

Приложение к отчету о выполнении проекта № 16-19-00044 «Принципы распределения задач между сервисными роботами и средствами киберфизического интеллектуального пространства при многомодальном обслуживании пользователей» в 2020 году

В ходе выполнения проекта в 2020 году были получены следующие результаты:

1. Прототип комбинированного устройства давления и приближения для анализа поверхностей, отличающийся формированием единой матричной структуры точечных емкостных датчиков давления и приближения, апробированный в шагающих роботах и других компонентах киберфизического интеллектуального пространства и позволяющий расширить потенциальный набор сценариев интерактивного взаимодействия с пользователями и объектами окружающей среды за счет введения дополнительной кинестетической модальности.

2. Подход к осуществлению захвата и дальнейшей манипуляции объектами интеллектуального пространства посредством робототехнических средств с использованием методов машинного обучения и разработанного метода получения виртуальных трехмерных представлений объектов, позволяющий средствам интеллектуального пространства не только эффективно производить манипуляции с различными объектами, но и оцифровывать их представление для последующего использования в контексте реализации сервиса мониторинга и управления пространственным положением объектов.

3. Подход к распределению задач между сервисными роботами и средствами киберфизического интеллектуального пространства при оценивании пространственных характеристик объектов, с которыми производится манипуляции и взаимодействие в процессе обслуживания пользователей.

4. Модель мониторинга и управления процессом манипуляций над объектами в киберфизическом интеллектуальном пространстве.

Рассмотрим подробнее каждый из них.

1. Прототип комбинированного устройства давления и приближения для анализа поверхностей, отличающийся формированием единой матричной структуры точечных емкостных датчиков давления и приближения, апробированный в шагающих роботах и других компонентах киберфизического интеллектуального пространства и позволяющий расширить потенциальный набор сценариев интерактивного взаимодействия с пользователями и объектами окружающей среды за счет введения дополнительной кинестетической модальности

На основе предыдущих исследований, в которых были разработаны единичные ячейки [1] и матрица из ячеек датчиков давления [2], а также единичная ячейка комбинированного датчика давления и приближения [3] была разработана матрица с ячейками комбинированного датчика (далее матричного датчика) давления и приближения следующей конструкции.

Каждая ячейка комбинированного матричного датчика давления и приближения состоит из двух основных частей: первичного емкостного преобразователя (ПЕП) и интерфейсной схемы. ПЕП преобразует изменения в приложенной к датчику силе или изменения в дистанции между датчиком и объектом в изменение электрической емкости. Интерфейсная схема служит для преобразования изменения емкости ПЕП в выходной сигнал датчика. В разработанном схемотехническом решении интерфейсная схема является общей для некоторого количества ячеек матрицы. Количество объединяемых интерфейсной схемой ячеек зависит от конкретной области применения, требований к масштабированию и может быть изменено. Выходным сигналом с ячейки матрицы является уровень напряжения в диапазоне от 0 до 3,8 В, что обеспечивает возможность напрямую получать показания с датчика посредством АЦП с высокой разрядностью. В разработанных решениях для получения показаний с ячеек матрицы применяется 16 битный АЦП.

Для обработки выходного сигнала первичного емкостного преобразователя (ПЕП) была разработана интерфейсная схема, состоящая из двух частей – генераторной и усилительной (рисунок 1).

Работа высокочастотного генератора импульсов основана на микросхеме TLC555IDR, при помощи частото задающей цепи R1R2C3 микросхема настроена на генерацию периодического сигнала прямоугольной формы (меандра) частотой 270 кГц. Усилительная часть интерфейсной схемы состоит из ПЕП, выходной сигнал которого поступает на операционный усилитель в неинвертирующем включении. Интерфейсная схема спроектирована таким образом, что количество усилительных частей схемы может быть различным, тем самым позволяя объединять первичные преобразователи давления и приближения в матрицы различных конфигураций.

