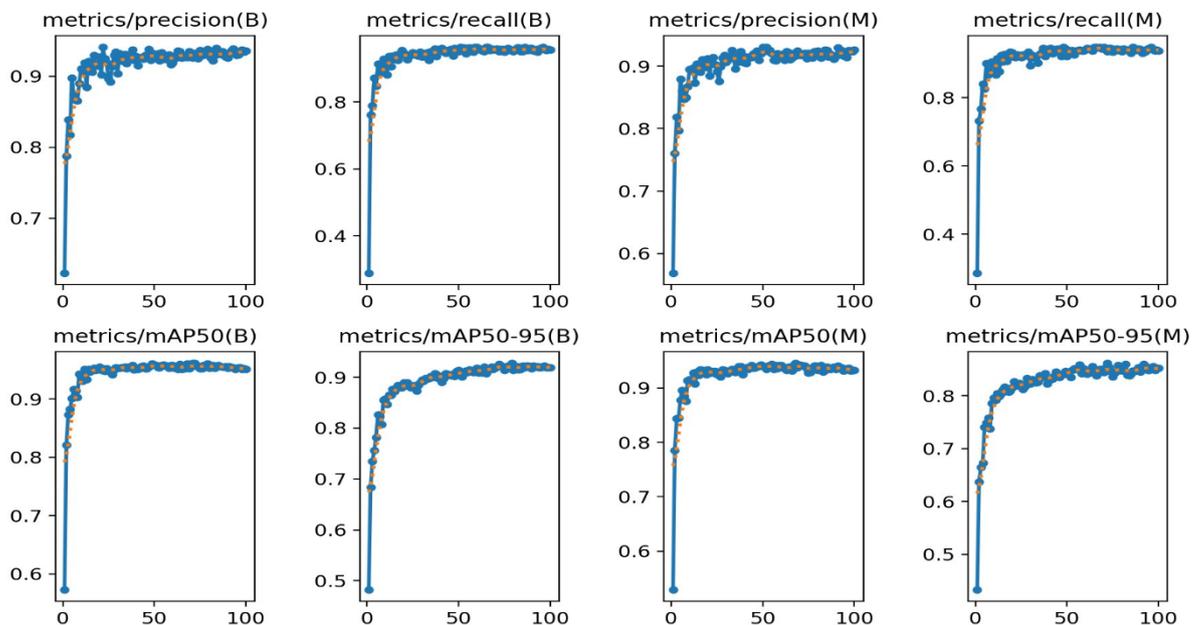


Рисунок 4.1 - Метрики при обучении и валидации модели YOLOv8m на 100 эпохах

Ключевая метрика общей производительности модели mAP50-95 (mean Average Precision от IoU=0.5 до 0.95 с шагом 0.05), дающая представление о точности детекции, достигла значения 0.9. Достижение такого значения метрики mAP50-95 свидетельствует о высоком качестве модели, способной надежно детектировать и сегментировать объекты с учетом их существенного перекрытия с IoU=0.5. Дополнительно анализ динамики метрик Precision и Recall для боксов (B) и масок (M) свидетельствует об эффективном процессе обучения модели. В результате можно сделать выводы, что модель демонстрирует устойчивую сходимость без признаков переобучения. Высокие итоговые значения метрик (mAP50→0.8, mAP50-95→0.7) подтверждают эффективность модели для поставленной задачи.

Пример сегментации на тестовой выборке приведен на рис.4.2.



Рисунок 4.2 - Сегментация объектов на тестовой выборке

В ходе валидации модели на тестовой выборке ложные срабатывания обнаружены не были, что свидетельствует о корректности функционирования модели. Однако при тестировании на видеопотоке выявлена ошибочная

сегментация объектов классов `square` и `hexagon`. Для устранения данного недостатка в обучающий датасет добавлено 240 изображений объектов класса `hexagon`, полученных с использованием ранее описанной методики сбора данных. Сбор дополнительных образцов осуществлялся посредством трех проходов роботом в рамках рабочей зоны при изменяющихся условиях освещения и ракурсов. В результате общий объем датасета достиг 940 изображений, а временные затраты на его формирование составили приблизительно 80 минут.

Распределение количества экземпляров по классам в итоговом обучающем наборе данных визуализировано на рисунке 4.3.

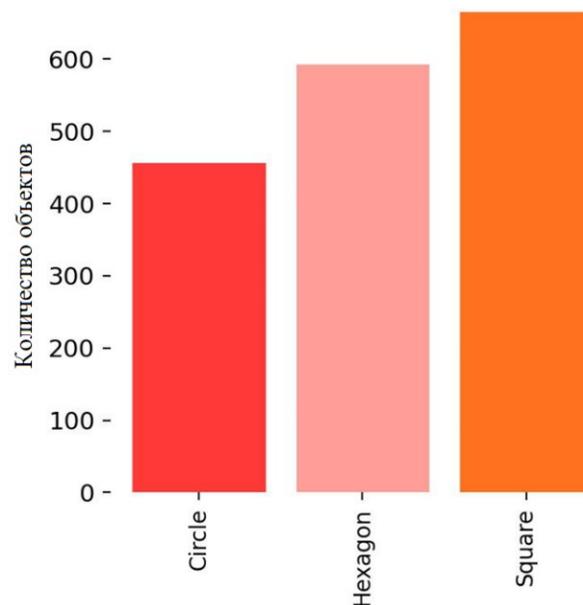


Рисунок 4.3 - Распределение классов в наборе данных

Анализ распределения показывает, что классы сбалансированы, это минимизирует риск смещения модели в сторону отдельных классов. Обучение на итоговом обучающем наборе данных производилось на архитектуре YOLOv8m в течение 250 эпох. Полученные в результате обучения метрики оценки производительности модели представлены на рисунке 4.4. Анализ графиков позволяет сделать вывод, что нейронная сеть показывает лучшие результаты, по сравнению с рисунком 4.1 и метрика mAP выше, чем в предобученной модели на датасете COCO (YOLOv8x):  $mAP_{50} \approx 0.89$ ;  $mAP_{50-95} \approx 0.67$ .

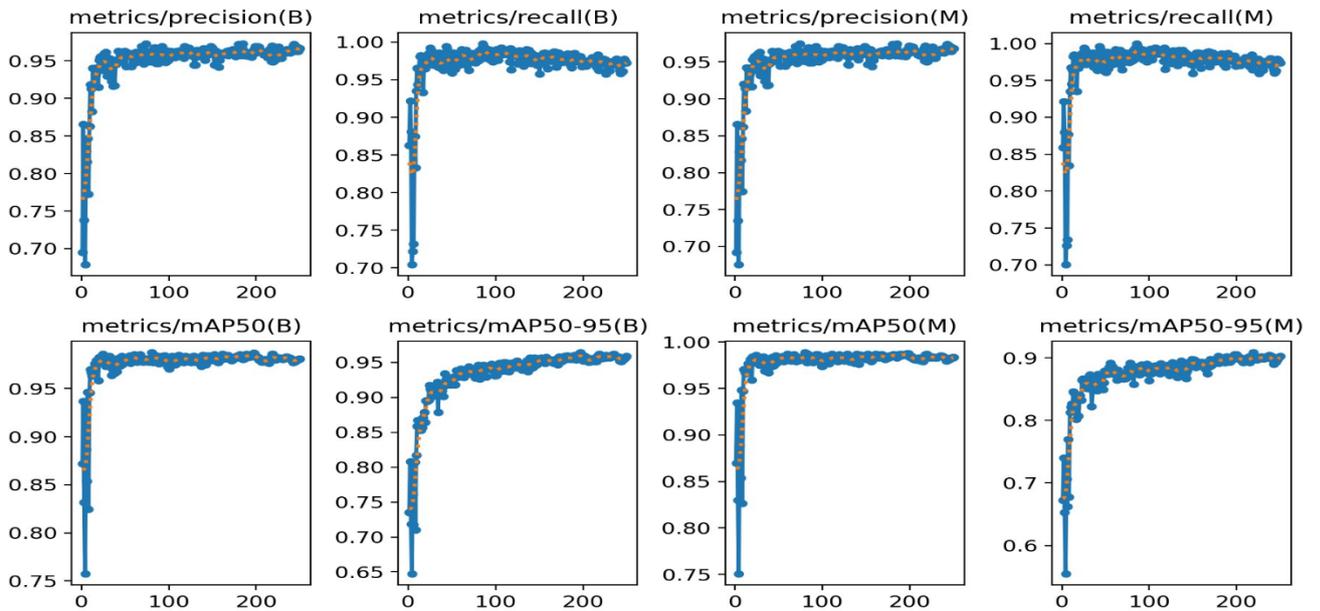


Рисунок 4.4 - Метрики при обучении и валидации модели YOLOv8m на 250 эпохах

Матрица ошибок представлена на рис.4.5 и позволяет в наглядном виде, оценить как часто модель правильно классифицирует объекты и какие именно ошибки она совершает. Процент неверной сегментации и перепутывания классов является минимальным, а большинство ошибок, присутствующих в матрице, носят единичный характер и не образуют выраженных зависимостей, что исключает наличие систематического смещения модели в отношении каких-либо конкретных классов. Пример сегментации объектов на видеопотоке приведен на рис.4.5.

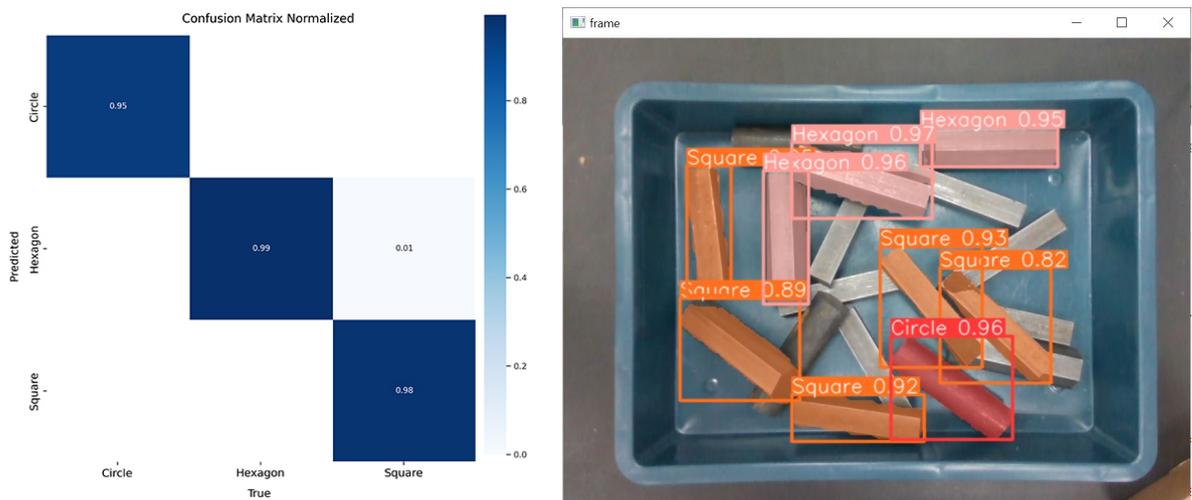


Рисунок 4.5 - Матрица ошибок дообученной модели YOLOv8m на 250 эпохах и пример сегментации объектов на видеопотоке

Дальнейшее тестирование проводилось на ноутбуке со следующими характеристиками: процессор Intel(R) Core (TM) i5-1035G1 CPU @ 1.00GHz 1.19 GHz, оперативная память 8 Гб, видеокарта NVIDIA GeForce MX350 и при закрепленной камере на роботе и расстоянии по вертикальной оси от центра захвата до основания в 450 миллиметров. Условия тестирования представлено на рис.4.6.



Рисунок 4.6 – Пример расположения объектов «навалом» в накопителе

Для тестирования сегментации объектов в накопителе и выбора объекта захвата проведена серия экспериментов с постепенным увеличением количества объектов и изменением их расположения. Пример объектов, расположенных в накопителе, приведен на рис.4.7, где:

- зеленым отмечен номер объекта;
- в зависимости от класса ограничивающие рамки выделены разными цветами: розовый цвет – класс Hexagon, оранжевый цвет – класс Square, красный цвет – класс Circle;
- красным цветом отмечено расстояние до объектов по Оси Z;
- красным контуром обведена маска сегментации, выбранного объекта захвата.

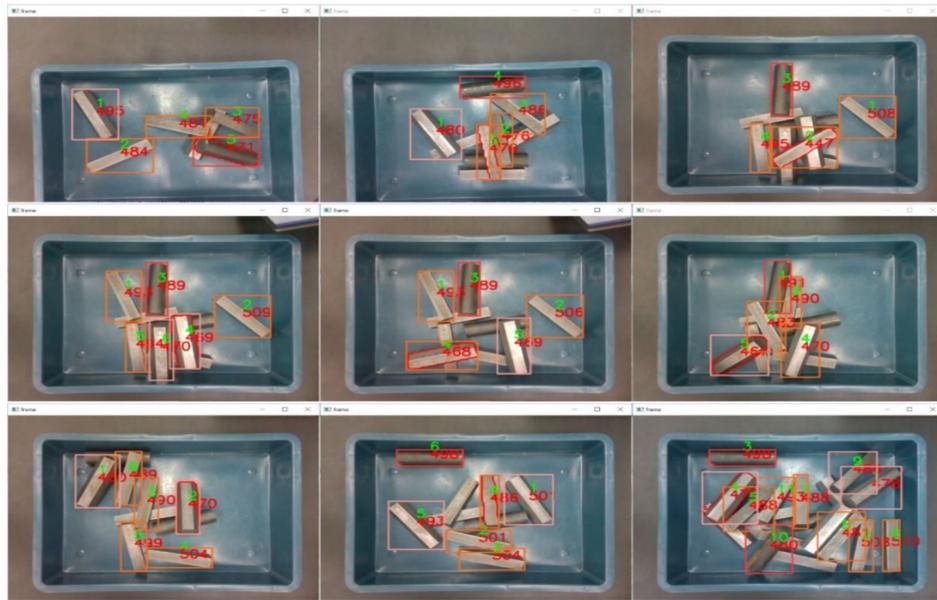


Рисунок 4.7 - Тестирование выбора объекта захвата

Анализ результатов тестирования с последовательно усложняющимися условиями проведения экспериментов подтверждает высокую эффективность дообученной нейросетевой модели. Несмотря на увеличение количества объектов, их плотность и вариативности расположения, модель демонстрирует стабильно точную семантическую сегментацию, корректно идентифицируя и выделяя объекты в соответствии с их классами, а алгоритм выбора целевого объекта захвата принимает верные решения даже в условиях частичных перекрытий и сложных компоновок.

## 4.2. Тестирование алгоритма определения пространственных координат объекта

### 4.2.1. Тестирование способов определения координат центра объекта

Тестирование способа перевода полученных значения центра объекта из пикселей в миллиметры, с помощью вычисления соотношения реальных размеров рабочей области в миллиметрах и количества пикселей, по каждой из осей, которые ей соответствуют (*Геометрический способ*) проводилось при закреплённой камере на роботе и расстоянии по вертикальной оси от центра захвата до объекта в 550 миллиметров. В рабочей зоне располагалась шахматная доска с размером ячеек

58x58 миллиметров. Расположение робота и шахматной доски в его рабочей зоне представлено на рис.4.8.

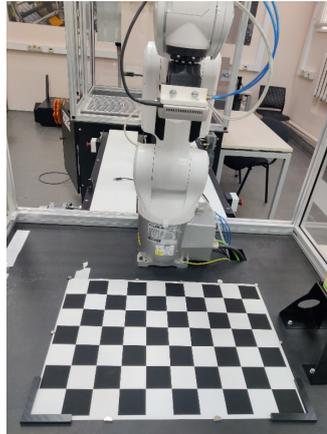


Рисунок 4.8 – Условия проведения тестирования способов преобразования координат центра объекта из пикселей в миллиметры

В результате эксперимента получены 54 точки с шагом 58 миллиметров по каждой из осей и итоговый размер рассматриваемой области 464x290 миллиметров. Таблица А.1 с полученными значениями, в результате эксперимента приведена в приложении А. График разброса погрешности измерений, при тестировании на шахматной доске, приведен на рис.4.9.



Рисунок 4.9 – График погрешности геометрического способа перевода из пикселей в миллиметры при тестировании на шахматной доске

Если значения погрешности положительные, то это означает что рассчитанная точка находится правее, нежели эталонное значение. В случае отрицательной погрешности рассчитанная точка находится левее. Среднее значение погрешности по модулю по Оси  $X \approx 2,6$  мм, по Оси  $Y \approx 0,47$ .

