

Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО
ОБРАЗОВАНИЯ
“НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО”
(УНИВЕРСИТЕТ ИТМО)

ЦЕНТР АВТОРИЗОВАННОГО ОБУЧЕНИЯ ИНФОРМАЦИОННЫМ ТЕХНОЛОГИЯМ

ИТОГОВАЯ АТТЕСТАЦИОННАЯ РАБОТА

**Разработка робототехнического комплекса с использованием системы
технического зрения для анализа продукции на конвейере**

Автор Варламов Игорь Андреевич _____
(Фамилия Имя Отчество) (Подпись)

Центр авторизованного обучения информационным технологиям

Наименование программы **«Python-разработчик»**

Руководитель _____
(Подпись)

К защите допустить

Заместитель директора ЦАО ИТ, к.т.н. _____ */ Т.В. Зудилова/*

Санкт-Петербург, 2024г.

Обучающийся Варламов Игорь Андреевич Группа 124/12
(Фамилия, И. О.)

Работа принята « ____ » _____ 2024г.

Работа выполнена с оценкой _____

Дата защиты « » 2024г.

Секретарь ИАК _____ (Фамилия, И. О.) _____ (подпись)

Листов хранения _____

Демонстрационных материалов _____

Оглавление

ВВЕДЕНИЕ.....	4
1. АНАЛИЗ ПОДХОДОВ К РАСПОЗНАВАНИЮ ЦИЛИНДРИЧЕСКИХ ЗАГОТОВОК С МЕТКАМИ	6
1.1. Особенности задачи	6
1.2. Методы алгоритмического обнаружения при помощи CMOS камеры	7
1.1. Методы повышения качества обнаружения.....	11
1.2. Выводы по главе 1	12
2 МАТЕМАТИЧЕСКОЕ ОПИСАНИЕ ОСНОВНЫХ ЭТАПОВ РАЗРАБОТКИ	13
2.2 Алгоритм определения центра детали	13
2.3 Алгоритм определения ориентации детали	13
2.4 Алгоритм определения типа детали.....	13
2.5 Передача данных.....	13
2.6 Переход в систему координат робота	14
2.7 Расчет траектории движения манипулятора	14
2.8 Предобработка изображения	16
2.9 Выводы по главе 2	20
3 РАЗРАБОТКА АЛГОРИТМА СТЗ И УПРАВЛЕНИЯ РОБОТОМ	21
3.1 Предобработка входного изображения	21
3.2 Архитектура системы распознавания	24
3.3 Разработка алгоритма движения робота	24
3.4 Реализация базы данных	25
3.5 Реализация передачи данных	25
3.6 Реализация графического интерфейса	25
3.7 Развертывание на конечной машине пользователя	25
3.8 Выводы по главе 3	25
4 ЭКСПЕРИМЕНТАЛЬНОЕ ИССЛЕДОВАНИЕ	25
4.1 Программа экспериментов.....	25
4.2 Результаты экспериментов.....	25
4.3 Анализ качества алгоритма обнаружения	26
4.4 Выводы по главе 4	26

Введение

Система технического зрения (СТЗ) играет важную роль в управлении промышленным роботом. Она отвечает за распознавание, автоматизацию контроля и анализ объектов на основе изображений.

Эти системы позволяют роботам получать изображения объектов, сцен и рабочих процессов, а затем обрабатывать их с помощью цифровых устройств. Системы технического зрения используются для обнаружения, распознавания или идентификации объектов, определения их местоположения и координат. Полученные данные затем помогают промышленному роботу выполнить задачи, на которые он запрограммирован.

Системы технического зрения (СТЗ) [1] в связке с промышленным роботом или робототехническим комплексом находят широкое применение в различных областях производства. Вот несколько вариантов использования:

1. Контроль качества: СТЗ могут использоваться для контроля качества изготавливаемой продукции. Например, обнаружение дефектов на поверхности деталей, коррекция параметров обработки, или проверка правильности сборки изделий.

2. Подбор и сортировка: СТЗ позволяют роботам выполнять задачи по подбору и сортировке предметов на производственной линии. Например, система может распознавать и классифицировать товары или детали, чтобы робот мог правильно распределить их в соответствующие контейнеры или упаковки.

3. Позиционирование и навигация: Системы технического зрения позволяют роботам определять свое местоположение, а также точно определять положение и ориентацию объектов в пространстве. Это особенно важно для задач погрузки, разгрузки и перемещения предметов на производственной площадке.

4. Сбор данных и аналитика: СТЗ могут использоваться для сбора данных о производственных процессах, например, для анализа производственной линии, контроля запасов, или для предупреждения аварийных ситуаций.

5. Сотрудничающие роботы: СТЗ позволяют роботам взаимодействовать с людьми и другими роботами безопасно. Они могут обнаруживать присутствие людей или препятствий, чтобы избегать столкновений и работать в совместном окружении.

Цель данной работы:

Исследование, разработка и реализация робототехнического комплекса (робот Kawasaki RS007), оснащенного системой технического зрения (камера ace Basler), с целью обеспечения автоматизации процесса обнаружения, захвата, транспортировки и сортировки заготовок на производстве. В особенности задачи можно отнести необходимость реализации максимально поддерживаемого программного кода, а также простого и понятно интерфейса взаимодействия с комплексом.

1. Анализ подходов к распознаванию цилиндрических заготовок с метками

1.1. Особенности задачи

В качестве объекта обнаружения в данной задаче выступает заготовка (рисунок 1.1)



Рисунок 1.1 – Объект обнаружения

Так как задачу обнаружения объектов требуется реализовать в условиях плохой освещенности возникает проблема засвета и бликов на металлической поверхности. Это необходимо будет учитывать при выборе алгоритма обнаружения.

Заготовки расположены специальной подложке (рисунок 1.2).