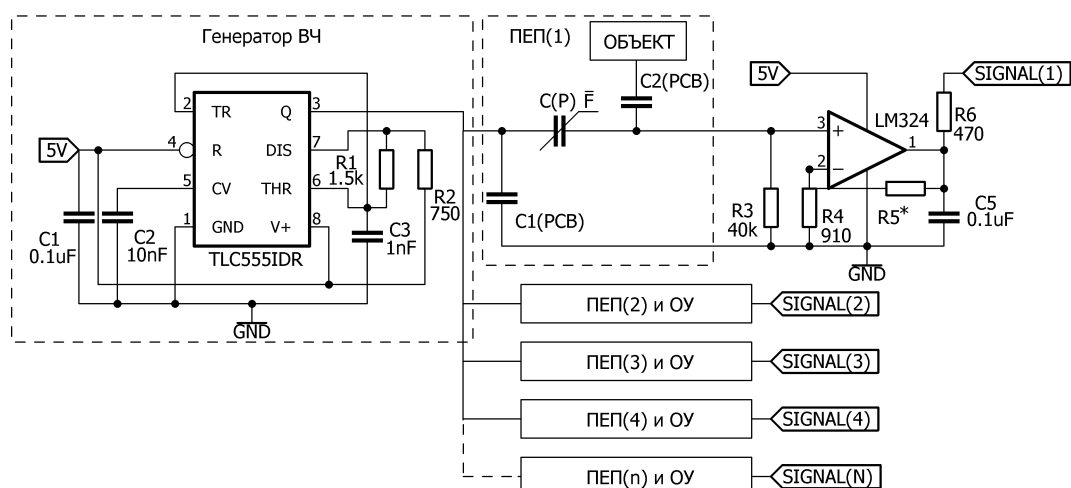


Рисунок 1 – Интерфейсная схема

Единичная ячейка представлена ПЕП выполненном в виде трехслойной структуры, габаритные размеры которой могут варьироваться в широком диапазоне в зависимости от необходимой чувствительности к давлению и приближению (рисунок 2).

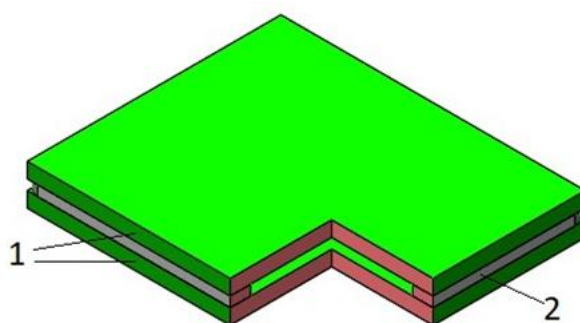


Рисунок 2 – Структурная схема ПЕП

Верхний и нижний слой структуры выполнены из фольгированного стеклотекстолита (1), где верхний слой стеклотекстолита металлизирован с двух сторон для обеспечения лучшей чувствительности ПЕП к изменению расстояния до объекта. Внутренние слои металлизации являются электродами С(Р) емкости для измерения давления. Центральный слой (2) представляет собой силиконовую полосу от толщины и высоты которой зависит чувствительность и рабочий диапазон датчика. Силиконовая полоса обеспечивает ПЕП возможностью линейно деформироваться при приложении силы, что позволяет в совокупности с линейным изменением реактивного сопротивления получать выходной сигнал, который также будет иметь линейную зависимость от приложенной силы. Эквивалентная схема замещения ПЕП представлена на рисунке 3.

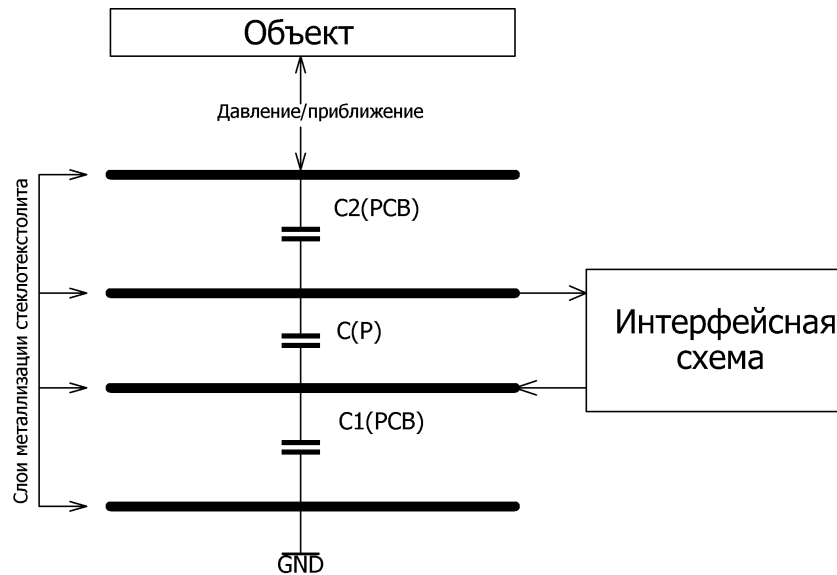


Рисунок 3 – Эквивалентная схема замещения ПЭП

На представленном выше рисунке $C1(PCB)$ и $C2(PCB)$ – емкости верхней и нижней пластин стеклотекстолита с двухсторонней металлизацией, электродами емкости являются наружный и внутренний слой металлизации пластины.