Тестирование на шахматной доске не учитывает:

- погрешность, получаемую при сегментации с помощью YOLOv8;
- высоту объекта.

Поэтому проведено второе тестирование, на реальном объекте, который представляет ферромагнитную заготовку размером 20x20x100 мм. Таблица А.3 с полученными значениями, в результате эксперимента приведена в приложении А.

График разброса погрешности измерений, при тестировании на ферромагнитной заготовке, приведен на рис.4.10.

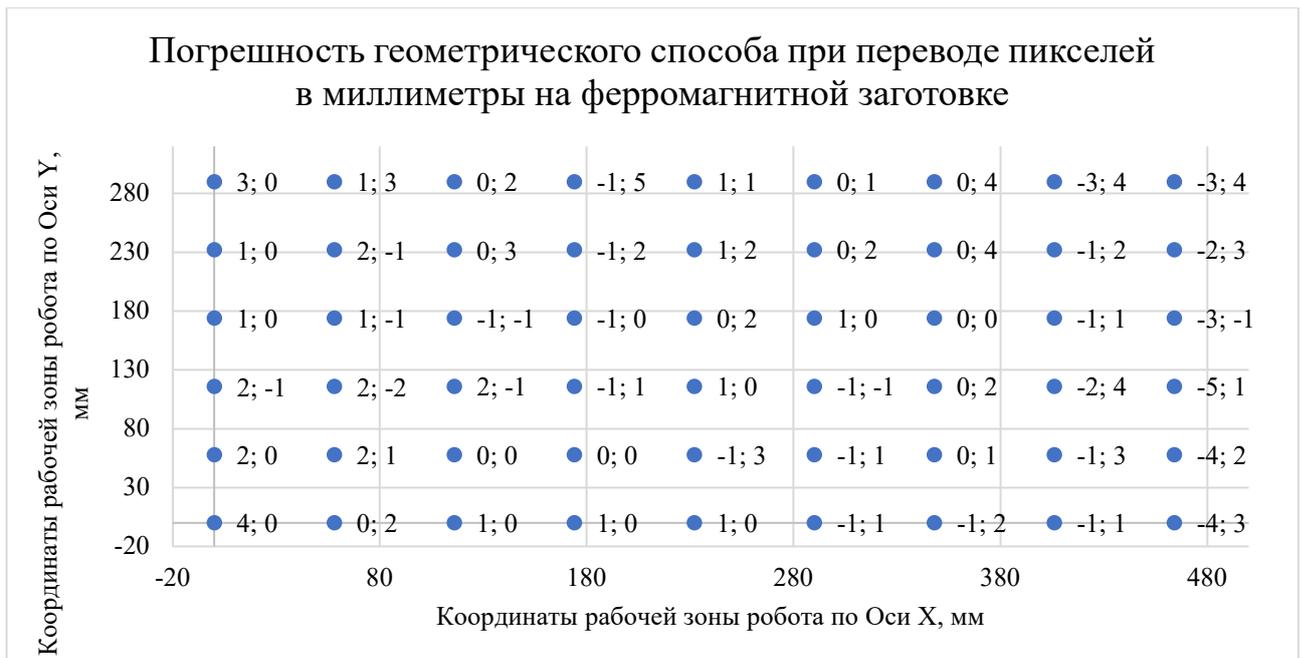


Рисунок 4.10 – График погрешности геометрического способа перевода из пикселей в миллиметры при тестировании на ферромагнитной заготовке

Среднее значение погрешности по модулю по Оси  $X \approx 1,3$  мм, по Оси  $Y \approx 1,5$ . Увеличение погрешности по сравнению с экспериментом на шахматной доске обусловлено неидеальным позиционированием объектов в эталонных точках

На основании проведенного тестирования сделаны выводы, что ошибки измерений связаны с несколькими факторами:

- несовпадение нулевых точек — расхождение между началом координат робота и камеры;
- ошибка определения центра — неточность выделения центра объекта системой технического зрения;
- дефекты калибровочного образца — неидеальная геометрия поверхности шахматной доски. Для идеальной калибровки необходимо использовать шахматные доски на жесткой основе [125];
- ошибка масштабирования — некорректный расчет пиксельного соотношения для ячеек калибровочной сетки.

Далее проведено тестирование способа преобразования координат из пикселей в миллиметры, через фокусное расстояние, на шахматной доске (п. 3.6) в идентичных условиях. Результаты измерений приведены в таблице А.2 приложения А. График разброса погрешности представлен на рис. 4.11.



Рисунок 4.11 – График погрешности преобразования координат из пикселей в миллиметры, через фокусное расстояние, на шахматной доске

Как следует из графика, представленного на рис.4.11 средняя погрешность по модулю Оси X  $\approx 8,9$  мм, по Оси Y  $\approx 5,76$ . При этом ячейки шахматной доски,

находящиеся под центром фланца робота имеют нулевую погрешность, а на краях шахматной доски погрешность максимальная.

Погрешность, возникающая при одинаковых значениях по Оси X и разных по Оси Y и наоборот связана с точностью вычисления расстояния до объекта по вертикальной оси камерой. В процессе тестирования определено, что расстояние до ячеек шахматной доски «прыгает» в диапазоне 1-3 миллиметр.

Второе тестирование, на ферромагнитной заготовке, проводилось в идентичных условиях, что и геометрическим способ. Таблица с полученными значениями, в результате эксперимента приведена в таблице А.4 в приложении А.

График разброса погрешности измерений, при тестировании на ферромагнитной заготовке, приведен на рис.4.12.



Рисунок 4.12 – График погрешности преобразования координат из пикселей в миллиметры, через фокусное расстояние, при тестировании на ферромагнитной заготовке

Как следует из графика, представленного на рис.4.12 средняя погрешность по модулю Оси X  $\approx 4,4$  мм, по Оси Y  $\approx 2,6$ . Распределение погрешностей носит более «хаотичный» характер по сравнению с тестами на шахматной доске, что объясняется неидеальностью поверхности объекта и погрешностями его

сегментации. При этом сохраняется общая тенденция: погрешность возрастает при удалении от центра фланца робота.

Основные преимущества способа преобразования координат из пикселей в миллиметры, через фокусное расстояние:

- отсутствие необходимости калибровки базы робота и привязки к точке (0,0) камеры;
- исключение этапа использования шахматной доски для предварительной настройки;
- независимость от высоты расположения объекта по вертикальной оси.

Недостатки способа:

- рост погрешности при удалении объекта от центра фланца робота;
- необходимость использования датчика, позволяющего вычислять расстояние до объекта по вертикальной оси.

Для устранения погрешностей в пункте 3.6 предложена модификация к способу преобразования координат из пикселей в миллиметры, через фокусное расстояние, тестирование которого проводилось согласно алгоритму, приведенному на рис.4.13.

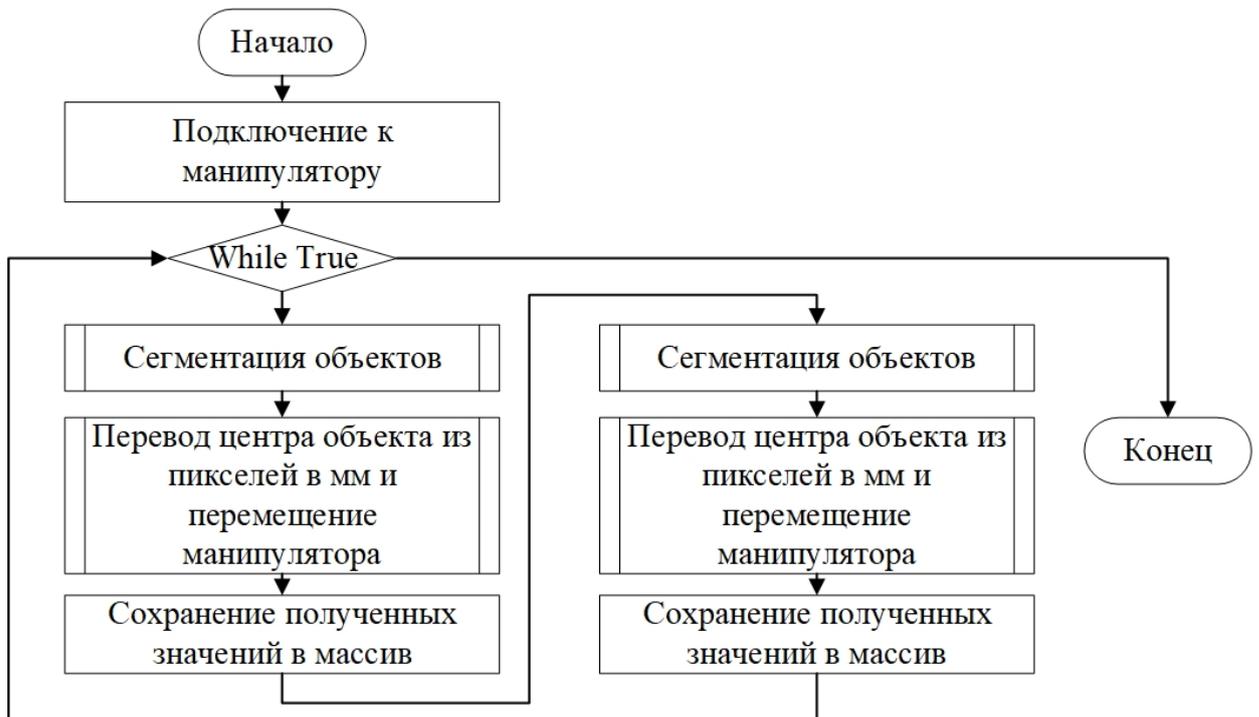


Рисунок 4.13 – Тестирование преобразования координат из пикселей в миллиметры, при предложенной модификации, на ферромагнитной заготовке

Таблица А.5 с полученными значениями, в результате эксперимента приведена в приложении А. График разброса погрешности измерений приведен на рис.4.14.



Рисунок 4.14 – График погрешности преобразования координат из пикселей в миллиметры, через фокусное расстояние при предложенной модификации, на ферромагнитной заготовке

Как следует из графика, представленного на рис.4.14 средняя погрешность по модулю Оси X  $\approx 0,7$  мм, по Оси Y  $\approx 0,6$ . Незначительная погрешность обусловлена двумя основными факторами. Во-первых, данные о глубине, получаемых с камеры Intel RealSense D415i изменяются в процессе измерения. Во-вторых, минимальные отклонения возникают из-за неидеального позиционирования объектов в эталонных точках в процессе проведения измерений.

#### 4.2.2. Тестирование алгоритма определения углов наклона объекта захвата

Экспериментальные исследования определения углов наклона объекта захвата проводились в различных условиях для оценки надежности и точности предложенного решения. Тестирование выполнялось в трех основных сценариях: при расположении заготовки на дне накопителя, при использовании регулируемой подставки, позволяющей устанавливать углы наклона от 5 до 27°, и в условиях

частичного перекрытия объектов при их совместном размещении в накопителе. Для измерения эталонных значений углов наклона использовался магнитный угломер RZ 838 с погрешностью  $\pm 0,3^\circ$ .

Для каждого класса объектов было выполнено по 9 замеров с изменением угла наклона по оси Y в диапазоне от  $-22^\circ$  до  $+22^\circ$ . Изменение угла по оси X проводилось в меньшем диапазоне из-за особенностей геометрии заготовок. Для объектов класса Circle измерения по оси X не проводились в связи с тем, что объект имеет круглую форму. Дополнительно выполнено 23 замера в реальных условиях при наличии множества объектов в накопителе. Общее количество измерений составило 50.

По результатам тестирования получены следующие данные:

1. Значения реального наклона заготовки относительно камеры;
2. Значения наклон заготовки относительно камеры, измеренные без поворота робота к маске сегментации;
3. Значения наклон заготовки относительно камеры, измеренный под прямым углом к маске сегментации;
4. Погрешность вычисления углов наклона.

В результате произведено 50 замеров. Таблица Б.1 с результатами тестирования приведена в приложении Б.

На всех графиках в дальнейшем по оси абсцисс откладывается номер замера, а по оси ординат угол наклона объекта относительно камеры.

На рис.4.15 приведен расчет наклон заготовки относительно Оси X и Оси Y камеры для классов Square и Hexagon, при тестировании, когда заготовка находилась на дне накопителя и на подставке.

Наклон заготовки относительно Оси X камеры обозначен синим цветом, Наклон заготовки относительно Оси X камеры, измеренный без поворота робота к маске сегментации обозначен оранжевым цветом, Наклон заготовки относительно Оси X камеры, измеренный под прямым углом к маске сегментации обозначен серым цветом. Аналогично для Оси Y.

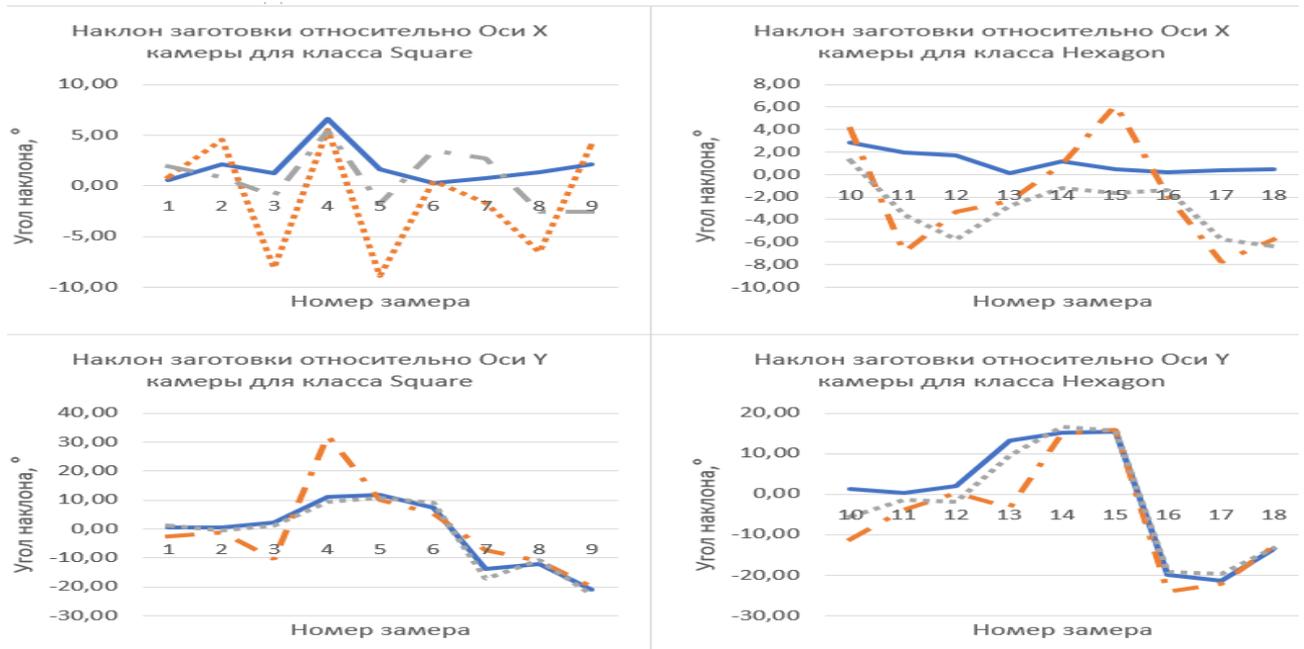


Рисунок 4.15 - Тестирование наклона заготовки относительно Оси X и Оси Y камеры для класса Square и Hexagon

Проведенные эксперименты выявили существенное влияние ориентации камеры относительно маски сегментации на точность измерений. Для класса Square погрешность определения углов наклона без коррекции позиции робота составила по Оси X  $4.01^\circ$ , по Оси Y  $5.65^\circ$ , в то время как при повороте камеры под прямым углом к маске сегментации погрешность снизилась до  $2.57^\circ$  по Оси X,  $1.55^\circ$  по Оси Y. Аналогичная тенденция наблюдается для класса Hexagon: погрешность уменьшилась с  $4.45^\circ$  по Оси X,  $4.57^\circ$  по Оси Y до  $4.06^\circ$  по Оси X,  $2.24^\circ$  по Оси Y. Увеличенная погрешность по оси X может быть связана с особенностями геометрии объекта. Заготовка класса Hexagon имеет шесть идентичных граней, что приводит к неоднозначности при сопоставлении точек в алгоритме ICP (Iterative Closest Point). В некоторых случаях алгоритм ошибочно определяет ориентацию объекта, что проявляется в виде повышенной погрешности по оси X.