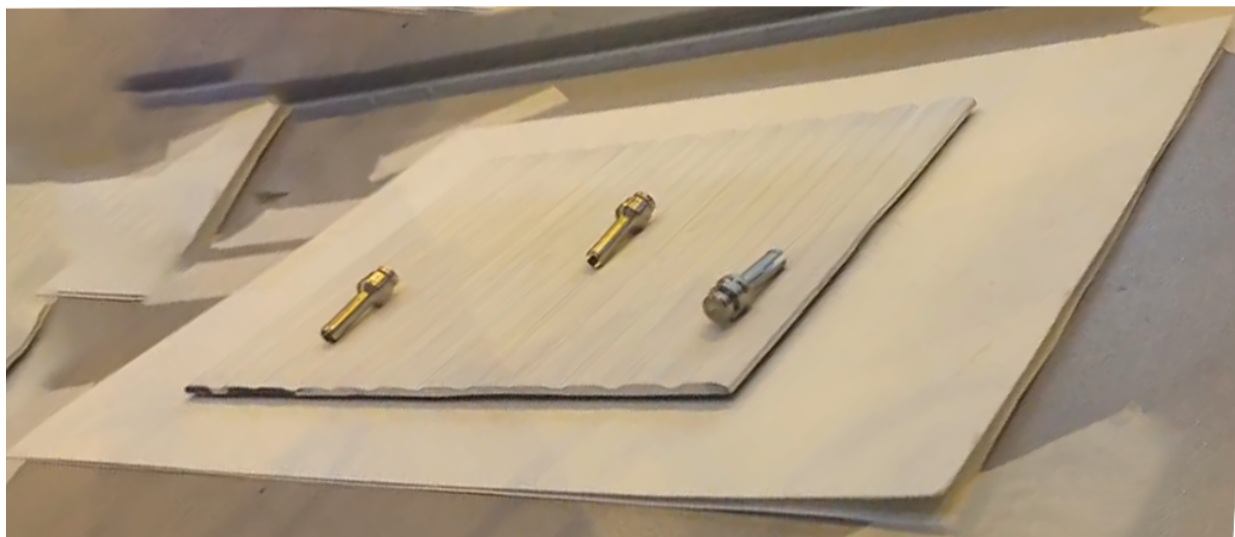


Рисунок 1.2 – Подложка для деталей

Транспортировку заготовок будет выполнять робот Kawasaki RS007 (рисунок 1.3). Передача данных между СТЗ и роботом будет осуществляться по TCP/IP.



Рисунок 1.3 – Робот Kawasaki RS007

1.2. Методы алгоритмического обнаружения при помощи CMOS камеры

Глобально для поиска объектов на подготовленном изображении применяется:

1. Использование геометрических признаков: Этот подход основан на распознавании геометрических особенностей формы заготовок. Алгоритмы компьютерного зрения могут быть использованы для анализа формы, размеров и расположения меток на поверхности цилиндрических объектов.

2. Использование цветовой информации: Техническое зрение также позволяет анализировать цветовую информацию и их контраст с поверхностью заготовок. Это может быть особенно полезно в условиях изменяющегося освещения.

Поиск можно реализовать следующими методами:

1. Алгоритм Хафа (Hough Transform): Алгоритм Хафа часто используется для обнаружения форм на изображениях. Он работает путем преобразования изображения в пространство параметров и поиска линейных структур в этом пространстве.

2. Метод Лапласа: Этот метод основан на выделении краевых структур на изображении с использованием оператора Лапласа. После выделения краев, можно применить алгоритм поиска с использованием этих структур.

3. Алгоритм Марра-Хилд्रेث (Marr-Hildreth Algorithm): Этот алгоритм комбинирует оператор Лапласа с гауссовым размытием для поиска форм на изображении.

4. Поиск при помощи нейросетевого обнаружения: Нейросетевые методы обнаружения объектов на изображениях стали очень популярными благодаря своей способности эффективно извлекать признаки и обучаться на больших объемах данных. В основе этого метода лежит использование сверточных нейронных сетей (CNN), которые способны автоматически извлекать признаки из изображений на разных уровнях абстракции.

5. Поиск при помощи анализа контуров: Этот метод основан на анализе контуров объектов на изображении. Контур представляет собой границу объекта, образованную изменениями интенсивности пикселей. Анализ контуров позволяет определять форму и структуру объектов на изображении. Преимущества метода анализа контуров включают простоту реализации, быструю скорость работы и возможность работы с изображениями высокого разрешения. Однако он может быть менее эффективным в условиях сильного шума или низкого контраста на изображении.

По результатам предварительного эксперимента были получены следующие результаты:

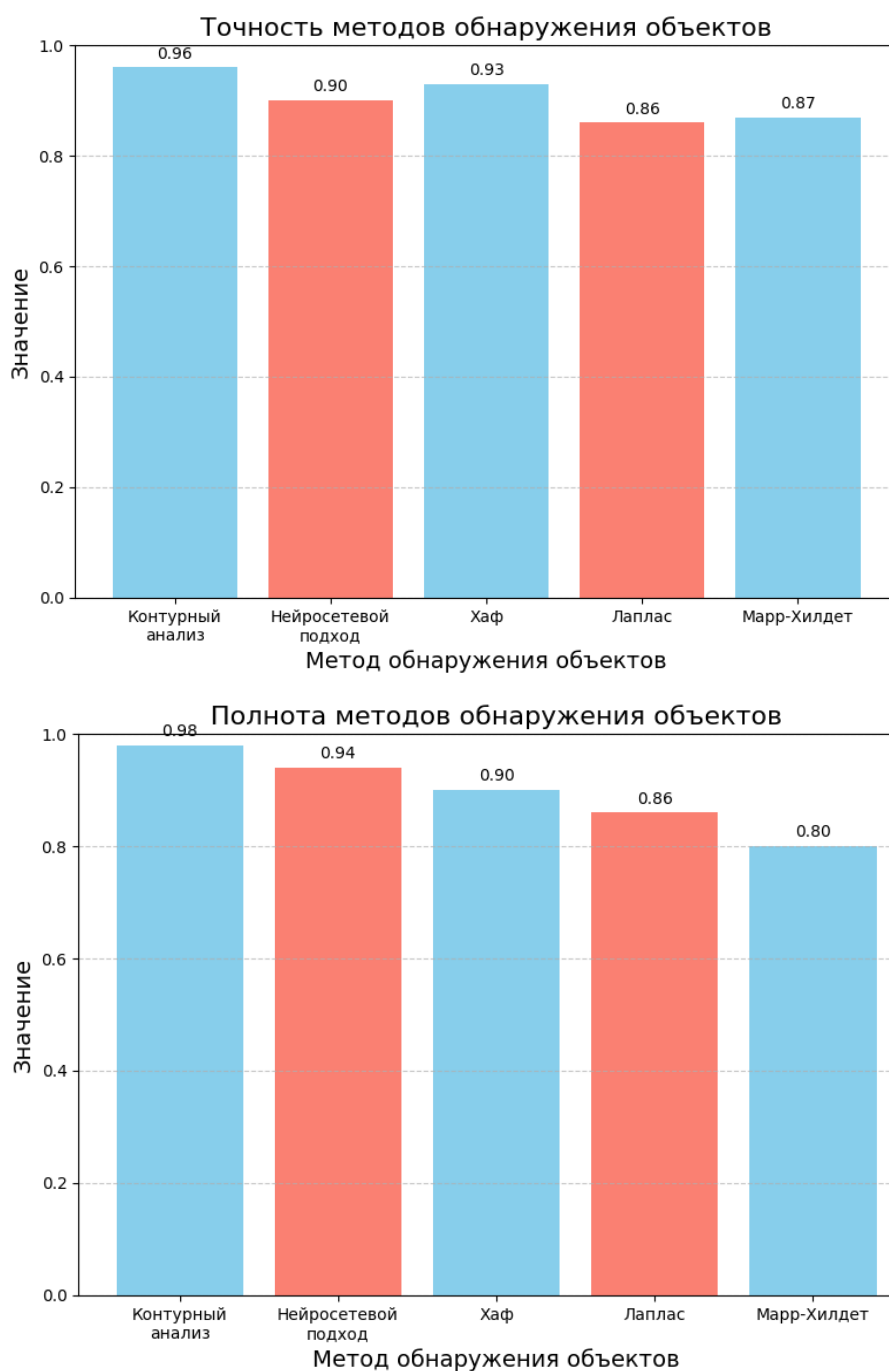


Рисунок 1.4 - Точность и полнота распознавания на подготовленном видео

Под «подготовленным» изображением имеется ввиду качественная настройка параметров захвата камеры, такие как ISO, диафрагма, а также постобработка полученного изображения программно: яркость, контраст, цветокоррекция. Также каждый метод требовал достаточно тонкой настройки путем подбора параметров вручную.

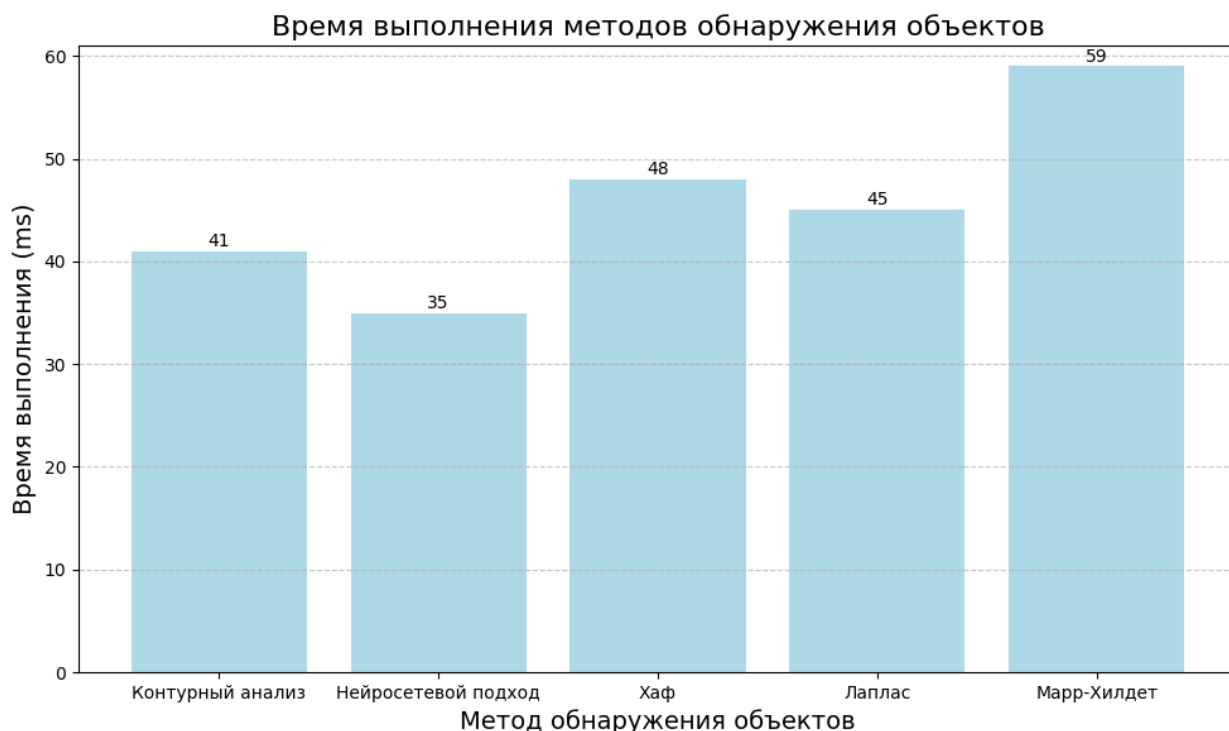


Рисунок 1.5 – Среднее затраченное время на обнаружение

Выше был описан самый простой случай, так как на качество обнаружения не влияли такие факторы, как размытие при движении объектов, изменяющееся в процессе съемки освещение и т.д.

Результаты тестов на «готовом» видео представлены на рисунке 1.4

Итоговые результаты выглядят так:

Контурный анализ – 96.23% точности и 98.15% полноты при среднем времени поиска – 41 ms

Нейросетевой подход – 90.27% точности и 94.84% полноты при среднем времени поиска – 35 ms

Хаф – 93.03% точности и 90.20% полноты при среднем времени поиска – 48 ms

Лаплас – 86.58% точности и 86.01% полноты при среднем времени поиска – 45 ms

Marr-Hildreth Algorithm – 87.91% точности и 80.84% полноты при среднем времени поиска – 59 ms

1.1. Методы повышения качества обнаружения

Стоит отметить, что на результат исследования алгоритмов сильно влияет качество ручного подбора параметров, один из вариантов решения этой проблемы – создание алгоритма динамической подстройки параметров метода в зависимости от изменений внешней среды. Такое решение может быть эффективно, но сильно осложнит решение задачи и увеличит время разработки. В данной работе было принято решение отказаться от такого подхода ввиду необходимости создания максимально простого и легко модернизируемого программного решения, и ограничения по времени разработки.

Как уже было сказано ранее, очень важно уделить внимание на подготовку данных:

1. Улучшение четкости и контрастности: Применение фильтров для улучшения четкости и контрастности изображения помогает сделать объекты более выразительными и улучшить различимость текстур и границ объектов.

2. Фильтрация шума: Уменьшение шума на видео поможет улучшить четкость объектов. Применение различных методов фильтрации, таких как медианный фильтр или фильтр Гаусса, может существенно улучшить качество изображения.

3. Нормализация яркости и баланс белого: Регулирование яркости, контрастности и баланса белого может сделать изображение более естественным и улучшить видимость объектов.

4. Использование методов улучшения резкости: Применение методов увеличения резкости изображения помогает сделать границы объектов более четкими, что может быть важно для точного обнаружения объектов.

5. Суперразрешение изображений: Применение техник суперразрешения позволяет улучшить разрешение и детализацию изображения. Это особенно полезно для обнаружения маленьких объектов или объектов с низкой контрастностью.

6. Компенсация движения: При наличии движения на видео можно использовать методы стабилизации изображения для уменьшения размытия и улучшения качества.

7. Коррекция искажений: Если на видео присутствуют искажения, их коррекция (например, дисторсия или аффинные искажения) может помочь в улучшении визуального восприятия объектов.