Максимальное значение конструктивной емкости ПЭП соответствует минимальному расстоянию между электродами, образующими емкость $C(P)$, и рассчитывается по формуле (1).

$$C(P) = \frac{\varepsilon \varepsilon_0 S}{d}, \quad (1)$$

где ε – диэлектрическая проницаемость зазора ППД; S – площадь электродов; d – расстояние между электродами.

При сообщении ППД внешнего давления изменяется емкость ППД и, соответственно, реактивное сопротивление, которое определяется по формуле (2).

$$X_C(P) = \frac{1}{2\pi f C} = \frac{d}{2\pi f \varepsilon \varepsilon_0 S}. \quad (2)$$

Напряжение, подводимое к неинвертирующему входу операционного усилителя, зависит от величины реактивного сопротивления ППД и значения сопротивления резистора R_3 . Сопротивление резистора R_3 влияет на скорость разряда емкости $C(P)$ и рассчитывается по формуле (3). Постоянную времени разряда емкости τ следует принимать равной четверти периода колебаний генератора ВЧ импульсов.

$$R_3 = \frac{\tau}{4C_{max}}, \quad (3)$$

Напряжение $U_{\text{вых}}$ в точке, образованной делителем $C(P)$ и R_3 , подается на неинвертирующий вход операционного усилителя (ОУ), функционирующего в режиме неинвертирующего усилителя. Коэффициент усиления операционного усилителя зависит от величины сопротивлений делителя напряжения $R_4 R_5$ и рассчитывается по формуле (4).

$$K = 1 + \frac{R_5}{R_4} \quad (4)$$

Величина выходного сигнала ОУ рассчитывается по формуле (5).

$$U_{\text{вых}} = KU_{\text{вх}} \quad (5)$$

Выходной сигнал операционного усилителя является выходным сигналом датчика и представляет собой значение постоянного напряжения, пульсации которого сглажены с помощью конденсатора С5.

Для тестирования разработанного решения был изготовлен образец матричного датчика размером 2×2 ячейки, каждая из которых имеет габаритные размеры $12 \times 12 \times 4,2$ мм. Центральный слой представляет собой силиконовую полосу шириной 2,5 мм и толщиной 1 мм, предел упругой деформации принят равным 50% от толщины воздушного зазора, равного 1 мм.

Получение характеристик экспериментального образца матрицы датчиков давления проведено с использованием испытательного стенда, конструкция которого представлена на рисунке 4.

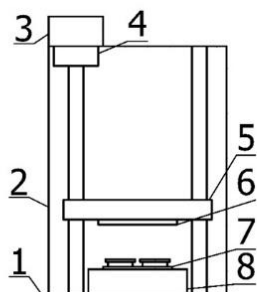


Рисунок 4 – Конструкция испытательного стенда

Испытательный стенд состоит из рамы (2), закрепленной на основании (1). Шаговый двигатель (3), расположенный в верхней части рамы, осуществляет перемещение винта передачи гайка-винт (4), приводя в движение каретку (5) с закрепленным на ней образцом (6). Каретка движется по двум направляющим осям, обеспечивая равномерное давление объекта на ПЕП. Испытательный стенд позволяет плавно и точно изменять расстояние между объектом и датчиком и силу давления на испытуемый датчик (7). Контроль и измерение силы давления осуществляется образцовым прибором измерения (8). Система управления на основе микроконтроллера обеспечивает высокую повторяемость экспериментов и позволяет перемещать каретку с точностью 1,64 мкм. Для обеспечения требуемой точности проведения экспериментов аналоговый выходной сигнал с ячеек датчика был подсоединен к АЦП с высокой разрядностью.