На рис.4.16 приведен расчет наклон заготовки относительно Оси Y камеры для класса Circle, при тестировании, когда заготовка находилась на дне накопителя и на подставке.

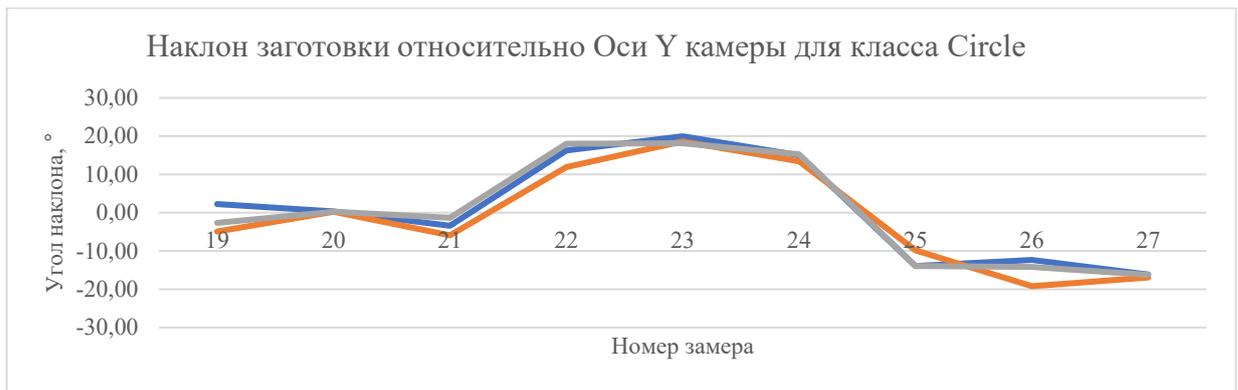


Рисунок 4.16 – Тестирование наклона заготовки относительно Оси Y камеры для класса Circle

Для круглых заготовок (класс Circle) погрешность определения угла наклона по оси Y без коррекции позиции робота составила  $3.21^\circ$ , в то время как при повороте камеры под прямым углом к маске сегментации погрешность снизилась до  $1.43^\circ$ . Это подтверждает универсальность метода коррекции для объектов различной геометрии.

Для определения угла поворота объекта относительно камеры по оси Z использовалась следующая методика:

- робот вручную позиционировался относительно заготовки таким образом, чтобы угол поворота составлял ровно  $90^\circ$ ;
- после возврата в стартовую позицию с помощью метода FindContours производилось определение угла наклона маски сегментации.

В ходе 27 замеров средняя погрешность составила  $0.62^\circ$  и связано с округлением значения угла наклона рамки сегментации методом FindContours и не точностью исходных данных, т.к. робот подводился к объекту вручную.

Для оценки надежности алгоритма в условиях, максимально приближенных к эксплуатационным, проведены замеры при наличии в накопителе множества объектов различных классов с частичным взаимным перекрытием. Результаты тестирования приведены на рис.4.17.

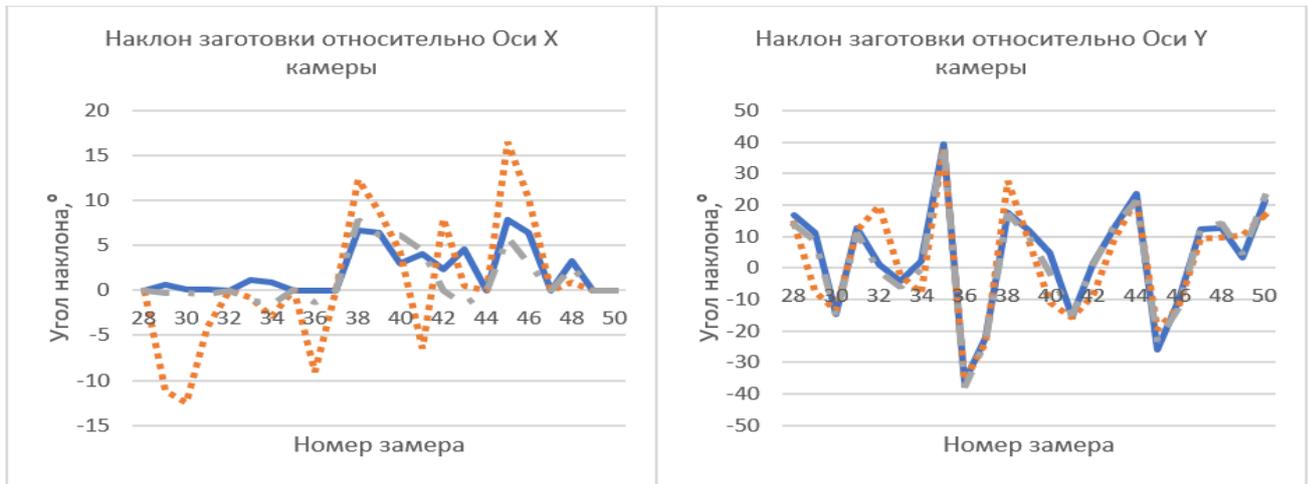


Рисунок 4.17 – Тестирование наклона заготовки относительно Оси X и Оси Y камеры в условиях перекрытия между объектами

Тестирование в условиях перекрытия между объектами подтвердило данные, которые получены ранее. Погрешность без поворота робота к маске сегментации равняется по Оси X  $5.49^\circ$ , по Оси Y  $5.57^\circ$ , а при расчете углов наклона под прямым углом к маске сегментации равняется по Оси X  $1.73^\circ$ , по Оси Y  $1.77^\circ$ .

Общая погрешность по результатам тестирования в различных условиях получилась равной без поворота робота к маске сегментации по Оси X  $4.79^\circ$ , по Оси Y  $4.93^\circ$ , а при расчете углов наклона под прямым углом к маске сегментации по Оси X  $2.61^\circ$ , по Оси Y  $1.75^\circ$ .

По результатам проведенного тестирования сделаны следующие выводы.

- Поворот камеры под прямым углом к маске сегментации позволяет снизить погрешность в среднем на 45% по оси X и на 64% по оси Y.
- Метод определения наклона маски сегментации по вертикальной оси имеет погрешность  $0.62^\circ$ , что никак не влияет на корректный захват объекта.
- Погрешность измерений не зависит от класса объекта, а следовательно этот метод может быть применим не только к ферромагнитным заготовкам, рассматриваемым в диссертации, но и к другим ферромагнитным объектам.
- При расчете углов наклона объекта относительно камеры без поворота робота к маске сегментации погрешность выше, что может приводить к ложным

срабатываниям. Поворот камеры под прямым углом к маске сегментации позволяет уменьшить погрешность на  $2.18^\circ$  по Оси X и на  $3.18^\circ$  по Оси Y.

### 4.3. Комплексное тестирование решения для сортировки ферромагнитных деталей промышленным манипулятором с использованием компьютерного зрения 3D зрения и нейросетевой модели

Для комплексного тестирования выбрано магнитное захватное устройство с удерживающей силой 155Н, т.к. при использовании ЗУ с удерживающей силой 56Н происходили случаи неудачного схвата заготовок из-за погрешности вычисления углов наклона объектов относительно камеры или в процессе перемещения объектов на максимальной скорости в автоматическом режиме (2 м/с) объект мог изменить свое положение из-за недостаточной удерживающей силы. Магнитное захватное устройство с удерживающей силой 175Н слишком большое для захвата заготовок, используемых в диссертации, т.к. площадь его основания выступает за край заготовки на 17,5 мм, а в случае использования магнитного ЗУ с удерживающей силой 155Н этот параметр равняется 12,5 мм. Такой большой выход за край заготовки может привести к соприкосновению с другими заготовками, тем самым произойдет изменение их расположения в накопителе. Чертежи оснований всех магнитных ЗУ приведены на рис.4.18.

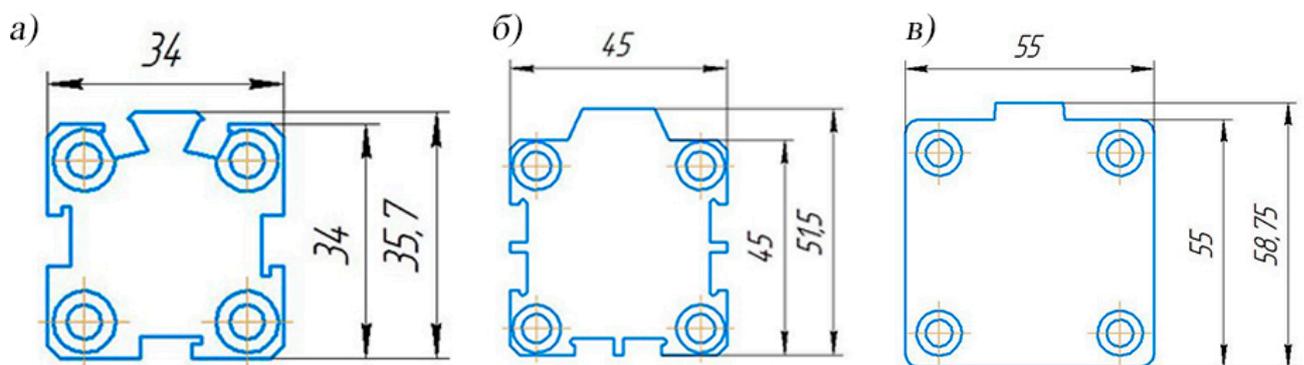


Рисунок 4.18 – Основания магнитных захватных устройств, где а) ЗУ с удерживающей силой 56 Н, б) ЗУ с удерживающей силой 155 Н, в) ЗУ с удерживающей силой 175 Н

Для проведения тестирования в накопителе случайным образом сформированы три конфигурации расположения объектов, где детали размещены навалом, каждая из которых содержит объекты всех трех классов. Конфигурации представлены на рис.4.19, где 1) Белый фон робототехнической ячейки, 2,3) Черный фон робототехнической ячейки.



Рисунок 4.19 – Конфигурации расположения объектов при проведении тестирования

При проведении испытаний системы осуществлялась оценка следующих ключевых параметров приведенных ниже.

- Идентификация класса объекта. Проверка корректности распознавания и классификации объектов (Square, Hexagon, Circle) с использованием дообученной нейросетевой модели.
- Измерение расстояния по вертикальной оси. Точность определения дистанции до объекта с использованием данных RGB-D камеры Intel RealSense D415i.
- Проверка успешности захвата. Контроль надёжности захвата объектов магнитным устройством с фиксацией неудачных попыток.
- Проверка точности позиционирования. Оценка отклонений установки объектов относительно эталонных значений по осям X и Y.

Для оценки точности позиционирования разработаны и изготовлены на 3D-принтере три специализированные формы, точно соответствующие геометрии тестируемых объектов. Формы размещены в робототехнической ячейке с шагом 50 мм по оси X.

Общие таблицы В.1-В.3 с результатами тестирования приведены в приложении В.

В первой конфигурации в накопителе расположено 20 заготовок, ошибок с неверным определением класса нет, средняя погрешность вычисления расстояния по вертикальной оси составила  $0,9 \pm 0,5$  миллиметра, средняя погрешность расположения объекта по Оси X равняется 0,72, по Оси Y 0,66.

Во второй конфигурации в накопителе расположено 21 заготовка, ошибок с неверным определением класса нет, средняя погрешность вычисления расстояния по вертикальной оси составила  $0,76 \pm 0,5$  миллиметра, средняя погрешность расположения объекта по Оси X равняется 0,62, по Оси Y 0,81.

В третьей конфигурации в накопителе расположено 26 заготовок, ошибок с неверным определением класса нет, средняя погрешность вычисления расстояния по вертикальной оси составила  $0,96 \pm 0,5$  миллиметра, средняя погрешность расположения объекта по Оси X равняется 0,8, по Оси Y 0,8.

Средняя погрешности вычисления расстояния по вертикальной оси по модулю равняется  $0,88 \pm 0,5$  миллиметра, что подтверждает исследование [126], в котором погрешность вычисления на расстоянии от 150 до 500 миллиметров равняется 0,99 миллиметра. Максимальное отклонение по модулю равняется 2 миллиметра, что компенсируется за счет подпружиненного демпфера ЗУ.

В процессе тестирования выявлены случаи, когда на второй итерации сегментации не найден объект. В таком случае робот возвращается в исходное положение и повторяет цикл заново. На первой сцене для захвата 20 объектов выполнено 23 итерации, на второй сцене для захвата 21 объекта выполнено 25 итераций, на третьей сцене для захвата 26 объектов выполнено 32 итерации. Средний процент не нахождения объектов на второй итерации сегментации составляет 16%, что не влияет на захват объектов, но увеличивает время, затрачиваемое на сортировку накопителя.

Тестирование проводилось в ручном режиме, но основной режим работы робота является автоматическим. Поэтому проведено несколько тестов на разных скоростях, которые показывают среднее затрачиваемое время на сортировки

одного объекта. В таблице 4.1 приведены результаты 5 испытаний на скоростях 30,50,75,100%.

Таблица 4.1 – Тестирование времени сортировки одного объекта.

Скорость выполнения	1 итерация, сек.	2 итерация, сек.	3 итерация, сек.	4 итерация, сек.	5 итерация, сек.	Общее время, сек.
30%	20,55	18,45	23,25	22,44	23,82	108,51
50%	16,03	16,58	17,24	17,13	15,05	82,03
75%	12,98	14,21	12,88	14,45	12,57	67,09
100%	11,33	10,60	11,51	10,40	11,46	55,30

Скорость выполнения отличается в зависимости от первоначального наклона объекта к камере, при увеличении угла отклонения маски сегментации к камере по вертикальной оси от прямого угла время, затрачиваемое роботом на поворот, увеличивается. На скорости 100% возникает проблема со смещением круглой заготовки в процессе перемещения, чтобы исправить эту проблему могут использоваться опциональные подъемные элементы позволяющие оптимально перемещать круглые или неплоские заготовки [127]. Использование такого типа дополнительного элемента приводит к невозможности захвата объектов другой формы. В рамках диссертационной работы предполагается что объекты могут быть разной формы, поэтому чтобы не происходило смещения круглой заготовки в процессе перемещения, рекомендуется использовать магнитный захват без дополнительных подъемных элементов и скорость движения робота 50%.

### Выводы по главе

1. В главе описано проведение и результаты экспериментального исследования применения разработанных методов и алгоритмов, обеспечивающих решение задачи захвата промышленным манипулятором произвольно расположенных ферромагнитных объектов из накопителя. Для обеспечения надежного распознавания объектов проведено обучение модели YOLOv8 на собранном в автоматическом режиме наборе данных, содержащем 940 изображений ферромагнитных заготовок трех классов: труба (tube), квадратная (square) и шестигранная (hexagon). Полученная модель демонстрирует высокую точность детекции с значением метрики mAP50-95 на уровне 0.89, что

подтверждает ее эффективность. Модель успешно идентифицирует объекты в условиях частичного перекрытия, изменяющегося освещения и различных ракурсов.

2. Сравнительное тестирование подтвердило высокую эффективность разработанных алгоритмов: способ преобразования координат из пикселей в миллиметры обеспечил погрешность не более 0.7 мм по оси X и 0.6 мм по оси Y, превзойдя существующие аналоги на 0.6 мм и 0.9 мм соответственно, а алгоритм определения углов наклона объектов продемонстрировал среднюю точность  $2.61^\circ$  по оси X и  $1.75^\circ$  по оси Y, что на  $2.18^\circ$  и  $3.18^\circ$  точнее существующих подходов. Повышение точности измерений обеспечено благодаря комплексному подходу, включающему несколько ключевых технологий. Интеграция данных с RGB-D камеры позволила получить точные данные о глубине сцены. Модифицированный алгоритм ICP с коррекцией позиции камеры относительно маски сегментации значительно улучшил определение пространственной ориентации объектов.

3. Для оценки эффективности разработанного решения в условиях, приближенных к реальным производственным, проведено 107 тестовых испытаний. Тестирование выполнялось на трех различных сценариях, отличающихся количеством объектов в накопителе и характеристиками фона, что позволило оценить устойчивость системы к изменяющимся условиям работы. Во всех случаях система продемонстрировала высокую точность распознавания и классификации объектов, правильно идентифицируя их принадлежность к одному из трех классов. Средняя погрешность определения расстояния по вертикальной оси составила 0,88 мм, что подтверждает высокую точность измерений. Для обеспечения максимальной стабильности процесса и минимизации смещения объектов, в особенности круглых заготовок, в ходе испытаний определена и рекомендована оптимальная скорость движения робота, составляющая 50% от максимальной.