1.2. Выводы по главе 1

Для распознавания целесообразно использовать алгоритмические методы.

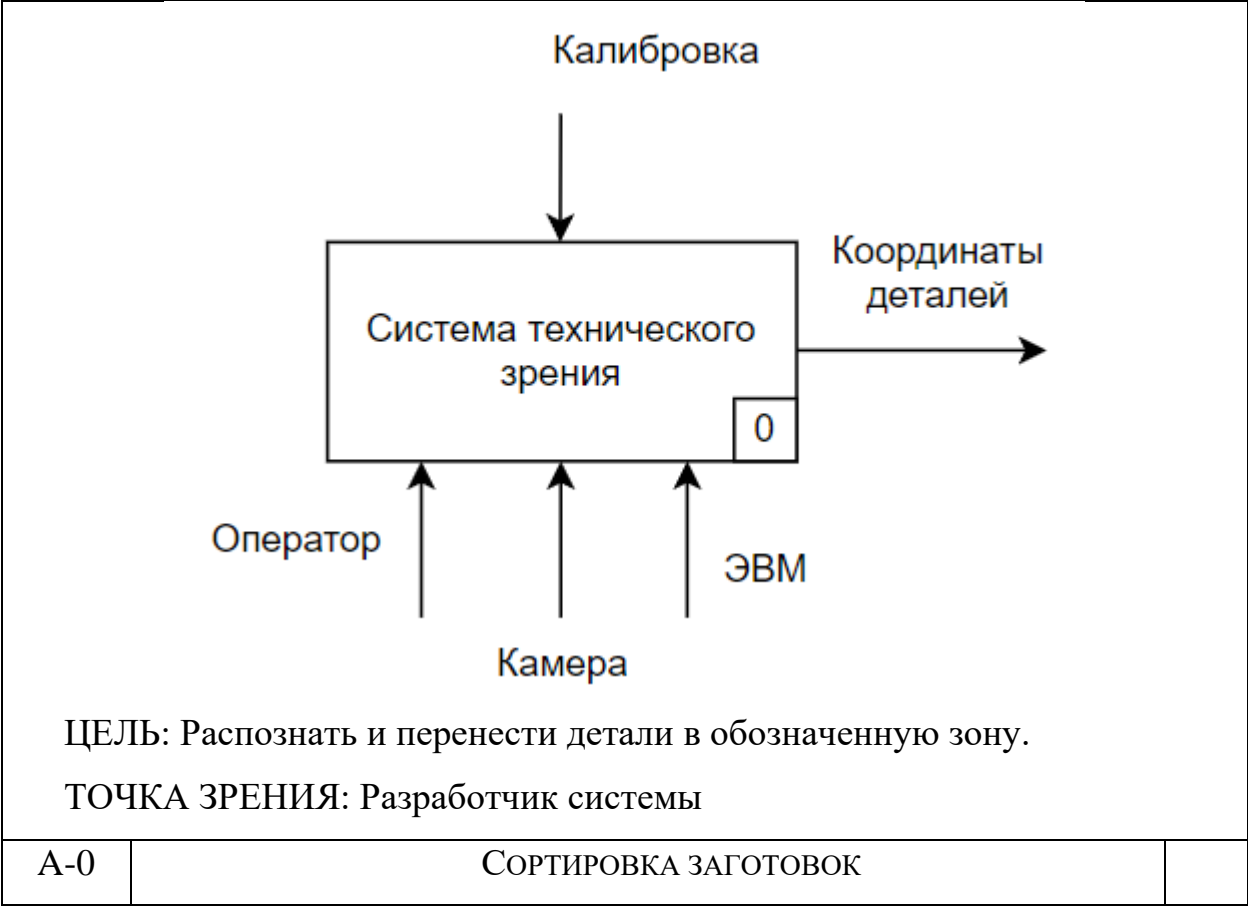


Рисунок 1.6 – Верхний уровень алгоритма

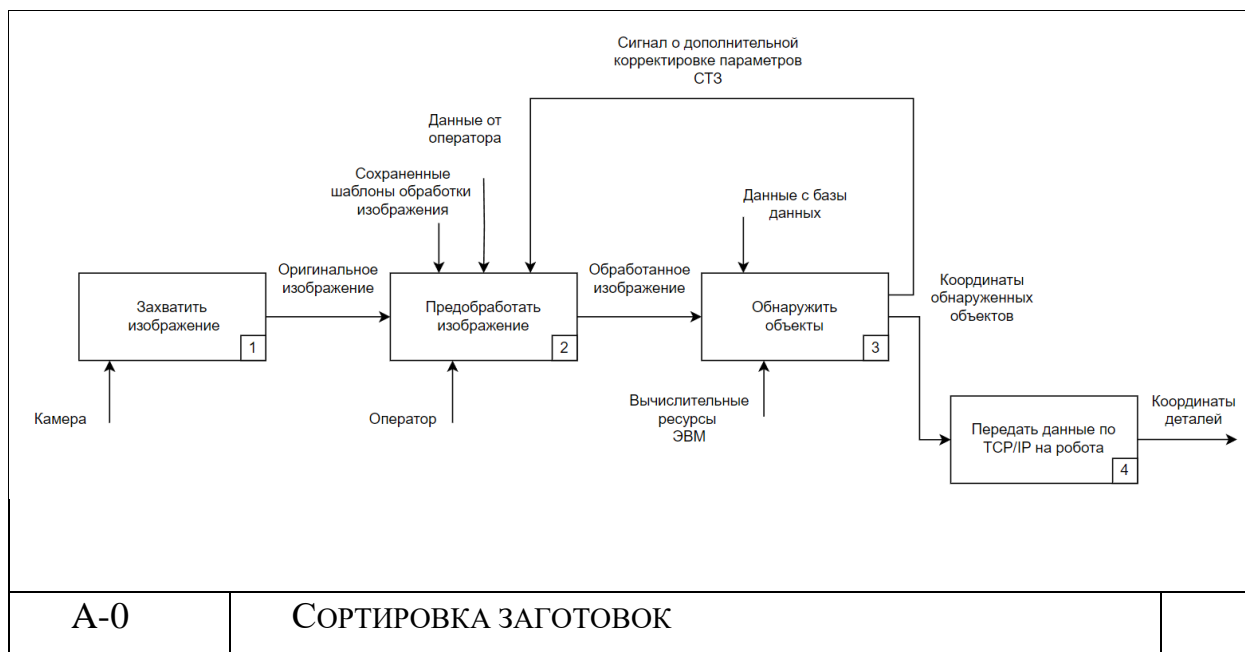


Рисунок 1.7 – Первый уровень декомпозиции алгоритма работы

комплекса

Подробнее про декомпозицию системы будет описано в главе 3.

Так же будет необходима точная настройка получаемого изображения под каждое освещение, блики и т.д., для удобной настрой будет реализован пользовательский интерфейс.