С помощью испытательного стенда был проведен ряд экспериментов по приближению к датчику геометрически идентичных образцов различных материалов, а также по сообщению матрице внешнего давления различной величины. Образцы из различных материалов крепятся на испытательном стенде двумя способами – без углового смещения (Рисунок 5а) и с угловым смещением 45° (Рисунок 5б).

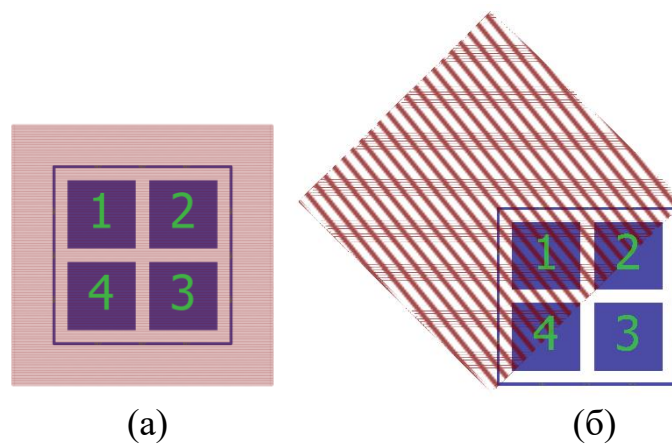


Рисунок 5 – Размещение образца над матрицей датчиков: а – без углового смещения, б – с угловым смещением

В процессе экспериментов измерялся выходной сигнал с каждой ячейки матрицы при приближении объекта к датчику. Нумерация ячеек матрицы на рисунке 5 совпадает с номерами кривых выходного сигнала с каждой ячейки на графиках ниже. Результаты экспериментов представлены в относительных единицах без перевода полученных значений в уровень напряжения. На рисунке 6 представлена зависимость выходных сигналов ПЕП при приближении объекта из алюминия. Образец был закреплен на стенде без углового смещения относительно датчика.

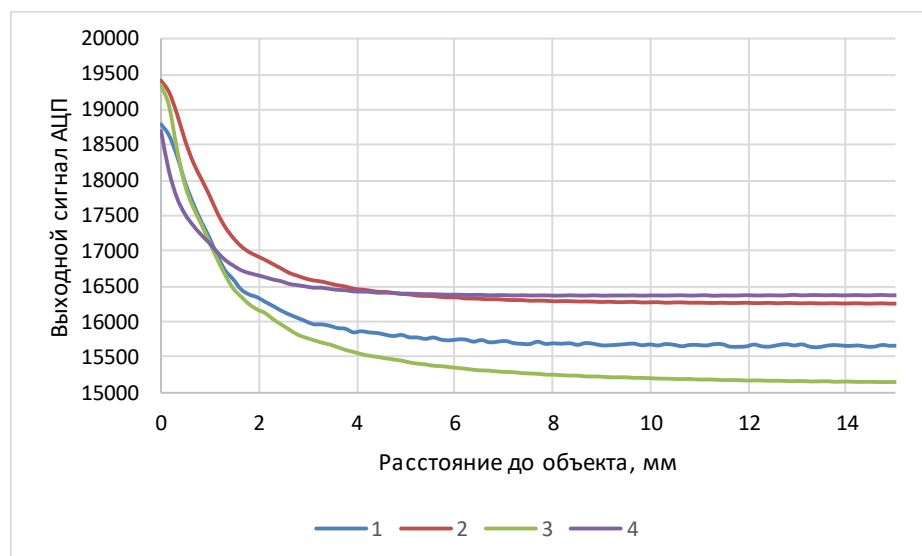


Рисунок 6 – Зависимость выходных сигналов датчика от приближения объекта без углового смещения для проводников

Изготовленная матрица датчиков может определять расстояние до объекта из проводящего материала в диапазоне расстояний до 8 мм. Данный диапазон можно расширить путем увеличения размеров ПЕП или изменением частоты работы интерфейсной схемы. Наибольшая чувствительность изготовленной матрицы датчиков наблюдается на линейном участке от 0 до 1,8 мм, чувствительность ячеек матрицы 1, 2, 3 и 4 составляет соответственно

1375, 1382, 1713 и 879 value/mm соответственно. Различия в показаниях между ячейками матрицы обусловлены неравномерной толщиной демпфирующего силиконового элемента ПЕП и особенностями его изготовления.