## Заключение

В результате диссертационных исследований связанных с разработкой и применением методов и алгоритмов сортировки ферромагнитных деталей промышленным манипулятором с использованием нейросетевого моделирования и компьютерного зрения получены следующие результаты.

1. Проанализированы современные исследования и методы управления РТК, с использованием современных интеллектуальных технологий, выявлены ограничения в функциональности действующих систем управления промышленными манипуляторами и спроектирована модульная архитектура системы управления включающая подсистему удаленного управления с реализацией методов двусторонней связи и параллельной обработки команд, а также подсистему компьютерного 3D-зрения, обеспечивающую определение пространственного положения объектов за счет комбинации RGB-D сенсоров и нейросетевых алгоритмов, специально оптимизированных для задач сегментации ферромагнитных деталей в условиях производственной среды.

2. Разработана архитектура системы дистанционного управления, обеспечивающая двустороннюю связь и передачу команд управления в потоковом режиме между контроллером робота и удаленным вычислительным устройством. Система позволяет в автоматическом режиме передавать управляющие программы, полученные с помощью компьютерного зрения на контроллер робота и производить непосредственное управление движением манипулятора и захватными устройствами.

3. Разработан алгоритм определения пространственных координат объекта на основе интеграции нейросетевого метода и построения облака точек. Реализованные в алгоритме методики позволяют точно определять координаты в трехмерной СК и пространственную ориентацию объектов в сложных условиях, включая случаи частичного перекрытия и наложения деталей. Проведенные лабораторные эксперименты подтвердили высокую эффективность алгоритма: достигнутая точность определения координат (погрешность не превышает  $\pm 0.7$  мм)

и углов ориентации (погрешность в пределах  $\pm 2.61^\circ$ ) является достаточной для надежного захвата объектов манипулятором в автоматическом режиме.

4. Предложен метод классификации и захвата произвольно расположенных разнородных деталей из накопителя промышленным манипулятором. Метод имеет преимущество перед существующими решениями, позволяя осуществлять захват объектов, расположенных в любом порядке, и обеспечивает стабильность выполнения операции захвата при различных конфигурациях расположения объектов.

5. Разработана конструкция пневматического магнитного захвата, отличающаяся от существующих моделей наличием пружинного демпфера. Это улучшение увеличивает допустимую погрешность при сближении с захватываемой деталью и позволяет вертикально крепить захват к фланцу робота, что делает его возможным для работы в системах сортировки объектов, где возможны погрешности при вычислении расстояния в вертикальном направлении.

6. Проведены экспериментальные исследования и тестирования разработанных методов и алгоритмов в лабораторных условиях на промышленном манипуляторе, позволившие оценить точность определения пространственных координат объекта и стабильность работы метода классификации и захвата произвольно расположенных деталей в накопителе. Нейросетевая классификация объектов демонстрирует стабильность при различном освещении и конфигурации расположения объектов, включая случаи частичного перекрытия деталей.

**СПИСОК ЛИТЕРАТУРЫ**

1. ГОСТ Р 70988-2023 Система стандартов в цифровой промышленности. Основные положения. Общие требования к системе
2. Chukurna O. et al. Modern trends in digital transformation of marketing & management. – 2023.
3. Цифровая трансформация: эффекты и риски в новых условиях / Рук. авт. колл. П. Б. Рудник, Т. С. Зинина; под ред. И. Р. Агамирзяна, Л. М. Гохберга, Т. С. Зининой, П. Б. Рудника; Нац. исслед. ун-т «Высшая школа экономики». – М.: ИСИЭЗ ВШЭ, 2024. – 156 с. – 150 экз. – ISBN 978-5-7598-3009-2 (в обл.).
4. Международная федерация робототехники URL: <https://ifr.org/> (дата обращения: 10.07.2025).
5. Черноусан, М. И. Трансформация управления промышленными компаниями в условиях цифровизации / М. И. Черноусан, Е. В. Козина // Тенденции развития науки и образования. – 2022. – № 88-2. – С. 93-97. – DOI 10.18411/trnio-08-2022-71. – EDN AUUXUR.
6. Ermolov, I. (2020). Industrial Robotics Review. In: Kravets, A. (eds) Robotics: Industry 4.0 Issues & New Intelligent Control Paradigms. Studies in Systems, Decision and Control, vol 272. Springer, Cham. [https://doi.org/10.1007/978-3-030-37841-7\\_16](https://doi.org/10.1007/978-3-030-37841-7_16)
7. Rudomanenko, A.P., Chernyadev, N.A., Vorotnikov, S.A. (2021). Adaptive Control System for Industrial Robotic Manipulator. In: Yuschenko, A. (eds) Modern Problems of Robotics. MPoR 2020. Communications in Computer and Information Science, vol 1426. Springer, Cham. [https://doi.org/10.1007/978-3-030-88458-1\\_12](https://doi.org/10.1007/978-3-030-88458-1_12)
8. Patel V. V., Liarokapis M. V., Dollar A. M. Open robot hardware: progress, benefits, challenges, and best practices //IEEE Robotics & Automation Magazine. – 2022. – Т. 30. – №. 3. – С. 123-148.
9. M. Laskey et al., "Robot grasping in clutter: Using a hierarchy of supervisors for learning from demonstrations," 2016 IEEE International Conference on Automation

Science and Engineering (CASE), Fort Worth, TX, USA, 2016, pp. 827-834, doi: 10.1109/COASE.2016.7743488.

10. Shauri R. L. A. et al. Sensor integration and fusion for autonomous screwing task by dual-manipulator hand robot //Procedia Engineering. – 2012. – Т. 41. – С. 1412-1420.

11. Zhang L. et al. Dynamics Decoupling and Control of a 3-DOF Force-Controlled End-Effector Based on Force Sensors //IEEE Robotics and Automation Letters. – 2025.

12. Zhou P. et al. Path planning with automatic seam extraction over point cloud models for robotic arc welding //IEEE robotics and automation letters. – 2021. – Т. 6. – №. 3. – С. 5002-5009.

13. Kroemer O., Niekum S., Konidaris G. A review of robot learning for manipulation: Challenges, representations, and algorithms //Journal of machine learning research. – 2021. – Т. 22. – №. 30. – С. 1-82.

14. Li H. et al. See, hear, and feel: Smart sensory fusion for robotic manipulation //arXiv preprint arXiv:2212.03858. – 2022.

15. Юревич Е.И. Основы робототехники, 4-е изд. СПб.: БХВ-Петербург, 2018. 293 с.

16. Александр Викторович Демин. "Адаптивное управление роботами с произвольно заданной модульной конструкцией" Известия Иркутского государственного университета. Серия: Математика, vol. 29, 2019, pp. 10-21. doi:10.26516/1997-7670.2019.29.10

17. Баймухамедов, М. Ф. Системы адаптивного управления промышленным роботом / М. Ф. Баймухамедов, А. А. Жикеев // Актуальные научные исследования в современном мире. – 2021. – № 1-1(69). – С. 24-28. – EDN UGRVVKR.

18. N. V. Belov, A. G. Vovik, L. I. Voronova, M. V. Khokhlova and V. I. Voronov, "Development of a Method for Gripping Fragile Objects with a Robot Manipulator," 2025 Wave Electronics and its Application in Information and

Telecommunication Systems (WECONF), St. Petersburg, Russian Federation, 2025, pp. 1-5, doi: 10.1109/WECONF65186.2025.11017140.

19. Kondratenko Y. et al. Machine learning techniques for increasing efficiency of the robot's sensor and control information processing //Sensors. – 2022. – T. 22. – №. 3. – C. 1062.

20. Mukherjee A. et al. Cognitive intelligence in industrial robots and manufacturing //Computers & Industrial Engineering. – 2024. – T. 191. – C. 110106.

21. Pollini A. et al. Human-machine interaction design in adaptive automation //Procedia Computer Science. – 2025. – T. 253. – C. 1034-1044.

22. Arents, J.; Abolins, V.; Judvaitis, J.; Vismanis, O.; Oraby, A.; Ozols, K. Human–Robot Collaboration Trends and Safety Aspects: A Systematic Review. *J. Sens. Actuator Netws.* 2021, 10, 48. De Pace, F.; Manuri, F.; Sanna, A.; Fornaro, C. A systematic review of Augmented Reality interfaces for collaborative industrial robots. *Comput. Ind. Eng.* 2020, 149, 106806.

23. Lenz, I.; Lee, H.; Saxena, A. Deep Learning for Detecting Robotic Grasps. *Int. J. Robot. Res.* 2013, 34, 705–724.

24. Arents J., Greitans M. Smart industrial robot control trends, challenges and opportunities within manufacturing //Applied Sciences. – 2022. – T. 12. – №. 2. – C. 937.

25. Lin, W.; Anwar, A.; Li, Z.; Tong, M.; Qiu, J.; Gao, H. Recognition and Pose Estimation of Auto Parts for an Autonomous Spray Painting Robot. *IEEE Trans. Ind. Inform.* 2019, 15, 1709–1719.

26. Kah, P.; Shrestha, M.; Hiltunen, E.; Martikainen, J. Robotic arc welding sensors and programming in industrial applications. *Int. J. Mech. Mater. Eng.* 2015, 10, 13.

27. Fujita, M.; Domae, Y.; Noda, A.; Garcia Ricardez, G.; Nagatani, T.; Zeng, A.; Song, S.; Rodriguez, A.; Causo, A.; Chen, I.M.; et al. What are the important technologies for bin picking? Technology analysis of robots in competitions based on a set of performance metrics. *Adv. Robot.* 2020, 34, 560–574.

28. Holz, D.; Topalidou-Kyniazopoulou, A.; Stückler, J.; Behnke, S. Real-time object detection, localization and verification for fast robotic depalletizing. In

Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–2 October 2015; pp. 1459–1466.

29. Arents J., Greitans M. Smart industrial robot control trends, challenges and opportunities within manufacturing // *Applied Sciences*. – 2022. – Т. 12. – №. 2. – С. 937.

30. Романов А.М. Обзор аппаратно-программного обеспечения систем управления роботов различного масштаба и назначения. Часть 1. Промышленная робототехника // *Российский технологический журнал*. 2019. Т. 7. № 5. С. 30–46. <https://doi.org/10.32362/2500-316X-2019-7-5-30-46>

31. Bilancia P. et al. An overview of industrial robots control and programming approaches // *Applied Sciences*. – 2023. – Т. 13. – №. 4. – С. 2582.

32. Soltani Sharif Abadi A. et al. Review on challenges for robotic eye surgery; surgical systems, technologies, cost-effectiveness, and controllers // *The International Journal of Medical Robotics and Computer Assisted Surgery*. – 2023. – Т. 19. – №. 4. – С. e2524.

33. R. Rosenstock, T. Wrütz and R. Biesenbach, "KUKA Robot Language Parser for Product Adaptive Manufacturing," 2023 20th International Multi-Conference on Systems, Signals & Devices (SSD), Mahdia, Tunisia, 2023, pp. 164-169, doi: 10.1109/SSD58187.2023.10411226.

34. KUKA Deutschland GmbH System Software 8.7—Operating and Programming Instructions for System Integrators. — URL: [www.kuka.com](http://www.kuka.com) (дата обращения 12.01.2024).

35. ABB Robotics Technical Reference Manual—RAPID Overview. — URL: [www.abb.com/robotics](http://www.abb.com/robotics) (дата обращения 12.01.2024).

36. FANUC Robotics FANUC Robotics SYSTEM J-30iB Controller KAREL Reference Manual. — URL: [www.fanucamerica.com](http://www.fanucamerica.com) (дата обращения 12.01.2024).

37. Sun, Y.; Gray, J.; Bulheller, K.; von Baillou, N. A Model-Driven Approach to Support Engineering Changes in Industrial Robotics Software. In *Lecture Notes in Computer Science; (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*; Springer: Berlin/Heidelberg, Germany, 2012; Volume 7590, pp. 368–382.

38. Estévez, E.; Sánchez-García, A.; Gámez-García, J.; Gómez-Ortega, J.; Satorres-Martínez, S. A Novel Model-Driven Approach to Support Development Cycle of Robotic Systems. *Int. J. Adv. Manuf. Technol.* 2016, 82, 737–751.
39. Fischer, H.; Vulliez, M.; Laguillaumie, P.; Vulliez, P.; Gazeau, J.P. RTRobMultiAxisControl: A Framework for Real-Time Multiaxis and Multirobot Control. *IEEE Trans. Autom. Sci. Eng.* 2019, 16, 1205–1217
40. Lambrecht, J.; Chemnitz, M.; Kruger, J. Control Layer for Multi-Vendor Industrial Robot Interaction Providing Integration of Supervisory Process Control and Multifunctional Control Units. In *Proceedings of the 2011 IEEE Conference on Technologies for Practical Robot Applications (TePRA)*, Woburn, MA, USA, 11–12 April 2011; pp. 115–120.
41. Lewis F. L., Dawson D. M., Abdallah C. T. *Robot manipulator control: theory and practice.* – CRC Press, 2003.
42. Wen Y., Pagilla P. Path-constrained and collision-free optimal trajectory planning for robot manipulators // *IEEE Transactions on Automation Science and Engineering.* – 2022. – T. 20. – №. 2. – C. 763-774.
43. Zhang H. M., Yang X. C. Interface between LabVIEW and FANUC robot // *Advanced Materials Research.* – 2012. – T. 443. – C. 464-470.
44. Dalvand M. M., Nahavandi S. Teleoperation of ABB industrial robots // *Industrial Robot: An International Journal.* – 2014. – T. 41. – №. 3. – C. 286-295.
45. Deatcu C. et al. MATLAB/Simulink Based Rapid Control Prototyping for Multivendor Robot Applications // *Simul. Notes Eur.* – 2015. – T. 25. – №. 2. – C. 69-78.
46. Sanfilippo, F.; Hatledal, L.I.; Zhang, H.; Fago, M.; Pettersen, K.Y. Controlling Kuka Industrial Robots. *IEEE Robot. Autom. Mag.* 2015, 22, 96–109.
47. Chan, S. F., & Kwan, R. (2003). Post-processing methodologies for off-line robot programming within computer integrated manufacture. *Journal of Materials Processing Technology*, 139(1), 8–14. [https://doi.org/10.1016/S0924-0136\(03\)00174-2](https://doi.org/10.1016/S0924-0136(03)00174-2)
48. Bolmsjö, G.; Cederberg, P.; Olsson, M. Remote Control of a Standard ABB Robot System in Real Time Using the Robot Application Protocol (RAP). In *Proceedings*

of the 33rd ISR (International Symposium on Robotics), Stockholm, Sweden, 7–11 October 2002.