Движение робота будет реализовано на встроенном в роботы Kawasaki языке программирования AS, при помощи K-IDE – программного обеспечения робота.

Реализация СТЗ будет выполнена на OpenCV, ввиду наличия и простоты настройки и использования большинства вышеописанных методов и алгоритмов.

2 Математическое описание основных этапов разработки

2.2 Алгоритм определения центра детали

2.3 Алгоритм определения ориентации детали

2.4 Алгоритм определения типа детали

2.5 Передача данных

2.6 Переход в систему координат робота

2.7 Расчет траектории движения манипулятора

Робот-манипулятор можно рассматривать как набор жестких последовательно соединенных звеньев, каждое из которых оснащено приводом. Робот имеет шесть вращательных кинематических пар (рисунок 2.5.1), причем оси J1, J2, J3 являются переносными, а оси J4, J5, J6 – ориентирующими.

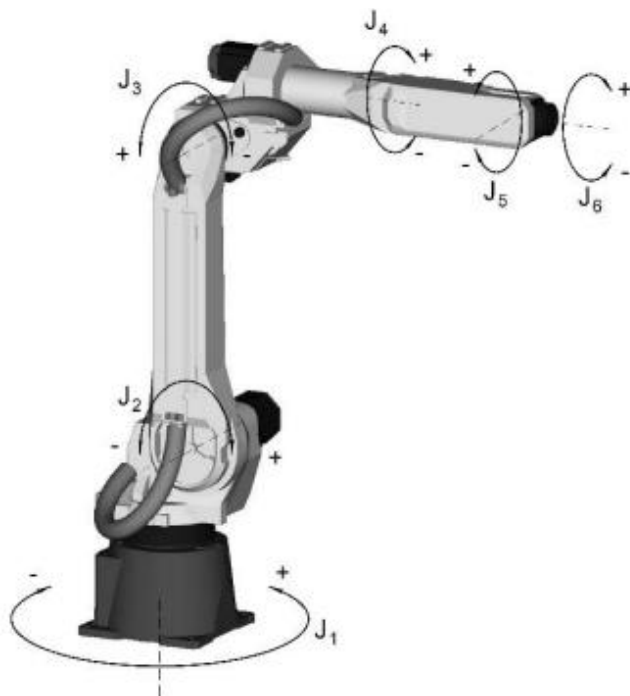


Рисунок 2.10 – 6-ти осевой робот

В общем случае для 6-ти осевого робота, оснащенного позиционером, математическая модель кинематики для обратной задачи можно представить в виде:

$$J_1 = J_1(x, y, z, i, j, k, \alpha_1, \theta_1, a_1, d_1, t)$$

$$J_2 = J_2(x, y, z, i, j, k, \alpha_2, \theta_2, a_2, d_2, t)$$

$$J_3 = J_3(x, y, z, i, j, k, \alpha_3, \theta_3, a_3, d_3, t)$$

$$J_4 = J_4(x, y, z, i, j, k, \alpha_4, \theta_4, a_4, d_4, t) \quad (15)$$

$$J_5 = J_5(x, y, z, i, j, k, \alpha_5, \theta_5, a_5, d_5, t)$$

$$J_6 = J_6(x, y, z, i, j, k, \alpha_6, \theta_6, a_6, d_6, t)$$

где x, y, z – декартовы координаты; i, j, k – направляющие орты оси инструмента; $\alpha_i, \theta_i, a_i, d_i$ – параметры звеньев; t – время; α_i – угловое смещение – угол, на который нужно повернуть ось z_{i-1} вокруг оси x_i , чтобы она стала сонаправлена с осью z_i ; θ_i – присоединительный угол, на который нужно повернуть ось x_{i-1} вокруг оси z_{i-1} , чтобы она стала сонаправлена с осью x_i ; a_i – линейное смещение – расстояния между пересечением оси z_{i-1} с осью x_i и началом i -й системы координат, отсчитываемое вдоль оси x_i ; d_i – расстояние между пересечением оси z_{i-1} с осью x_i и началом $(i-1)$ -й системы координат, отсчитываемое вдоль оси z_{i-1} . Для вращательных сочленений параметры α_i, θ_i, d_i являются постоянными величинами для каждой конкретной модели робота, характеризующими конструкцию звеньев.

Существует множество методов решения обратной задачи кинематики: метод обратных преобразований, прямые геометрические методы, метод на основе нелинейного математического программирования. Последний получил наибольшую популярность, так как при большом количестве возможных решений можно наложить ограничения, дающие оптимальное решение.

В качестве исходных данных для расчета выступают координаты начальной X_0 и конечной X_1 точек траектории, координаты базы робота O относительно мировой системы координат, которые представлены векторами в следующем виде:

$$X_0 = \begin{bmatrix} x_0 \\ y_0 \\ z_0 \\ i_0 \\ j_0 \\ k_0 \end{bmatrix}, X_1 = \begin{bmatrix} x_1 \\ y_1 \\ z_1 \\ i_1 \\ j_1 \\ k_1 \end{bmatrix}, O = \begin{bmatrix} x_0 \\ y_0 \\ z_0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (16)$$

2.8 Предобработка изображения

Очень важно в контексте выбранных алгоритмов обеспечить качественные выходные данные для них. Этого можно достичь при помощи предобработки входного изображения.

Фильтрация в пространственной может рассматриваться, как некоторый оператор действующий на окрестность точки (x,y) на изображении

$$g(x, y) = T[f(x, y)] \quad (17)$$
$$T: R^{m \times n} \rightarrow R$$

f – исходное изображение размером $N \times M$,

T – оператор, действующий на область $m \times n$ изображения f в окрестности точки (x,y)

g – результирующее фильтрованное изображение

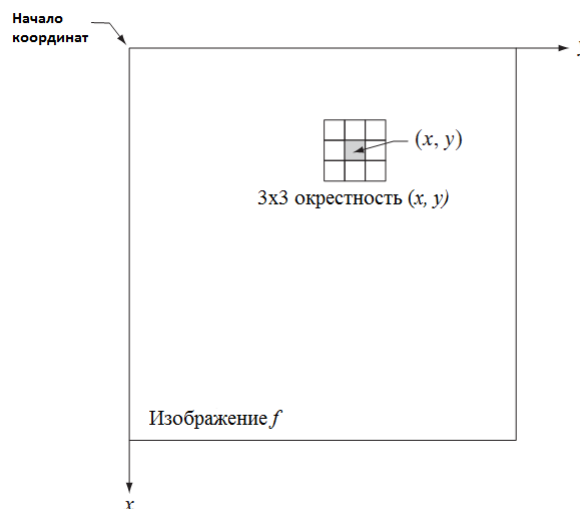


Рисунок 2.11 – Фильтрация в пространственной области

1. Понижение шума:

- Фильтрация: применение фильтров, таких как медианный, гауссовский или билатеральный, для удаления шумов изображения.