На рисунке 7 представлена зависимость выходных сигналов ПЕП при приближении объекта из алюминия при угловом смещении образца на 45^0 таким образом, что ячейка 3 не перекрыта образцом, 2 и 4 перекрыты частично, а ячейка 1 перекрыта полностью.

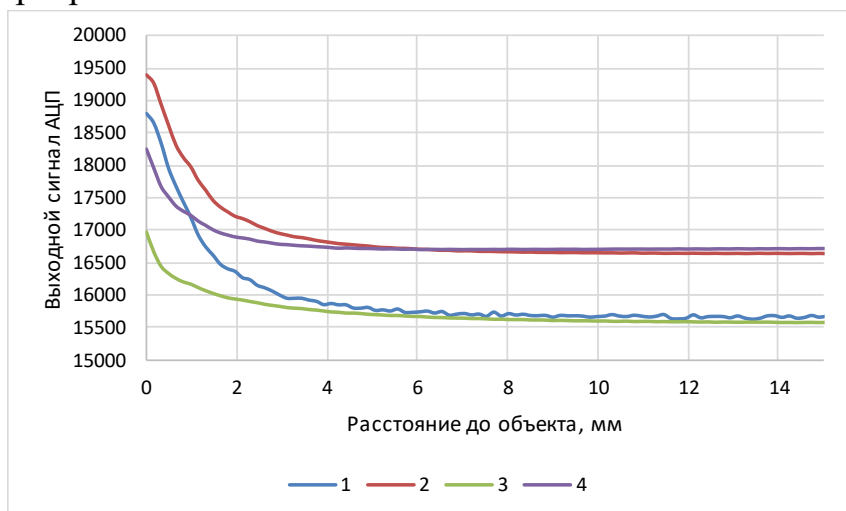


Рисунок 7 – Зависимость выходных сигналов датчика от приближения объекта с угловым смещением для проводников

Выходные сигналы полностью и частично перекрытых ячеек не имеют значительных отличий от значений при полном перекрытии ПЕП образцом. В тоже время наблюдается значительно меньшее изменение сигнала ячейки матрицы, которую в рамках данного эксперимента не перекрывал образец, чем в предыдущем эксперименте. Также были проведены эксперименты с проводящими образцами из стали и меди. Полученные характеристики сигналов имеют схожий вид с результатами на рисунках 6 и 7.

Кроме того, был проведен ряд экспериментов с образцами из диэлектрических материалов – из фанеры и из пластика. На рисунках 8 и 9 представлены зависимости выходных сигналов ПЕП от расстояния до образца из пластика без углового смещения и с угловым смещением соответственно.

Изменение в уровне выходных сигналов с ячеек матрицы при приближении диэлектрического объекта значительно меньше, чем при приближении объектов из проводящего материала. Для ячейки №4 изменение выходного сигнала при приближении диэлектрического объекта составило 585 отсчетов АЦП, а при приближении проводящего объекта 2331 отсчет.

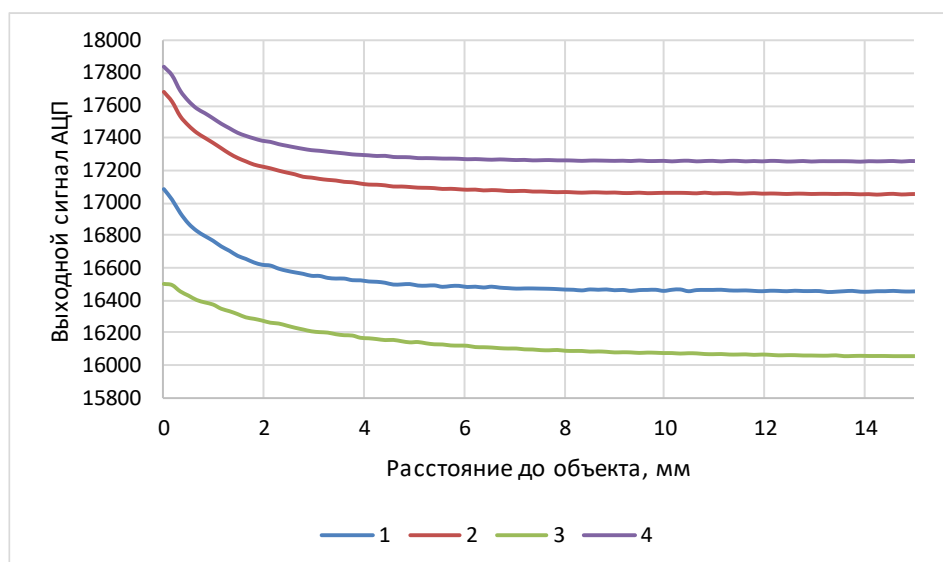


Рисунок 8 – Зависимость выходных сигналов датчика от приближения объекта без углового смещения для диэлектриков