49. Chinello, F.; Scheggi, S.; Morbidi, F.; Prattichizzo, D. KUKA Control Toolbox. *IEEE Robot. Autom. Mag.* 2011, 18, 69–79.
50. De Coninck, E.; Verbelen, T.; Van Molle, P.; Simoens, P.; Dhoedt, B. Learning robots to grasp by demonstration. *Robot. Auton. Syst.* 2020, 127, 103474.
51. Sughashini, K.R.; Sunanthini, V.; Johnsi, J.; Nagalakshmi, R.; Sudha, R. A pneumatic robot arm for sorting of objects with chromatic sensor module. *Mater. Today Proc.* 2021, 45, 6364–6368.
52. Shaikat, A.S.; Akter, S.; Salma, U. Computer Vision Based Industrial Robotic Arm for Sorting Objects by Color and Height. *J. Eng. Adv.* 2020, 1, 116–122.
53. Chen, P.; Elangovan, V. Object Sorting using Faster R-CNN. *Int. J. Artif. Intell. Appl.* 2020, 11, 27–36.
54. Yu, Y.; Zou, S.; Yin, K. A novel detection fusion network for solid waste sorting. *Int. J. Adv. Robot. Syst.* 2020, 17, 172988142094177.
55. Li, M.; Duan, Y.; He, X.; Yang, M. Image positioning and identification method and system for coal and gangue sorting robot. *Int. J. Coal Prep. Util.* 2020, 1–19.
56. Ciszak, O.; Juszkiwicz, J.; Suszyński, M. Programming of Industrial Robots Using the Recognition of Geometric Signs in Flexible Welding Process. *Symmetry* 2020, 12, 1429.
57. Krishnan J. V. G., Manoharan N., Rani B. S. Estimation of distance to texture surface using complex log mapping // *J. Comput. Appl.* – 2010. – T. 3. – №. 3. – C. 16.
58. Tippetts, B., Lee, D.J., Lillywhite, K. et al. Review of stereo vision algorithms and their suitability for resource-limited systems. *J Real-Time Image Proc* 11, 5–25 (2016). <https://doi.org/10.1007/s11554-012-0313-2>
59. Ke, F.; Liu, H.; Zhao, D.; Sun, G.; Xu, W.; Feng, W. A high precision image registration method for measurement based on the stereo camera system. *Optik* 2020, 204, 164186.
60. Bi, Z.M.; Wang, L. Advances in 3D data acquisition and processing for industrial applications. *Robot. Comput. Integr. Manuf.* 2010, 26, 403–413

61. Laganiere, R.; Gilbert, S.; Roth, G. Robust object pose estimation from feature-based stereo. *IEEE Trans. Instrum. Meas.* 2006, 55, 1270–1280
62. Arachchi, S.P.K.; Hakim, N.L.; Hsu, H.-H.; Klimenko, S.V.; Shih, T.K. Real-Time Static and Dynamic Gesture Recognition Using Mixed Space Features for 3D Virtual World's Interactions. In *Proceedings of the 2018 32nd International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, Krakow, Poland, 16–18 May 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 627–632.
63. Jones, L. Your Body of Water: A Display that Visualizes Aesthetic Heart Rate Data from a 3D Camera. In *Proceedings of the Twelfth International Conference on Tangible, Embedded, and Embodied Interaction*, Stockholm, Sweden, 18–21 March 2018; 2018.
64. Liao, B.; Li, J.; Ju, Z.; Ouyang, G. Hand Gesture Recognition with Generalized Hough Transform and DC-CNN Using Realsense. In *Proceedings of the 2018 Eighth International Conference on Information Science and Technology (ICIST)*, Cordoba, Granada and Seville, Spain, 30 June–6 July 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 84–90.
65. Chiang, T.; Fan, C.-P. 3D Depth Information Based 2D Low-Complexity Hand Posture and Gesture Recognition Design for Human Computer Interactions. In *Proceedings of the 2018 3rd International Conference on Computer and Communication Systems (ICCCS)*, Nagoya, Japan, 27–30 April 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 233–238.
66. Karambakhsh, A.; Kamel, A.; Sheng, B.; Li, P.; Yang, P.; Feng, D.D. Deep gesture interaction for augmented anatomy learning. *Int. J. Inf. Manag.* 2018, 45, 328–336.
67. Аналитический обзор мирового рынка робототехники 2019. — URL: [https://www.tadviser.ru/images/b/bf/Sberbank\\_robotics\\_review\\_2019\\_17.07.2019\\_m.pdf](https://www.tadviser.ru/images/b/bf/Sberbank_robotics_review_2019_17.07.2019_m.pdf) (дата обращения: 30.04.24)
68. Kuka KR4 R600 [Электронный источник] URL: <https://clck.ru/34kUУe> (Дата обращения: 18.05.2022)

69. KUKA KR C5 micro [Электронный источник] URL: <https://www.kuka.com/en-us/products/robotics-systems/robot-controllers/kr-c5-micro> (Дата обращения: 04.06.2022)
70. D.S. Lavygin, V.V. Levshchanov, V.V. Prikhodko, Software interface for remote control of industrial robots KUKA //Extreme Robotics. – 2020. – Vol. 1. – No. 1. – pp. 73-77.
71. Shabarath B. B., Naidu T. Data acquisition System using TCP/IP client-server paradigm for real-time data. – ISSN:2278-621X, 2015. – С.192-195.
72. Singh A. Socket Programming with Python. – Ajit Singh, 2019
73. Sanfilippo F. et al. JOpenShowVar: An open-source cross-platform communication interface to Kuka robots //2014 IEEE International Conference on Information and Automation (ICIA). – IEEE, 2014. – С. 1154-1159.
74. N. V. Belov and A. G. Vovik, "Adaptive Control System for the Process of Sorting Objects Using a Robotic Arm," 2024 International Russian Smart Industry Conference (SmartIndustryCon), Sochi, Russian Federation, 2024, pp. 30-34, doi: 10.1109/SmartIndustryCon61328.2024.10516071.
75. Программирование промышленного робота KUKA KR AGILUS: методические указания по выполнению лабораторных работ для студентов направлений 15.03.06 «Мехатроника и робототехника», 15.04.06 «Мехатроника и робототехника» всех форм обучения / Юго-Зап. гос. ун-т; сост. С.Ф. Яцун, П.А. Безмен. Курск, 2020. – 62 с.
76. Воронов В. И. О Программе повышения квалификации "Управление промышленными манипуляторами и их применение в робототехнических комплексах" в Центре Робототехники МТУСИ / В. И. Воронов, Л. И. Воронова // Технологии информационного общества: Сборник трудов XVI Международной отраслевой научно-технической конференции, Москва, 02–03 марта 2022 года. – Москва: ООО "Издательский дом Медиа публицер", 2022. – С. 356-358.
77. Giancola, S.; Valenti, M.; Sala, R. A Survey on 3D Cameras: Metrological Comparison of Time-of-Flight, Structured-Light and Active Stereoscopy Technologies; Springer: Berlin/Heidelberg, Germany, 2018.

78. Carfagni, M.; Furferi, R.; Governi, L.; Santarelli, C.; Servi, M.; Uccheddu, F.; Volpe, Y. Metrological and Critical Characterization of the Intel D415 Stereo Depth Camera. *Sensors* 2019, 19, 489. <https://doi.org/10.3390/s19030489>
79. VDI/VDE 2634 Blatt 2 [Электронный источник] URL: [https://www.vdi.eu/guidelines/vdivde\\_2634\\_blat\\_2-optische\\_3\\_d\\_messsysteme\\_bildgebende\\_systeme\\_mit\\_flaechenhafter\\_antastung/](https://www.vdi.eu/guidelines/vdivde_2634_blat_2-optische_3_d_messsysteme_bildgebende_systeme_mit_flaechenhafter_antastung/) (Дата обращения: 04.06.2024)
80. Grunnet-Jepsen, A.; Tong, D. Depth Post-Processing FOR Intel® REALSENSE™ Depth Camera D400 Series [Электронный источник] URL: <https://dev.intelrealsense.com/docs/depth-post-processing> (Дата обращения: 04.06.2022)
81. Bradski G. et al. OpenCV //Dr. Dobb's journal of software tools. – 2000. – Т. 3. – №. 2.
82. G. Flandin, F. Chaumette and E. Marchand, "Eye-in-hand/eye-to-hand cooperation for visual servoing," Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065), San Francisco, CA, USA, 2000, pp. 2741-2746 vol.3, doi: 10.1109/ROBOT.2000.846442.
83. Ding L., Goshtasby A. On the Canny edge detector //Pattern recognition. – 2001. – Т. 34. – №. 3. – С. 721-725.
84. Deng G., Cahill L. W. An adaptive Gaussian filter for noise reduction and edge detection //1993 IEEE conference record nuclear science symposium and medical imaging conference. – IEEE, 1993. – С. 1615-1619.
85. Kanopoulos N., Vasanthavada N., Baker R. L. Design of an image edge detection filter using the Sobel operator //IEEE Journal of solid-state circuits. – 1988. – Т. 23. – №. 2. – С. 358-367.
86. Shet Reshma Prakash, Paras Nath Singh, Object detection through region proposal based techniques, *Materials Today: Proceedings*, Volume 46, Part 9, 2021, Pages 3997-4002, ISSN 2214-7853, <https://doi.org/10.1016/j.matpr.2021.02.533>. (Дата обращения: 22.01.2024)

87. Liu, W. et al. (2016). SSD: Single Shot MultiBox Detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds) Computer Vision – ECCV 2016. ECCV 2016. Lecture Notes in Computer Science, vol 9905. Springer, Cham. [https://doi.org/10.1007/978-3-319-46448-0\\_2](https://doi.org/10.1007/978-3-319-46448-0_2)
88. J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016 pp. 779-788. doi: 10.1109/CVPR.2016.91
89. Ерохин Д. Ю., Ершов М. Д. Современные сверточные нейронные сети для обнаружения и распознавания объектов // Цифровая обработка сигналов. – 2018. – №. 3. – С. 64.
90. Sapkota, Ranjan & Ahmed, Dawood & Karkee, Manoj. (2023). Comparing YOLOv8 and Mask RCNN for object segmentation in complex orchard environments. 10.32388/ZB9SB0.
91. Srivastava, S., Divekar, A.V., Anilkumar, C. et al. Comparative analysis of deep learning image detection algorithms. J Big Data 8, 66 (2021). <https://doi.org/10.1186/s40537-021-00434-w>
92. Lin T. Y. et al. Microsoft coco: Common objects in context //Computer vision–ECCV 2014: 13th European conference, zurich, Switzerland, September 6-12, 2014, proceedings, part v 13. – Springer International Publishing, 2014. – С. 740-755.
93. Jocher G. et al. ultralytics/yolov5: v6. 1-tensorrt, tensorflow edge tpu and openvino export and inference //Zenodo. – 2022.
94. Kukartsev V. V. et al. Deep Learning for Object Detection in Images Development and Evaluation of the YOLOv8 Model Using Ultralytics and Roboflow Libraries //Computer Science On-line Conference. – Cham : Springer Nature Switzerland, 2024. – С. 629-637.
95. Wang G. et al. UAV-YOLOv8: A small-object-detection model based on improved YOLOv8 for UAV aerial photography scenarios //Sensors. – 2023. – Т. 23. – №. 16. – С. 7190.

96. Zheng, Z.; Wang, P.; Liu, W.; Li, J.; Ye, R.; Ren, D. Distance-IoU loss: Faster and better learning for bounding box regression. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 12993–13000.
97. Li, X.; Wang, W.; Wu, L.; Chen, S.; Hu, X.; Li, J.; Tang, J.; Yang, J. Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection. *Adv. Neural Inf. Process. Syst.* 2020, 33, 21002–21012.
98. Terven, J.; Córdova-Esparza, D.-M.; Romero-González, J.-A. A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS. *Mach. Learn. Knowl. Extr.* 2023, 5, 1680-1716. <https://doi.org/10.3390/make5040083>
99. Nazarkevych M. et al. A YOLO-based Method for Object Contour Detection and Recognition in Video Sequences //CPITS. – 2024. – C. 49-58.
100. Kevin, P., Gokul, C., Ravisankar, S., Shubham, S., Vamsi, K., Vivek, M., Santosh, T., & Deebul, N. (2023). RoboCup @work 2023 dataset [Data set]. Zenodo. <https://doi.org/10.5281/zenodo.10003915>
101. Thongprasith J. et al. Portioning Algorithm Using the Bisection Method for Slicing Food //2023 5th International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA). – IEEE, 2023. – C. 1-6.
102. Jo, HyunJun, and Jae-Bok Song. 2020. "Object-Independent Grasping in Heavy Clutter" *Applied Sciences* 10, no. 3: 804. <https://doi.org/10.3390/app10030804>.
103. N. V. Belov, L. I. Voronova and M. V. Khokhlova, "Development of a Method for Robotic Sorting of Metal Objects," 2024 Intelligent Technologies and Electronic Devices in Vehicle and Road Transport Complex (TIRVED), Moscow, Russian Federation, 2024, pp. 1-6, doi: 10.1109/TIRVED63561.2024.10769969.
104. Wilczkowiak M., Boyer E., Sturm P. Camera calibration and 3D reconstruction from single images using parallelepipeds //Proceedings eighth IEEE international conference on computer vision. ICCV 2001. – IEEE, 2001. – T. 1. – C. 142-148.

105. Z. Zhang, "A flexible new technique for camera calibration," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330-1334, Nov. 2000, doi: 10.1109/34.888718
106. Andriyanov, N.; Khasanshin, I.; Utkin, D.; Gataullin, T.; Ignar, S.; Shumaev, V.; Soloviev, V. Intelligent System for Estimation of the Spatial Position of Apples Based on YOLOv3 and Real Sense Depth Camera D415. *Symmetry* 2022, 14, 148. <https://doi.org/10.3390/sym14010148>
107. Shi H., Wang M., Wang C. Pattern-based autonomous smooth switching control for constrained flexible joint manipulator // *Neurocomputing*. – 2022. – T. 492. – C. 162-173.
108. Krull, A., Brachmann, E., Michel, F., Ying Yang, M., Gumhold, S., Rother, C.: Learning analysis-by-synthesis for 6D pose estimation in RGB-D images. In: *IEEE International Conference on Computer Vision (ICCV)*. pp. 954–962 (2015)
109. Xiang, Y., Schmidt, T., Narayanan, V., and Fox, D. (2017). PoseCNN: a convolutional neural network for 6D object pose estimation in cluttered scenes. *arXiv preprint arXiv:1711.00199*. doi: 10.15607/rss.2018.xiv.019
110. Kehl, W., Manhardt, F., Tombari, F., Ilic, S., and Navab, N. (2017). "SSD-6D: making rgb-based 3D detection and 6D pose estimation great again," in *Proceedings of the IEEE International Conference on Computer Vision*, 1521–1529.89
111. Do, T.-T., Cai, M., Pham, T., and Reid, I. (2018). Deep-6DPose: recovering 6D object pose from a single RGB image. *arXiv preprint arXiv:1802.10367*. doi: 10.48550/arXiv.1802.10367
112. Roovere, Peter & Moonen, Steven & Michiels, Nick & wyffels, Francis. (2022). Dataset of Industrial Metal Objects. 10.48550/arXiv.2208.04052.
113. Mian, Ajmal & Bennamoun, Mohammed & A.owens, R.. (2011). Automatic correspondence for 3D modeling: An extensive review. *International Journal of Shape Modeling*. 11. 10.1142/S0218654305000797.
114. Eric Brachmann, Alexander Krull, Frank Michel, Stefan Gumhold, Jamie Shotton, and Carsten Rother. Learning 6d object pose estimation using 3d object coordinates. In *European conference on computer vision*, pages 536–551. Springer, 2014.

115. Szilvsi-Nagy M., Matyasi G. Y. Analysis of STL files //Mathematical and computer modelling. – 2003. – Т. 38. – №. 7-9. – С. 945-960.
116. Tymchyshyn Beginner's guide to mapping simplexes affinely. / Tymchyshyn, & K. Vitalii, Andrii. — Kiev : Bogolyubov Institute for Theoretical Physics, National Academy of Sciences, 2019. — 16 с.
117. Кудрина М. А., Мурзин А. В. Аффинные преобразования объектов в компьютерной графике // Труды Международного симпозиума «Надежность и качество». – 2014. – Т. 1. – С. 307-310.
118. S. Rusinkiewicz and M. Levoy, "Efficient variants of the ICP algorithm," Proceedings Third International Conference on 3-D Digital Imaging and Modeling, Quebec City, QC, Canada, 2001, pp. 145-152, doi: 10.1109/IM.2001.924423.
119. Фу К., Гонсалес Р., Ли К. Робототехника, 1989 г. М., издательство "Мир", твердый переплет, стр. 29-45
120. Мингазова И.Н. Обзор различных захватов для промышленных роботов манипуляторов // Форум молодых ученых. 2018. №7 (23). URL: <https://cyberleninka.ru/article/n/obzor-razlichnyh-zahvatov-dlya-promyshlennyh-robotov-manipulyatorov> (дата обращения: 14.11.2024).
121. Авторское свидетельство СССР № 903108, кл. В 25 J 19/00, 1982.
122. Авторское свидетельство СССР № 2823813, кл. В 25 J 15/06, 1979
123. Авторское свидетельство СССР MS 1142409, кл. В 66 С 1/02,1/06, 1985.
124. Соболев С.Ф., Кузьмин Ю.П. Методические указания по разработке технологических процессов изготовления деталей механической обработкой. – СПб: СПбГУ ИТМО, 2007. – 118с.
125. Jancarczyk D., Rysiński J., Worek J. Comparative Analysis of Measurement Tools in the Cognex D900 Vision System //Applied Sciences. – 2024. – Т. 14. – №. 18. – С. 8296.
126. Carfagni, M.; Furferi, R.; Governi, L.; Santarelli, C.; Servi, M.; Uccheddu, F.; Volpe, Y. Metrological and Critical Characterization of the Intel D415 Stereo Depth Camera. Sensors 2019, 19, 489. <https://doi.org/10.3390/s19030489>

127. Roy D. Development of novel magnetic grippers for use in unstructured robotic workspace //Robotics and Computer-Integrated Manufacturing. – 2015. – T. 35. – C. 16-41.