2. Улучшение контрастности и резкости:

- Изменение яркости и контраста: коррекция яркости и контраста для улучшения визуального эффекта и облегчения дальнейшего анализа.

- Увеличение резкости: увеличение резкости изображения для более четкого отображения особенностей объектов.

3. Удаление фонов и объектов в заднем плане:

- Бинаризация изображения: преобразование изображения в черно-белое, чтобы выделить объекты от фона.

- Вычитание фона: методы для удаления фона из изображения, например, с помощью вычитания фона или алгоритмов машинного обучения.

4. Устранение искажений и коррекция:

- Искажение перспективы и виньетирование: коррекция искажений, возникающих из-за перспективы и объективов камер.

- Цветовая коррекция: приведение цветового различия между изображениями к стандартным значениям, если нужно.

5. Сегментация объектов:

- Границы и контуры: выделение контуров объектов для дальнейшей сегментации и анализа.

- Кластеризация: разделение изображения на различные кластеры для выделения объектов разного типа или цвета.

6. Улучшение освещения:

- Нормализация освещения: методы для устранения изменчивости освещения и выделения объектов на изображении.

Для достижения удовлетворительного результата, необходимого для корректной и стабильной работы выбранных алгоритмов выберем следующие методы:

Гауссов фильтр [7]– фильтр ядром которого является двумерный гесссиан

$$w(i, j) = e^{-\left(\frac{i^2 + j^2}{2\sigma}\right)} \quad (18)$$

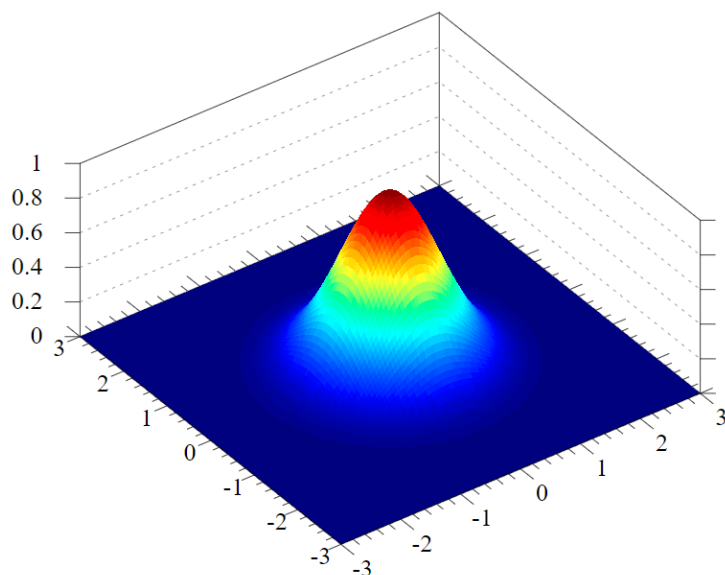


Рисунок 2.12 – График двумерной функции фильтра Гаусса

Сам процесс фильтрации представлен нормированной формулой выше (17).

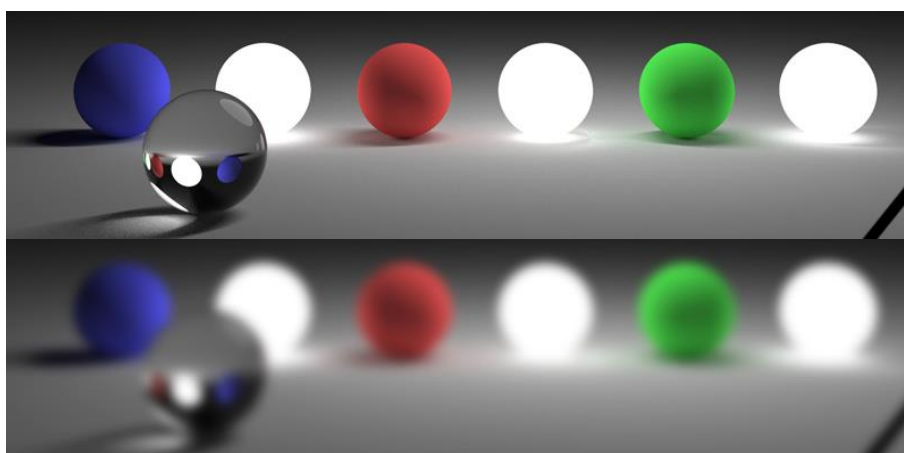


Рисунок 2.13 – Пример фильтрации Гаусса

Фильтрация по уровню интенсивности – на выходе данного фильтра остаются только те пиксели изображения, интенсивность которых попадает в заданный диапазон

$$g(x, y) = \begin{cases} f(x, y), & level_{max} \leq f(x, y) \leq level_{min} \\ 0 & \text{иначе} \end{cases} \quad (19)$$

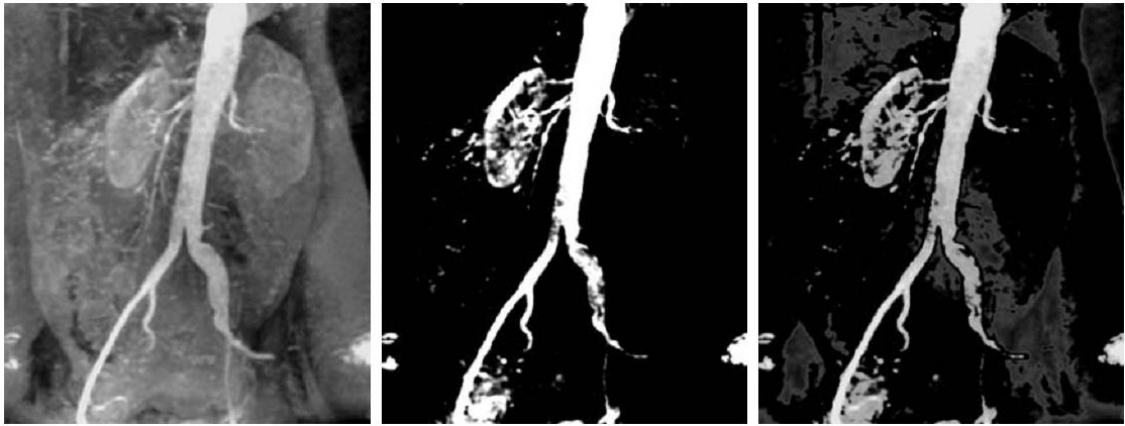


Рисунок 2.14 – Пример фильтрации изображения по уровню (выделение сосудов на снимке)

Нормализация интенсивности изображения – процедура, при которой интенсивность каждого пикселя умножается на константу, приводящее к масштабированию интенсивностей к максимальному диапазону формата представления данных

$$g(x, y) = f(x, y) \cdot \frac{255}{\max f} \quad (20)$$

Изменение средней яркости изображения – добавление фиксированного значения яркости к каждому пикселю

$$g(x, y) = f(x, y) + c \quad (21)$$

где c – постоянный сдвиг интенсивности.