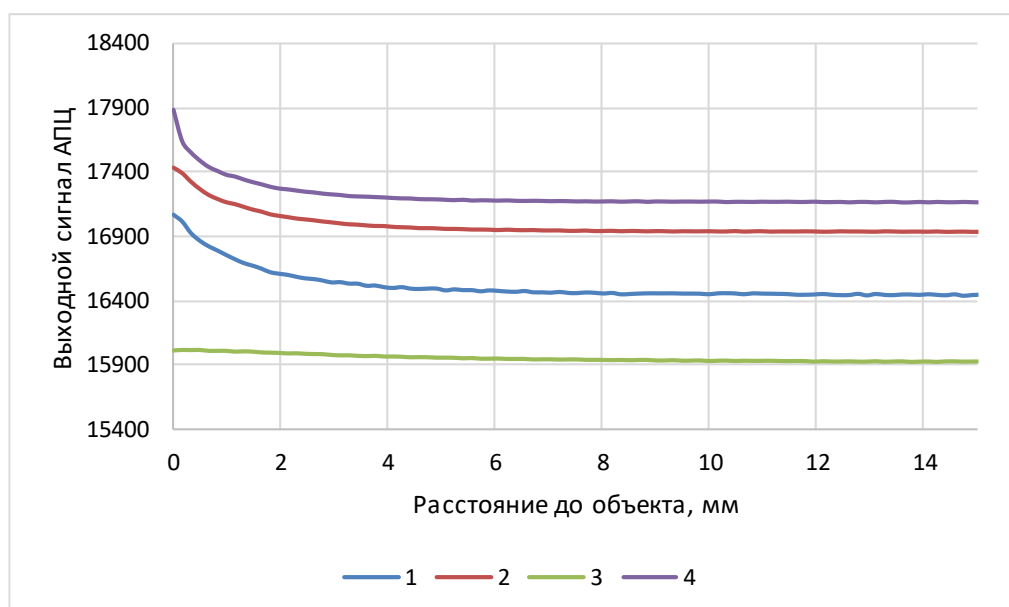


Рисунок 9 – Зависимость выходных сигналов датчика от приближения объекта с угловым смещением для диэлектриков

Как показывают результаты исследований (рис. 8 и 9), изготовленный прототип комбинированного матричного датчика может определять приближение объекта из диэлектрического материала на расстоянии до 6 мм, максимальная чувствительность ячеек матрицы в диапазоне от 0 до 1.8 mm составляет 236, 237, 127 и 238 value/mm соответственно. На рисунке 10 представлена зависимость выходного сигнала ячеек матрицы датчика при приложении к ним внешней силы.