## Приложение А. Результаты экспериментов преобразования координат центра объекта

Таблица А.1 - Результаты эксперимента геометрического способа преобразования координат из пикселей в миллиметры на шахматной доске

№ замера	Координаты пересечения ячеек шахматной доски по Оси X, пикс	Координаты пересечения ячеек шахматной доски по Оси Y, пикс	Координаты пересечения ячеек шахматной доски по Оси X, мм (Эталонные значения)	Координаты пересечения ячеек шахматной доски по Оси Y, мм (Эталонные значения)	Координаты пересечения ячеек шахматной доски по Оси X, мм (Измеренные предложенным способом)	Координаты пересечения ячеек шахматной доски по Оси Y, мм (Измеренные предложенным способом)	Погрешность измерений по Оси X, мм	Погрешность измерений по Оси Y, мм
1	72.48535	399.11005	0	0	0	0	0	0
2	72.10861	337.25537	0	58	0	58	0	0
3	72.80313	274.92706	0	116	0	116	0	0
4	72.646126	213.08173	0	174	0	174	0	0
5	72.61649	151.1312	0	232	0	232	0	0
6	72.615585	89.335495	0	290	0	290	0	0
7	134.30064	399.44104	58	0	58	0	0	0
8	135.44383	336.64844	58	58	60	59	-2	-1
9	135.38756	274.86606	58	116	60	116	-2	0
10	135.59813	213.3123	58	174	60	174	-2	0
11	135.3938	151.47981	58	232	60	232	-2	0
12	134.58604	89.305145	58	290	59	290	-1	0
13	197.2927	398.42398	116	0	118	1	-2	-1
14	197.37509	336.55856	116	58	118	59	-2	-1
15	197.85509	274.69946	116	116	119	117	-3	-1
16	197.8235	213.43832	116	174	119	174	-3	0
17	197.522	151.80388	116	232	118	231	-2	1
18	196.7575	89.63314	116	290	118	290	-2	0
19	259.2734	398.0063	174	0	177	1	-3	-1
20	259.41528	336.11816	174	58	177	59	-3	-1
21	259.30975	274.70172	174	116	177	117	-3	-1
22	259.40997	213.48875	174	174	177	174	-3	0
23	259.35434	152.19365	174	232	177	231	-3	1
24	258.7068	90.04193	174	290	176	289	-2	1

№ замера	Координаты пересечения ячеек шахматной доски по Оси X, пикс	Координаты пересечения ячеек шахматной доски по Оси Y, пикс	Координаты пересечения ячеек шахматной доски по Оси X, мм (Эталонные значения)	Координаты пересечения ячеек шахматной доски по Оси Y, мм (Эталонные значения)	Координаты пересечения ячеек шахматной доски по Оси X, мм (Измеренные предложенным способом)	Координаты пересечения ячеек шахматной доски по Оси Y, мм (Измеренные предложенным способом)	Погрешность измерений по Оси X, мм	Погрешность измерений по Оси Y, мм
25	320.5733	397.96677	232	0	235	1	-3	-1
26	320.56177	336.10434	232	58	235	59	-3	-1
27	320.53265	274.64584	232	116	235	117	-3	-1
28	320.5153	213.33496	232	174	235	174	-3	0
29	320.47873	152.28473	232	232	235	231	-3	1
30	320.4126	90.09493	232	290	234	289	-2	1
31	382.42606	398.0624	290	0	293	1	-3	-1
32	381.9375	336.18985	290	58	293	59	-3	-1
33	381.9016	274.65585	290	116	293	117	-3	-1
34	381.68958	213.42177	290	174	292	174	-2	0
35	381.5975	152.19096	290	232	292	231	-2	1
36	381.8292	90.20003	290	290	293	289	-3	1
37	444.2438	398.48584	348	0	352	1	-4	-1
38	443.58804	336.40857	348	58	351	59	-3	-1
39	443.38794	274.61646	348	116	351	117	-3	-1
40	443.39294	213.31328	348	174	351	174	-3	0
41	443.19116	151.84285	348	232	351	231	-3	1
42	443.55673	89.77634	348	290	351	290	-3	0
43	506.74142	399.25366	406	0	411	0	-5	0
44	505.75778	336.61697	406	58	410	59	-4	-1
45	505.38852	274.73108	406	116	409	116	-3	0
46	505.20132	213.21382	406	174	409	174	-3	0
47	505.32608	151.46587	406	232	409	232	-3	0
48	505.41043	89.58525	406	290	409	290	-3	0
49	569.51776	399.5566	464	0	470	0	-6	0
50	569.27527	337.2643	464	58	470	58	-6	0
51	567.66095	274.74817	464	116	468	116	-4	0
52	567.5805	212.93079	464	174	468	174	-4	0
53	567.4173	151.25043	464	232	468	232	-4	0
54	567.3634	89.40337	464	290	468	290	-4	0

Таблица А.2 - Результаты эксперимента способа преобразования координат из пикселей в миллиметры, через фокусное расстояние, на шахматной доске

№ замера	Координаты пересечения ячеек шахматной доски по Оси Х, пикс	Координаты пересечения ячеек шахматной доски по Оси Y, пикс	Координаты пересечения ячеек шахматной доски по Оси Х, мм (Эталонные значения)	Координаты пересечения ячеек шахматной доски по Оси Y, мм (Эталонные значения)	Координаты пересечения ячеек шахматной доски по Оси Х, мм (Измеренные предложенным способом)	Координаты пересечения ячеек шахматной доски по Оси Y, мм (Измеренные предложенным способом)	Погрешность измерений по Оси Х, мм	Погрешность измерений по Оси Y, мм
1	72.48535	399.11005	0	0	-18	-11	18	11
2	72.10861	337.25537	0	58	-17	51	17	7
3	72.80313	274.92706	0	116	-17	114	17	2
4	72.646126	213.08173	0	174	-19	176	19	-2
5	72.61649	151.1312	0	232	-17	238	17	-6
6	72.615585	89.335495	0	290	-18	301	18	-11
7	134.30064	399.44104	58	0	46	-11	12	11
8	135.44383	336.64844	58	58	47	52	11	6
9	135.38756	274.86606	58	116	47	114	11	2
10	135.59813	213.3123	58	174	47	176	11	-2
11	135.3938	151.47981	58	232	46	238	12	-6
12	134.58604	89.305145	58	290	45	300	13	-10
13	197.2927	398.42398	116	0	109	-10	7	10
14	197.37509	336.55856	116	58	109	52	7	6
15	197.85509	274.69946	116	116	108	114	8	2
16	197.8235	213.43832	116	174	109	175	7	-1
17	197.522	151.80388	116	232	109	238	7	-6
18	196.7575	89.63314	116	290	108	300	8	-10
19	259.2734	398.0063	174	0	171	-8	3	9
20	259.41528	336.11816	174	58	171	53	3	5
21	259.30975	274.70172	174	116	171	114	3	2
22	259.40997	213.48875	174	174	171	176	3	-2
23	259.35434	152.19365	174	232	171	236	3	-4
24	258.7068	90.04193	174	290	170	300	4	-10
25	320.5733	397.96677	232	0	232	-8	0	9
26	320.56177	336.10434	232	58	232	53	0	5
27	320.53265	274.64584	232	116	232	114	0	2
28	320.5153	213.33496	232	174	232	175	0	-1

№ замера	Координаты пересечения ячеек шахматной доски по Оси Х, пикс	Координаты пересечения ячеек шахматной доски по Оси Y, пикс	Координаты пересечения ячеек шахматной доски по Оси Х, мм (Эталонные значения)	Координаты пересечения ячеек шахматной доски по Оси Y, мм (Эталонные значения)	Координаты пересечения ячеек шахматной доски по Оси Х, мм (Измеренные предложенным способом)	Координаты пересечения ячеек шахматной доски по Оси Y, мм (Измеренные предложенным способом)	Погрешность измерений по Оси Х, мм	Погрешность измерений по Оси Y, мм
29	320.47873	152.28473	232	232	232	237	0	-5
30	320.4126	90.09493	232	290	232	299	0	-9
31	382.42606	398.0624	290	0	294	-9	-4	9
32	381.9375	336.18985	290	58	293	52	-3	6
33	381.9016	274.65585	290	116	293	114	-3	2
34	381.68958	213.42177	290	174	293	175	-3	-1
35	381.5975	152.19096	290	232	293	236	-3	-4
36	381.8292	90.20003	290	290	293	299	-3	-9
37	444.2438	398.48584	348	0	356	-10	-8	10
38	443.58804	336.40857	348	58	355	52	-7	6
39	443.38794	274.61646	348	116	356	114	-8	2
40	443.39294	213.31328	348	174	355	175	-7	-1
41	443.19116	151.84285	348	232	355	238	-7	-6
42	443.55673	89.77634	348	290	355	300	-7	-10
43	506.74142	399.25366	406	0	418	-10	-12	10
44	505.75778	336.61697	406	58	417	52	-11	6
45	505.38852	274.73108	406	116	419	114	-13	2
46	505.20132	213.21382	406	174	418	176	-12	-2
47	505.32608	151.46587	406	232	418	238	-12	-6
48	505.41043	89.58525	406	290	419	301	-13	-11
49	569.51776	399.5566	464	0	482	-11	-18	11
50	569.27527	337.2643	464	58	481	51	-17	7
51	567.66095	274.74817	464	116	480	114	-16	2
52	567.5805	212.93079	464	174	481	177	-17	-3
53	567.4173	151.25043	464	232	480	238	-16	-6
54	567.3634	89.40337	464	290	481	301	-17	-11

Таблица А.3 - Результаты эксперимента геометрического способа преобразования координат из пикселей в миллиметры на ферромагнитной заготовке

№ замера	Координаты центра объекта по Оси X, пикс	Координаты центра объекта по Оси Y, пикс	Координаты центра объекта по Оси X, мм (Эталонные значения)	Координаты центра объекта по Оси Y, мм (Эталонные значения)	Координаты центра объекта по Оси X, мм (Измеренные предложенным способом)	Координаты центра объекта по Оси Y, мм (Измеренные предложенным способом)	Погрешность измерений по Оси X, мм	Погрешность измерений по Оси Y, мм
1	68	399	0	0	-4	0	4	0
2	70	337	0	58	-2	58	2	0
3	70	274	0	116	-2	117	2	-1
4	134	401	0	174	-1	174	1	0
5	132	338	0	232	-1	232	1	0
6	132	273	0	290	-3	290	3	0
7	194	399	58	0	58	-2	0	2
8	195	337	58	58	56	57	2	1
9	193	274	58	116	56	118	2	-2
10	255	399	58	174	57	175	1	-1
11	256	337	58	232	56	233	2	-1
12	257	276	58	290	57	287	1	3
13	317	399	116	0	115	0	1	0
14	319	340	116	58	116	58	0	0
15	317	275	116	116	114	117	2	-1
16	380	400	116	174	117	175	-1	-1
17	380	338	116	232	116	229	0	3
18	380	276	116	290	116	288	0	2
19	442	401	174	0	173	0	1	0
20	441	338	174	58	174	58	0	0
21	441	277	174	116	175	115	-1	1
22	503	400	174	174	175	174	-1	0
23	503	340	174	232	175	230	-1	2
24	504	280	174	290	175	285	-1	5
25	567	402	232	0	231	0	1	0
26	567	339	232	58	233	55	-1	3
27	568	276	232	116	231	116	1	0
28	71	213	232	174	232	172	0	2

№ замера	Координаты центра объекта по Оси X, пикс	Координаты центра объекта по Оси Y, пикс	Координаты центра объекта по Оси X, мм (Эталонные значения)	Координаты центра объекта по Оси Y, мм (Эталонные значения)	Координаты центра объекта по Оси X, мм (Измеренные предложенным способом)	Координаты центра объекта по Оси Y, мм (Измеренные предложенным способом)	Погрешность измерений по Оси X, мм	Погрешность измерений по Оси Y, мм
29	71	151	232	232	231	230	1	2
30	69	89	232	290	231	289	1	1
31	133	212	290	0	291	-1	-1	1
32	132	150	290	58	291	57	-1	1
33	133	92	290	116	291	115	-1	1
34	196	212	290	174	289	174	1	0
35	195	154	290	232	290	230	0	2
36	195	91	290	290	290	289	0	1
37	257	213	348	0	349	-2	-1	2
38	258	153	348	58	348	57	0	1
39	258	95	348	116	348	114	0	2
40	318	215	348	174	348	174	0	0
41	317	153	348	232	348	228	0	4
42	317	90	348	290	348	286	0	4
43	378	213	406	0	407	-1	-1	1
44	379	153	406	58	407	55	-1	3
45	379	90	406	116	408	112	-2	4
46	440	213	406	174	407	173	-1	1
47	440	155	406	232	407	230	-1	2
48	441	94	406	290	409	286	-3	4
49	503	214	464	0	468	-3	-4	3
50	503	153	464	58	468	56	-4	2
51	505	94	464	116	469	115	-5	1
52	566	212	464	174	467	175	-3	-1
53	565	154	464	232	466	229	-2	3
54	566	93	464	290	467	286	-3	4

Таблица А.4 - Результаты эксперимента способа преобразования координат из пикселей в миллиметры, через фокусное расстояние, на ферромагнитной заготовке

№ замера	Координаты центра объекта по Оси X, пикс	Координаты центра объекта по Оси Y, пикс	Координаты центра объекта по Оси X, мм (Эталонные значения)	Координаты центра объекта по Оси Y, мм (Эталонные значения)	Координаты центра объекта по Оси X, мм (Измеренные предложенным способом)	Координаты центра объекта по Оси Y, мм (Измеренные предложенным способом)	Погрешность измерений по Оси X, мм	Погрешность измерений по Оси Y, мм
1	68	399	0	0	-9,427133883	8,464845522	9,427133883	-8,464845522
2	70	337	0	58	-8,240687262	55,71872776	8,240687262	2,281272242
3	70	274	0	116	-9,894667897	116,0783664	9,894667897	-0,078366398
4	134	401	0	174	-7,161312731	174,5179902	7,161312731	-0,517990208
5	132	338	0	232	-7,279724513	234,5879714	7,279724513	-2,587971438
6	132	273	0	290	-9,616799151	294,4609856	9,616799151	-4,46098563
7	194	399	58	0	55,02901744	0,034987305	2,970982564	-0,034987305
8	195	337	58	58	54,13753841	56,2169385	3,862461586	1,783061497
9	193	274	58	116	55,66745396	118,9037675	2,332546039	-2,903767533
10	255	399	58	174	52,8256109	176,2973143	5,174389102	-2,297314296
11	256	337	58	232	50,4061581	235,996534	7,593841899	-3,996534042
12	257	276	58	290	55,70220409	288,6314234	2,297795905	1,36857664
13	317	399	116	0	110,7102921	-4,167527915	5,28970794	4,167527915
14	319	340	116	58	110,5630547	62,46195226	5,436945296	-4,461952258
15	317	275	116	116	109,1175087	116,0783664	6,882491292	-0,078366398
16	380	400	116	174	110,2749247	173,7946743	5,725075339	0,205325724
17	380	338	116	232	113,3268894	230,7065621	2,673110575	1,293437853
18	380	276	116	290	111,6729088	292,5343501	4,32709121	-2,534350058
19	442	401	174	0	169,2148951	-4,693876808	4,785104912	4,693876808
20	441	338	174	58	170,9218031	56,36093962	3,078196896	1,639060384
21	441	277	174	116	171,1467445	114,2013864	2,85325553	1,798613617
22	503	400	174	174	171,0425437	175,1436502	2,95745631	-1,143650213
23	503	340	174	232	173,0355904	231,8006396	0,964409645	0,199360436
24	504	280	174	290	175,1890731	281,961027	-1,189073142	8,038972991
25	567	402	232	0	229,2361984	2,41183325	2,763801641	-2,41183325
26	567	339	232	58	231,0754248	56,47514739	0,924575175	1,524852606
27	568	276	232	116	229,1022259	115,1680145	2,897774073	0,831985461
28	71	213	232	174	230,0780745	173,041565	1,921925498	0,958434989