Гамма коррекция

$$g(x, y) = c \cdot (f(x, y) + \epsilon)^\gamma \quad (22)$$

$\gamma, c = \text{const} > 0$

ϵ – малая величина, если необходим не нулевое значение на выходе при нулевом входе

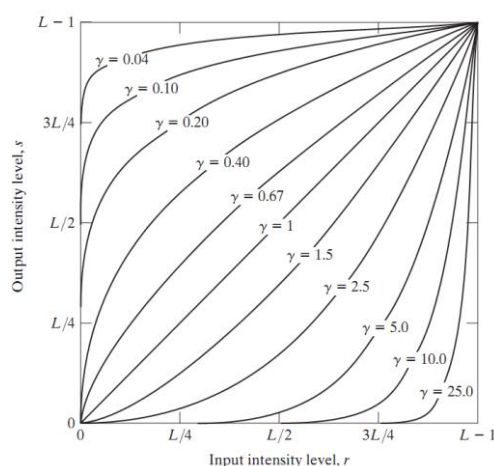


Рисунок 2.15 – Функция гамма преобразования интенсивности



Рисунок 2.16 – Справа исходное изображение, слева – после гамма преобразования

2.9 Выводы по главе 2

В этой главе были рассмотрены математические представления используемых методов.

Было разработано несколько собственных алгоритмов обработки изображения. Следующий шаг – разработать и реализовать целую систему для обнаружения, настройки, преобразования, отправки данных и движения робота, которая объединит в себе все вышеописанные этапы.

3 Разработка алгоритма СТЗ и управления роботом

3.1Предобработка входного изображения

При реализации каждого блока системы (рисунок 1.7) необходимо определить, что мы будем получать на вход, и какие данные будем передавать следующему блоку (функции или алгоритма). Для четкого определения этих данных воспользуемся методологией функционального моделирования IDEF0 или же SADT (Structural Analysis and Design Technique).

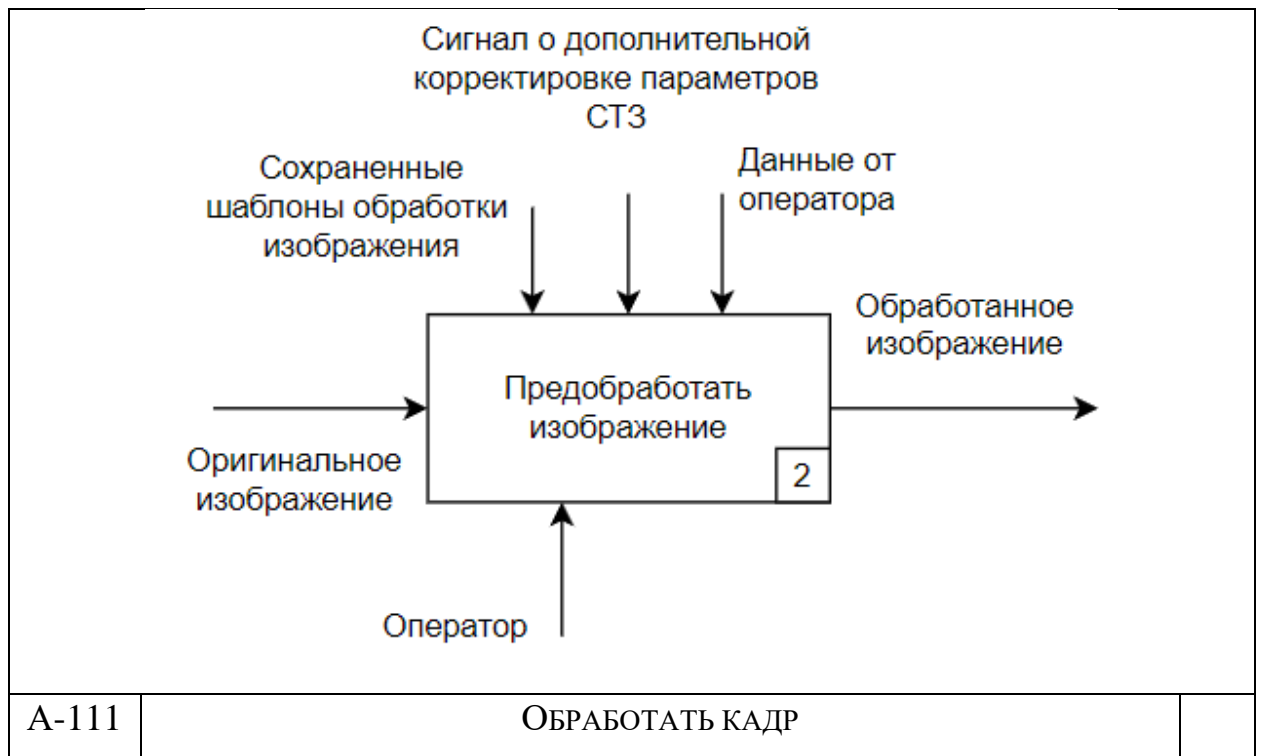


Рисунок 3.1 – Верхний уровень алгоритма предобработки изображения

Теперь декомпозируем систему. Необходимый уровень декомпозиции определим так: будем считать систему декомпозированной при достижении уровня описания математической модели, представленной в главе 2.

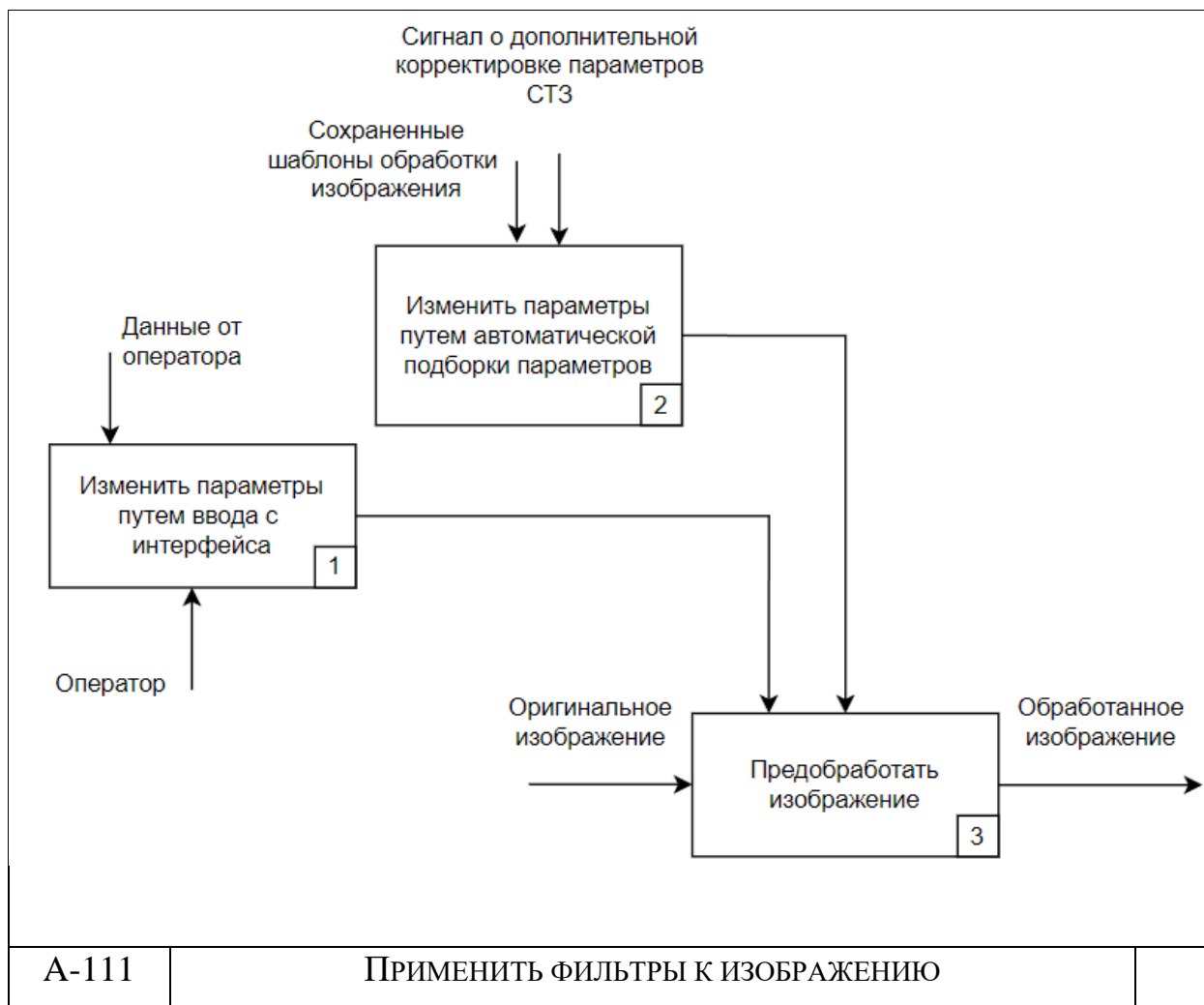


Рисунок 3.2 – Первый уровень декомпозиции алгоритма предобработки изображения

Графический интерфейс реализуется простым инструментом OpenCV при помощи ползунков, значения которых постоянно записываются в отдельный json файл, а также передаются в функцию применения фильтров. Эта функция математически подробно описана в главе 2.6. Получение изображения с камеры также просто реализуется встроенной функцией OpenCV (камера подключена по USB).

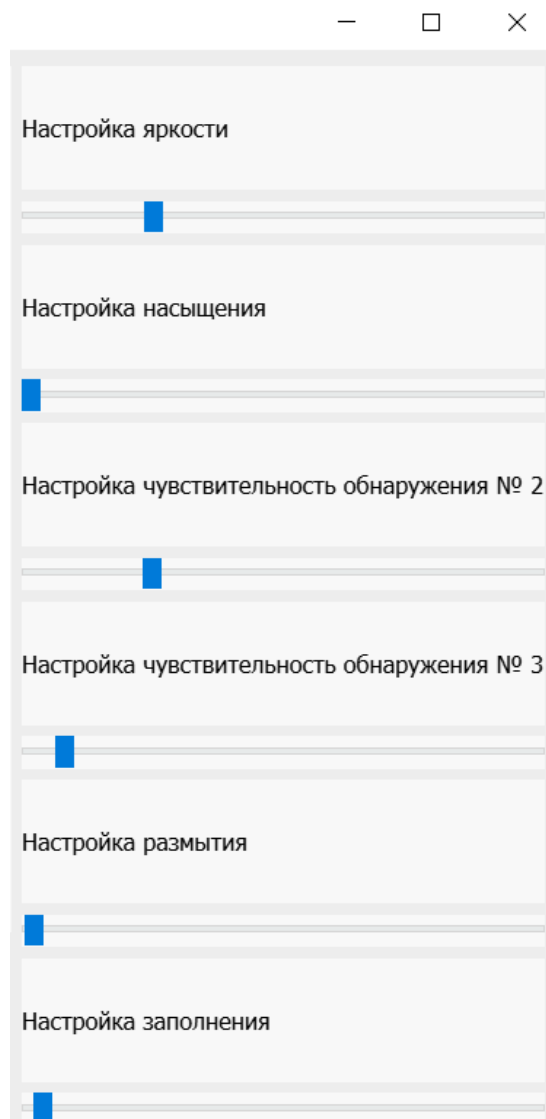


Рисунок 3.3 – Графический интерфейс настройки фильтров изображения

3.2 Архитектура системы распознавания

3.3 Разработка алгоритма движения робота

3.4 Реализация базы данных

3.5 Реализация передачи данных

3.6 Реализация графического интерфейса

3.7 Развертывание на конечной машине пользователя

3.8 Выводы по главе 3

В данной главе был спроектирован робототехнический комплекс с использованием технического зрения. Была проделана следующая работа:

- Была разработана архитектура системы распознавания объектов на видео, а также алгоритм предобработки изображения.
- Была написана программа движения робота и система безопасности его движения.
- Был разработан графический интерфейс для пользователя.
- Была спроектированная база данных

Следующий шаг – экспериментальное исследование разработанных решений.

4 Экспериментальное исследование

4.1 Программа экспериментов

4.2 Результаты экспериментов

В ходе тестирования были получены результаты, приведенные в таблице 4.2

Таблица 4.2 - Результаты экспериментов

4.3 Анализ качества алгоритма обнаружения

4.4 Выводы по главе 4

Параметр	Первый день	Второй день	Третий день	Четвертый день
Количество отсортированных паллет (каждая содержит 40 деталей) с деталями в час (кол-во)	100	80	60	40
Ошибка (кол-во деталей)	43	36	24	19
Неотсортированные детали (%)	0.0107%	0.0112%	0.01%	0.0118%
Общая скорость выполнения (с)	7,80	8,59	7,3	7,85
Общее время сортировки	8 часа 40 минут	7 часов 38 минут	4 часов 52 минут	3 часа 29 минут
Эффективность (деталей в час)	461	436	493	459