№ замера	Координаты центра объекта по Оси X, пикс	Координаты центра объекта по Оси Y, пикс	Координаты центра объекта по Оси X, мм (Эталонные значения)	Координаты центра объекта по Оси Y, мм (Эталонные значения)	Координаты центра объекта по Оси X, мм (Измеренные предложенным способом)	Координаты центра объекта по Оси Y, мм (Измеренные предложенным способом)	Погрешность измерений по Оси X, мм	Погрешность измерений по Оси Y, мм
29	71	151	232	232	229,1121498	232,8086473	2,887850189	-0,80864735
30	69	89	232	290	229,1022259	293,9942234	2,897774073	-3,994223404
31	133	212	290	0	290,1539591	-6,190164227	-0,153959133	6,190164227
32	132	150	290	58	290,0547203	54,10823256	-0,054720295	3,89176744
33	133	92	290	116	285,0927784	117,1211331	4,90722161	-1,121133074
34	196	212	290	174	287,9277012	175,0542702	2,072298802	-1,054270212
35	195	154	290	232	288,9895568	233,0966496	1,010443234	-1,096649574
36	195	91	290	290	286,963447	286,1182233	3,036552972	3,881776742
37	257	213	348	0	347,219599	-3,162830499	0,780400991	3,162830499
38	258	153	348	58	345,6714897	60,92097484	2,328510326	-2,920974838
39	258	95	348	116	346,0750444	114,0921442	1,924955591	1,90785584
40	318	215	348	174	347,9109629	175,0989602	0,089037086	-1,098960213
41	317	153	348	232	342,1551103	227,0833617	5,844889696	4,916638253
42	317	90	348	290	344,6741228	285,052913	3,325877189	4,94708704
43	378	213	406	0	402,1052924	0,165746935	3,894707593	-0,165746935
44	379	153	406	58	404,5267201	54,65444368	1,473279943	3,345556324
45	379	90	406	116	400,60017	112,321096	5,399829971	3,678904003
46	440	213	406	174	400,2892217	172,9273572	5,71077833	1,072642768
47	440	155	406	232	404,8293985	231,2246351	1,170601487	0,775364886
48	441	94	406	290	402,740421	283,8446278	3,259579029	6,155372235
49	503	214	464	0	472,2175315	-8,666321285	-8,217531533	8,666321285
50	503	153	464	58	469,7663322	53,63153922	-5,766332232	4,368460777
51	505	94	464	116	473,5010867	116,8231997	-9,501086665	-0,823199738
52	566	212	464	174	473,7491838	174,2581261	-9,749183761	-0,258126132
53	565	154	464	232	474,1940715	230,5642162	-10,19407147	1,43578378
54	566	93	464	290	474,3280439	287,93128	-10,3280439	2,068719979

Таблица А.5 - Результаты эксперимента способа преобразования координат из пикселей в миллиметры, через фокусное расстояние при предложенной модификации, на ферромагнитной заготовке

№ замера	Координаты центра объекта по Оси X, пикс	Координаты центра объекта по Оси Y, пикс	Координаты центра объекта по Оси X, мм (Эталонные значения)	Координаты центра объекта по Оси Y, мм (Эталонные значения)	Координаты центра объекта по Оси X, мм (Измеренные предложенным способом)	Координаты центра объекта по Оси Y, мм (Измеренные предложенным способом)	Погрешность измерений по Оси X, мм	Погрешность измерений по Оси Y, мм
1	343	316	0	0	-0,180515439	-0,650482348	0,180515439	0,650482348
2	361	304	0	58	-0,402148629	58,24266141	0,402148629	-0,242661409
3	363	307	0	116	-0,188784752	115,8033851	0,188784752	0,196614862
4	352	307	0	174	-0,043234812	174,8146369	0,043234812	-0,814636926
5	357	304	0	232	0,019616454	232,2694213	-0,019616454	-0,269421312
6	350	309	0	290	-1,526855552	290,9738936	1,526855552	-0,973893562
7	358	303	58	0	59,09153589	-0,239996558	-1,091535891	0,239996558
8	348	312	58	58	58,09914703	58,74087231	-0,099147032	-0,740872309
9	358	305	58	116	57,79481395	116,6922195	0,205186052	-0,692219456
10	356	302	58	174	58,6929258	175,0248494	-0,692925796	-1,024849357
11	354	304	58	232	58,10245481	232,3091522	-0,102454807	-0,309152153
12	356	305	58	290	58,18846311	289,1150919	-0,188463106	0,884908147
13	355	309	116	0	115,6411298	-0,673655315	0,358870215	0,673655315
14	354	305	116	58	115,8015713	57,2263783	0,198428738	0,773621702
15	355	305	116	116	114,1112	117,5247777	1,888799951	-1,524777745
16	354	300	116	174	117,0304769	175,137402	-1,030476917	-1,137401971
17	354	304	116	232	114,9828481	230,816169	1,017151873	1,183831002
18	359	308	116	290	115,8363068	288,6467017	0,16369323	1,353298256
19	354	303	174	0	172,3263521	-0,435308343	1,673647914	0,435308343
20	360	310	174	58	171,9194738	58,99080516	2,080526166	-0,990805162
21	356	304	174	116	174,0200329	115,9705549	-0,020032874	0,029445142
22	358	306	174	174	173,7471248	174,2038696	0,25287515	-0,203869582
23	355	305	174	232	174,0630295	232,0972909	-0,063029485	-0,097290926
24	361	303	174	290	173,1748481	289,4163519	0,825151903	0,58364809
25	345	298	232	0	231,2014445	0,200282746	0,798555494	-0,200282746
26	349	301	232	58	232,0730988	58,13673002	-0,073098834	-0,136730021
27	368	308	232	116	231,0095832	116,9371849	0,990416817	-0,937184858
28	387	353	232	174	232,1508412	173,3382119	-0,150841247	0,661788074

№ замера	Координаты центра объекта по Оси X, пикс	Координаты центра объекта по Оси Y, пикс	Координаты центра объекта по Оси X, мм (Эталонные значения)	Координаты центра объекта по Оси Y, мм (Эталонные значения)	Координаты центра объекта по Оси X, мм (Измеренные предложенным способом)	Координаты центра объекта по Оси Y, мм (Измеренные предложенным способом)	Погрешность измерений по Оси X, мм	Погрешность измерений по Оси Y, мм
29	395	358	232	232	231,3999317	231,1521834	0,600068271	0,847816609
30	397	361	232	290	231,28249	289,9062933	0,717510012	0,093706665
31	398	356	290	0	291,2078712	0,213524325	-1,207871176	-0,213524325
32	396	361	290	58	291,1648635	56,52623686	-1,164863517	1,473763144
33	386	353	290	116	290,9928464	115,7653122	-0,992846437	0,234687797
34	371	351	290	174	289,5836631	174,6391933	0,416336873	-0,63919329
35	384	353	290	232	290,1212103	231,6321789	-0,121210265	0,367821126
36	389	361	290	290	289,3868296	289,8698724	0,613170398	0,130127594
37	386	355	348	0	348,1048082	0,643872248	-0,104808151	-0,643872248
38	383	353	348	58	347,6731025	57,39355192	0,32689746	0,606448077
39	377	339	348	116	349,1865051	116,4042186	-1,186505086	-0,404218604
40	385	353	348	174	347,4696617	174,6838703	0,530338319	-0,68387029
41	385	356	348	232	346,9569513	232,4399071	1,043048705	-0,439907078
42	385	361	348	290	347,9030277	289,8847714	0,096972289	0,11522863
43	384	354	406	0	407,2246806	0,422077529	-1,224680577	-0,422077529
44	383	356	406	58	406,6656545	55,99326351	-0,665654472	2,006736493
45	395	327	406	116	408,3708845	115,9192475	-2,370884535	0,080752497
46	380	354	406	174	406,5432403	173,7470384	-0,543240293	0,252961551
47	390	343	406	232	405,4367424	231,8589447	0,563257596	0,141055294
48	387	344	406	290	405,9693117	290,2505705	0,030688289	-0,250570494
49	393	352	464	0	464,5069958	-0,226755541	-0,50699579	0,226755541
50	382	352	464	58	465,9740739	59,06528829	-1,974073914	-1,065288288
51	387	341	464	116	468,2830484	115,242271	-4,283048448	0,757728984
52	401	350	464	174	464,3184368	175,9418251	-0,31843677	-1,941825122
53	379	352	464	232	463,2284608	231,1985137	0,77153918	0,801486294
54	381	350	464	290	464,4110565	289,6149661	-0,411056543	0,385033878

## Приложение Б. Результаты экспериментов определения пространственных координат объекта

Таблица Б.1 - Тестирование алгоритма определения углов наклона объекта в пространстве

№	Класс	Наклон заготовки относительно Оси X камеры, ° (Эталонные значения)	Наклон заготовки относительно Оси Y камеры, ° (Эталонные значения)	Наклон заготовки относительно Оси Z камеры, ° (Эталонные значения)	Наклон относительно Оси Z объекта измеренный с помощью FindContours, °	Наклон заготовки относительно Оси X камеры, измеренный без поворота робота к маске сегментации, °	Наклон заготовки относительно Оси Y камеры, измеренный без поворота робота к маске сегментации, °	Наклон заготовки относительно Оси Z камеры, измеренный без поворота робота к маске сегментации, °	Наклон заготовки относительно Оси X камеры, измеренный под прямым углом к маске сегментации, °	Наклон заготовки относительно Оси Y камеры, измеренный под прямым углом к маске сегментации, °	Наклон заготовки относительно Оси Z камеры, измеренный под прямым углом к маске сегментации, °
1	Square	0,60	0,40	90,45	90	0,82	-2,69	0,17	1,92	1,25	-0,19
2	Square	2,15	0,60	41,25	41	4,53	-1,40	50,92	0,90	-0,44	1,80
3	Square	1,25	2,15	132,65	133	-8,08	-10,58	-41,24	-1,03	1,17	0,72
4	Square	6,60	11,20	135,78	136	5,49	32,16	-60,40	5,45	9,49	-0,08
5	Square	1,60	11,60	90,23	90	-8,91	10,13	0,31	-1,76	10,66	0,57
6	Square	0,30	7,40	63,98	64	0,44	5,14	29,43	3,50	9,12	-0,78
7	Square	0,80	-13,80	56,23	56	-1,71	-7,23	35,10	2,69	-17,42	0,28
8	Square	1,35	-12,00	90,76	91	-6,59	-11,29	-0,22	-2,58	-10,66	-1,04
9	Square	2,15	-21,00	123,86	124	4,06	-19,90	-53,22	-2,59	-22,72	-0,98
10	Hexagon	2,85	1,35	142,98	143	4,10	-11,04	-50,13	1,28	-5,53	-3,52
11	Hexagon	2,00	0,30	94,46	94	-6,92	-3,93	-5,39	-3,53	-1,45	-2,05
12	Hexagon	1,70	1,90	58,84	59	-3,32	0,27	30,15	-5,79	-1,79	-1,00
13	Hexagon	0,10	13,10	130,32	130	-2,31	-3,24	-36,34	-2,80	9,58	-1,12
14	Hexagon	1,20	15,10	90,90	91	0,92	14,81	99,41	-1,19	16,60	1,29
15	Hexagon	0,45	15,40	50,44	50	6,16	15,85	39,64	-1,67	15,54	1,54
16	Hexagon	0,25	-20,00	113,89	114	-1,92	-24,06	-26,77	-1,38	-19,15	-1,98
17	Hexagon	0,35	-21,30	96,12	96	-7,73	-22,13	-5,49	-5,80	-19,79	-1,71
18	Hexagon	0,45	-13,70	45,54	46	-5,78	-12,79	44,49	-6,34	-13,41	0,48
19	Circle	-	2,27	61,97	62	-	-4,96	29,94	-	-2,68	-0,79
20	Circle	-	0,30	95,24	95	-	0,24	-3,40	-	0,26	-4,14
21	Circle	-	-3,40	138,65	139	-	-5,92	-48,75	-	-1,33	-2,23
22	Circle	-	16,25	120,98	121	-	11,89	-29,00	-	18,04	-3,44
23	Circle	-	19,95	90,34	90	-	18,59	0,36	-	18,12	-0,11

№	Класс	Наклон заготовки относительно Оси X камеры, ° (Эталонные значения)	Наклон заготовки относительно Оси Y камеры, ° (Эталонные значения)	Наклон заготовки относительно Оси Z камеры, ° (Эталонные значения)	Наклон относительно Оси Z объекта измеренный с помощью FindContours, °	Наклон заготовки относительно Оси X камеры, измеренный без поворота робота к маске сегментации, °	Наклон заготовки относительно Оси Y камеры, измеренный без поворота робота к маске сегментации, °	Наклон заготовки относительно Оси Z камеры, измеренный без поворота робота к маске сегментации, °	Наклон заготовки относительно Оси X камеры, измеренный под прямым углом к маске сегментации, °	Наклон заготовки относительно Оси Y камеры, измеренный под прямым углом к маске сегментации, °	Наклон заготовки относительно Оси Z камеры, измеренный под прямым углом к маске сегментации, °
24	Circle	-	15,00	55,83	56	-	13,40	36,48	-	15,25	2,22
25	Circle	-	-13,90	54,65	55	-	-9,81	33,59	-	-13,92	-2,54
26	Circle	-	-12,35	129,78	130	-	-19,20	-21,61	-	-14,16	-3,22
27	Circle	-	-16,15	41,13	41	-	-16,95	50,45	-	-16,22	1,72
28	Circle	-	16,60	72,86	73	-	14,32	17,24	-	13,70	2,08
29	Hexagon	0,65	11,05	165,34	165	-11,15	-7,70	-75,85	-0,34	8,66	-0,46
30	Square	0,15	-14,80	79,97	80	-12,45	-13,53	10,62	-0,31	-15,21	0,47
31	Hexagon	0,15	12,80	86,45	86	-4,15	11,90	5,69	-0,40	11,27	1,82
32	Circle	-	1,15	0,00	0	-	19,69	89,31	-	-1,21	1,20
33	Square	1,20	-4,15	85,63	86	-0,85	-2,39	3,54	-1,13	-6,19	0,45
34	Square	0,85	2,15	50,66	51	-2,85	-8,07	42,42	-1,54	-0,49	1,15
35	Hexagon	0,00	39,20	94,98	95	0,10	38,55	-0,81	0,05	39,32	-0,71
36	Hexagon	0,00	-35,90	90,78	91	-9,12	-35,62	1,37	-1,40	-38,04	1,63
37	Circle	-	-22,10	96,49	96	-	-23,72	-4,89	-	-23,15	-2,13
38	Square	6,70	17,65	117,17	117	12,28	27,51	-54,38	7,90	17,04	-2,24
39	Hexagon	6,35	12,00	54,58	55	8,77	8,34	34,50	6,16	10,16	-0,81
40	Hexagon	3,00	4,90	42,98	43	4,29	-10,91	46,26	6,13	-1,54	1,45
41	Square	4,10	-15,55	63,25	63	-6,47	-15,96	35,30	4,37	-15,04	0,81
42	Hexagon	2,40	1,50	142,96	143	7,95	-8,63	-72,98	-0,04	0,85	-1,80
43	Square	4,60	12,70	51,93	52	0,45	8,99	38,78	-1,56	13,49	1,82
44	Circle	-	23,50	68,68	69	-	22,11	20,33	-	21,30	0,47
45	Hexagon	7,90	-26,05	142,37	142	16,49	-19,49	-52,87	5,86	-23,32	-1,12
46	Square	6,45	-10,55	129,95	130	10,17	-12,56	-74,35	2,96	-12,69	-0,57
47	Circle	-	12,10	44,36	44	-	9,39	42,68	-	12,68	1,68
48	Square	3,25	12,65	155,95	156	0,95	9,57	-83,36	2,57	14,26	-2,68
49	Circle	-	3,25	180,36	180	-	10,25	-89,45	-	4,51	0,86
50	Circle	-	21,25	124,83	125	-	16,85	-19,90	-	22,91	-1,22

Таблица Б.2 - Погрешность алгоритма определения углов наклона объекта в пространстве

№	Класс	Наклон заготовки относительно Оси X камеры, ° (Эталонные значения)	Наклон заготовки относительно Оси Y камеры, ° (Эталонные значения)	Наклон заготовки относительно Оси Z камеры, ° (Эталонные значения)	Погрешность вычисления наклона заготовки относительно Оси X камеры, измеренный без поворота робота к маске сегментации, °	Погрешность вычисления наклона заготовки относительно Оси Y камеры, измеренный без поворота робота к маске сегментации, °	Погрешность вычисления наклона заготовки относительно Оси Z камеры, измеренный без поворота робота к маске сегментации, °	Погрешность вычисления наклона заготовки относительно Оси X камеры, измеренный под прямым углом к маске сегментации, °	Погрешность вычисления наклона заготовки относительно Оси Y камеры, измеренный под прямым углом к маске сегментации, °	Погрешность вычисления наклона заготовки относительно Оси Z камеры, измеренный под прямым углом к маске сегментации, °
1	Square	0,60	0,40	90,45	0,22	3,09	0,62	1,32	0,85	-0,19
2	Square	2,15	0,60	41,25	2,38	2,00	2,17	1,25	1,04	1,80
3	Square	1,25	2,15	132,65	9,33	12,73	1,41	2,28	0,98	0,72
4	Square	6,60	11,20	135,78	1,11	20,96	14,62	1,15	1,71	-0,08
5	Square	1,60	11,60	90,23	10,51	1,47	0,54	3,36	0,94	0,57
6	Square	0,30	7,40	63,98	0,14	2,26	3,41	3,20	1,72	-0,78
7	Square	0,80	-13,80	56,23	2,51	6,57	1,33	1,89	3,62	0,28
8	Square	1,35	-12,00	90,76	7,94	0,71	0,54	3,93	1,34	-1,04
9	Square	2,15	-21,00	123,86	1,91	1,10	19,36	4,74	1,72	-0,98
10	Hexagon	2,85	1,35	142,98	1,25	12,39	2,85	1,57	6,88	-3,52
11	Hexagon	2,00	0,30	94,46	8,92	4,23	0,93	5,53	1,75	-2,05
12	Hexagon	1,70	1,90	58,84	5,02	1,63	1,01	7,49	3,69	-1,00
13	Hexagon	0,10	13,10	130,32	2,41	16,34	3,98	2,90	3,52	-1,12
14	Hexagon	1,20	15,10	90,90	0,28	0,29	100,31	2,39	1,50	1,29
15	Hexagon	0,45	15,40	50,44	5,71	0,45	0,08	2,12	0,14	1,54
16	Hexagon	0,25	-20,00	113,89	2,17	4,06	2,88	1,63	0,85	-1,98
17	Hexagon	0,35	-21,30	96,12	8,08	0,83	0,63	6,15	1,51	-1,71
18	Hexagon	0,45	-13,70	45,54	6,23	0,91	0,02	6,79	0,29	0,48
19	Circle	-	2,27	61,97		7,23	1,91		4,95	-0,79
20	Circle	-	0,30	95,24		0,06	1,84		0,04	-4,14
21	Circle	-	-3,40	138,65		2,52	0,10		2,07	-2,23
22	Circle	-	16,25	120,98		4,36	1,98		1,79	-3,44
23	Circle	-	19,95	90,34		1,36	0,70		1,83	-0,11
24	Circle	-	15,00	55,83		1,60	2,31		0,25	2,22
25	Circle	-	-13,90	54,65		4,09	1,76		0,02	-2,54

№	Класс	Наклон заготовки относительно Оси X камеры, ° (Эталонные значения)	Наклон заготовки относительно Оси Y камеры, ° (Эталонные значения)	Наклон заготовки относительно Оси Z камеры, ° (Эталонные значения)	Погрешность вычисления наклона заготовки относительно Оси X камеры, измеренный без поворота робота к маске сегментации, °	Погрешность вычисления наклона заготовки относительно Оси Y камеры, измеренный без поворота робота к маске сегментации, °	Погрешность вычисления наклона заготовки относительно Оси Z камеры, измеренный без поворота робота к маске сегментации, °	Погрешность вычисления наклона заготовки относительно Оси X камеры, измеренный под прямым углом к маске сегментации, °	Погрешность вычисления наклона заготовки относительно Оси Y камеры, измеренный под прямым углом к маске сегментации, °	Погрешность вычисления наклона заготовки относительно Оси Z камеры, измеренный под прямым углом к маске сегментации, °
26	Circle	-	-12,35	129,78		6,85	18,17		1,81	-3,22
27	Circle	-	-16,15	41,13		0,80	1,58		0,07	1,72
28	Circle	-	16,60	72,86		2,28	0,10		2,90	2,08
29	Hexagon	0,65	11,05	165,34	11,80	18,75	0,50	0,99	2,39	-0,46
30	Square	0,15	-14,80	79,97	12,60	1,27	0,59	0,46	0,41	0,47
31	Hexagon	0,15	12,80	86,45	4,30	0,90	2,14	0,55	1,53	1,82
32	Circle	-	1,15	0,00		18,54	0,69		2,36	-1,20
33	Square	1,20	-4,15	85,63	2,05	1,76	0,83	2,33	2,04	0,45
34	Square	0,85	2,15	50,66	3,70	10,22	3,08	2,39	2,64	1,15
35	Hexagon	0,00	39,20	94,98	0,10	0,65	4,17	0,05	0,12	-0,71
36	Hexagon	0,00	-35,90	90,78	9,12	0,28	2,15	1,40	2,14	1,63
37	Circle	-	-22,10	96,49		1,62	1,60		1,05	-2,13
38	Square	6,70	17,65	117,17	5,58	9,86	27,21	1,20	0,61	-2,24
39	Hexagon	6,35	12,00	54,58	2,42	3,66	0,92	0,19	1,84	-0,81
40	Hexagon	3,00	4,90	42,98	1,29	15,81	0,76	3,13	6,44	1,45
41	Square	4,10	-15,55	63,25	10,57	0,41	8,55	0,27	0,51	0,81
42	Hexagon	2,40	1,50	142,96	5,55	10,13	20,02	2,44	0,65	-1,80
43	Square	4,60	12,70	51,93	4,15	3,71	0,71	6,16	0,79	1,82
44	Circle	-	23,50	68,68		1,39	0,99		2,20	0,47
45	Hexagon	7,90	-26,05	142,37	8,59	6,56	0,50	2,04	2,73	-1,12
46	Square	6,45	-10,55	129,95	3,72	2,01	34,40	3,49	2,14	-0,57
47	Circle	-	12,10	44,36		2,71	2,96		0,58	1,68
48	Square	3,25	12,65	155,95	2,30	3,08	17,41	0,68	1,61	-2,68
49	Circle	-	3,25	180,36		7,00	0,91		1,26	0,86
50	Circle	-	21,25	124,83		4,40	14,93		1,66	-1,22
<b>Средняя погрешность</b>					<b>4,82</b>	<b>4,96</b>	<b>6,66</b>	<b>2,57</b>	<b>1,75</b>	<b>-0,41</b>

**Приложение В. Результаты экспериментов решения для сортировки ферромагнитных деталей промышленным манипулятором**

Таблица В.1 - Комплексное тестирование разработанного решения для технологического процесса сортировки ферромагнитных объектов (Эксперимент 1)

№	Класс	Распознанный класс	Расстояние по вертикальной оси, мм (Эталонные значения)	Измеренное расстояние по вертикальной оси, мм	Погрешность вычисления расстояния по вертикальной оси, мм	Захват объекта
1	circle	circle	389	387	2	Да
2	circle	circle	390	392	-2	Да
3	circle	circle	391	392	-1	Да
4	hexagon	hexagon	395	394	1	Да
5	square	square	395	395	0	Да
6	square	square	395	396	-1	Да
7	square	square	395	396	-1	Да
8	square	square	397	398	-1	Да
9	square	square	404	403	1	Да
10	hexagon	hexagon	404	404	0	Да
11	hexagon	hexagon	405	405	0	Да
12	circle	circle	409	409	0	Да
13	hexagon	hexagon	415	416	-1	Да
14	hexagon	hexagon	416	418	-2	Да
15	circle	circle	417	418	-1	Да
16	hexagon	hexagon	419	419	0	Да
17	hexagon	hexagon	419	419	0	Да
18	circle	circle	420	421	-1	Да
19	square	square	425	426	-1	Да
20	hexagon	hexagon	425	427	-2	Да
Средняя погрешность вычисления расстояния по вертикальной оси					0,9	

Таблица В.2 - Комплексное тестирование разработанного решения для технологического процесса сортировки ферромагнитных объектов (Эксперимент 2)

№	Класс	Распознанный класс	Расстояние по вертикальной оси, мм (Эталонные значения)	Измеренное расстояние по вертикальной оси, мм	Погрешность вычисления расстояния по вертикальной оси	Захват объекта
21	circle	circle	398	397	1	Да
22	square	square	402	402	0	Да
23	hexagon	hexagon	404	403	1	Да
24	square	square	405	407	-2	Да
25	circle*	circle	406	407	-1	Да
26	square	square	409	410	-1	Да
27	hexagon	hexagon	410	411	-1	Да
28	square	hexagon	411	411	0	Да
29	hexagon	hexagon	416	417	-1	Да
30	circle	circle	418	418	0	Да
31	hexagon*	hexagon	418	418	0	Да
32	circle	circle	419	419	0	Да
33	circle	circle	420	419	1	Да
34	circle	circle	421	420	1	Да
35	circle*	circle	422	420	2	Да
36	circle	circle	423	424	-1	Да
37	square	square	424	424	0	Да
38	circle	circle	428	427	1	Да
39	square	square	428	429	-1	Да
40	square	square	429	429	0	Да
41	hexagon	hexagon	432	431	1	Да
Средняя погрешность вычисления расстояния по вертикальной оси					0,761905	

\*- Объект расположен возле стенки накопителя

Таблица В.3 - Комплексное тестирование разработанного решения для технологического процесса сортировки ферромагнитных объектов (Эксперимент 3)

№	Класс	Распознаваемый класс	Расстояние по вертикальной оси, мм (Эталонные значения)	Измеренное расстояние по вертикальной оси, мм	Погрешность вычисления расстояния по вертикальной оси	Захват объекта
42	square	square	379	380	-1	Да
43	square	square	381	380	1	Да
44	circle	circle	384	384	0	Да
45	square	square	392	393	-1	Да
46	circle	circle	397	398	-1	Да
47	circle	circle	401	400	1	Да
48	hexagon	hexagon	405	406	-1	Да
49	square*	square	408	409	-1	Да
50	square	square	410	409	1	Да
51	square	square	412	413	-1	Да
52	circle	circle	416	418	-2	Да
53	hexagon*	hexagon	417	418	-1	Да
54	hexagon	hexagon	419	419	0	Да
55	circle	circle	420	419	1	Да
56	hexagon	hexagon	420	420	0	Да
57	circle*	circle	420	420	0	Да
58	circle*	circle	421	420	1	Да
59	circle	circle	421	420	1	Да
60	hexagon*	hexagon	422	420	2	Да
61	circle	circle	422	420	2	Да
62	circle	circle	423	422	1	Да
63	square	square	425	426	-1	Да
64	square	square	426	426	0	Да
65	square	square	428	427	1	Да
66	square	square	429	427	2	Да
67	square	square	432	431	1	Да
Средняя погрешность вычисления расстояния по вертикальной оси					0,961538	

\*- Объект расположен возле стенки накопителя

# Приложение Г. Патент на полезную модель

## РОССИЙСКАЯ ФЕДЕРАЦИЯ



### ПАТЕНТ

НА ПОЛЕЗНУЮ МОДЕЛЬ

**№ 226513**

### Пневматический магнитный захват робота-манипулятора

Патентообладатель: *Ордена Трудового Красного Знамени федеральное государственное бюджетное образовательное учреждение высшего образования "Московский технический университет связи и информатики" (МТУСИ) (RU)*

Авторы: *Белов Никита Вадимович (RU), Вовик Андрей Геннадьевич (RU), Воронова Лилия Ивановна (RU), Воронов Вячеслав Игоревич (RU), Хохлова Мария Владимировна (RU)*

Заявка № **2024104452**

Приоритет полезной модели **22 февраля 2024 г.**

Дата государственной регистрации в Государственном реестре полезных моделей Российской Федерации **06 июня 2024 г.**

Срок действия исключительного права на полезную модель истекает **22 февраля 2034 г.**

*Руководитель Федеральной службы по интеллектуальной собственности*

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ  
Сертификат: 429b6a0fe3851164ba96f83b73b4aa7  
Владелец: **Зубов Юрий Сергеевич**  
Действителен с 10.05.2023 по 02.08.2024

*Ю.С. Зубов*



## Приложение Д. Акты внедрения

Акт об использовании результатов диссертационной работы в учебном процессе кафедры «Интеллектуальные системы в управлении и автоматизации»

УТВЕРЖДАЮ  
Зам. проректор по учебной работе,  
Московского технического университета  
связи и информатики

*В.В. Краснов*  
В.В. Краснов

« 02 » 10 2010 г.



### АКТ

об использовании результатов диссертационной работы

Белова Никиты Вадимовича на тему:

«Методы и алгоритмы сортировки ферромагнитных деталей промышленным манипулятором с использованием компьютерного зрения» в учебном процессе кафедры «Интеллектуальные системы в управлении и автоматизации»

Комиссия в составе: заведующего кафедрой «Интеллектуальные системы в управлении и автоматизации» к.т.н., доцента Воронова Вячеслава Игоревича; декана факультета «Кибернетика и информационная безопасность» к.т.н., доцента Иевлева Олега Павловича; начальника Отдела планирования и организации учебного процесса МТУСИ Кузнецовой Виктории Анатольевны составили настоящий акт о том, что результаты диссертационной работы Белова Никиты Вадимовича на тему: «Методы и алгоритмы сортировки ферромагнитных деталей промышленным манипулятором с использованием компьютерного зрения» используются в учебном процессе кафедры «Интеллектуальные системы в управлении и автоматизации», а именно:

- на основе полученных результатов модифицирована рабочая программа учебной дисциплины «Программная и аппаратная производственная интеграция. Тестирование» для магистрантов по направлению 15.04.04 «Автоматизация технологических процессов и производств», программа «Системы искусственного интеллекта промышленного интернета вещей». В нее введен раздел «Применение компьютерного зрения для промышленных манипуляторов», включающий разработанные в диссертации Белова Н.В. методические подходы и рекомендации по интеграции систем стереозрения и методов определения координат объектов захвата для промышленных манипуляторов.

- система дистанционного управления промышленным манипулятором Kuka, предложенная в диссертации, применяется при разработке интеллектуальных программно-аппаратных комплексов магистрантами направления 15.04.04 «Автоматизация технологических процессов и производств» при написании магистерских диссертаций с использованием оборудования центра робототехники МТУСИ, в частности двух промышленных манипуляторов Kuka.

Указанные новации расширяют научно-методическую базу учебных дисциплин и способствуют овладению профессиональными компетенциями: ПК-3 Способен разрабатывать управляющие программы для изготовления машиностроительных изделий высокой сложности, ПК-6 Способен осуществлять руководство инженеринговой деятельностью.

Заведующий кафедрой ИСУиА

*В.И. Воронов*

Воронов В.И.

Декан факультета КиИБ

*О.П. Иевлев*

Иевлев О.П.

Начальник ОПиОУП

*В.А. Кузнецова*

Кузнецова В.А.

**Акт об использовании результатов диссертационной работы в научно-производственной деятельности ООО «Геларм»**



ООО «ГЕЛАРМ» ИНН/КПП 7743196892 / 774301001  
127238, Москва, Дмитровское ш., 71Б, офис 201  
тел. +7 (495) 008-67-21  
email: info@gelarm.ru

**АКТ**

о внедрении результатов диссертационной работы  
Белова Никиты Вадимовича на тему «Методы и алгоритмы сортировки ферромагнитных деталей промышленным манипулятором с использованием компьютерного зрения»

Комиссия в составе: генеральный директор Ушанов Илья Александрович и Директор департамента систем мониторинга Соломаха Александр Владимирович составили настоящий акт о том, что результаты диссертационной работы Белова Никиты Вадимовича на тему: «Методы и алгоритмы сортировки ферромагнитных деталей промышленным манипулятором с использованием компьютерного зрения» используются в работе департамента разработки ООО "Геларм" а именно:

- подсистема дистанционного управления промышленным манипулятором Кука, для удаленного взаимодействия, в режиме потоковой передачи данных, с манипулятором;

- алгоритм распознавания объектов при обработке видеопотока на удаленном устройстве, обеспечивающий расчет координат объекта в рабочей зоне манипулятора, с помощью компьютерного зрения.

Результаты диссертационного исследования обеспечили эффективную обработку исполнительных команд на удаленном устройстве и их передачу на контроллер робота, что упростило взаимодействие оператора с промышленным манипулятором Кука.

Генеральный директор

 (И.А. Ушанов)

Директор департамента систем мониторинга

 (А.В. Соломаха)