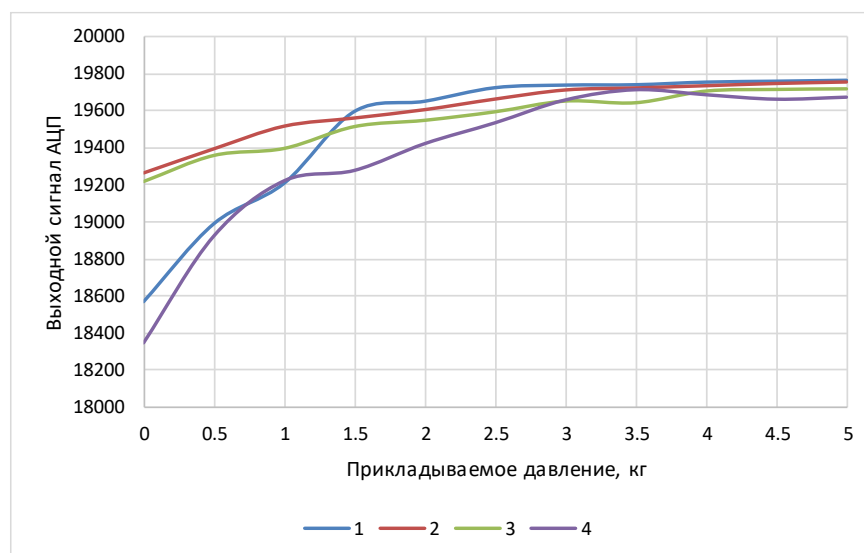


Рисунок 10 – Зависимость выходных сигналов от приложенной нагрузки

Изготовленные датчики имеют высокую чувствительность на линейном участке от 0 до 10 Н и позволяют однозначно определять приложенную к ПЕП силу давления до 25 Н. Разность показаний объясняется неидентичностью ячеек изготовленной матрицы. На рисунке 11 представлены совмещенные зависимости приближения алюминиевого объекта и последующее сообщение давления датчику со стороны объекта.

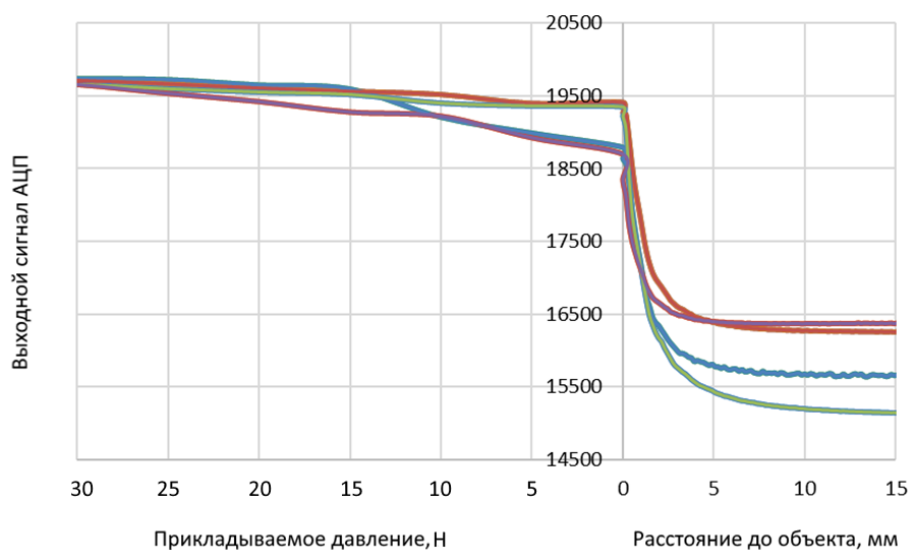


Рисунок 11 – Совмещенные зависимости приближения объекта к датчику и последующее давление

Момент касания объекта к датчику определяется однозначно. Характер графика сохраняется при использовании объектов из других материалов.

Из графиков, представленных на рисунках 6-11, можно заметить, что уровни выходных сигналов ячеек матрицы датчика отличаются друг от друга, однако при этом наблюдаемые зависимости имеют достаточно схожую форму. Данная особенность главным образом связана с неидеальным изготовлением самого датчика. Деформируемая проставка может отличаться толщиной по

площади матрицы, что приводит к различиям в емкостях ячеек. Разницу в уровнях сигналов с ячеек возможно значительно снизить, улучшив технологию изготовления датчика, а также применением первичной обработки сигналов. Введение корректирующих поправок и коэффициентов позволит получать идентичные сигналы с каждой из ячеек матрицы. Поправка вводится следующим образом. Сначала рассчитывается средний уровень для выходных сигналов – E_{av} :

$$E_{av} = \frac{E_{min} + E_{max}}{2}, \quad (6)$$

где E_{min} , E_{max} – минимальный и максимальный уровень выходного сигнала с ячеек при отсутствии объекта в зоне чувствительности датчика и отсутствии давления на датчик.

Далее рассчитывается величина поправки для выходного сигнала каждой ячейки матрицы – Δ_n :

$$\Delta_n = E_{av} - E_n, \quad (7)$$

где E_n – выходной сигнал с ячейки при отсутствии объекта в зоне чувствительности и отсутствии давления на неё.

Введение поправок в выходные сигналы ячеек позволит улучшить качество работы высокоуровневого программного обеспечения с разработанным комбинированным матричным датчиком.

В рамках проекта на базе разработанных схмотехнических и конструктивных решений были изготовлены экспериментальные прототипы датчиков различной конфигурации. На рисунке 12 представлен прототип матрицы датчиков приближения и давления из 4 ячеек.

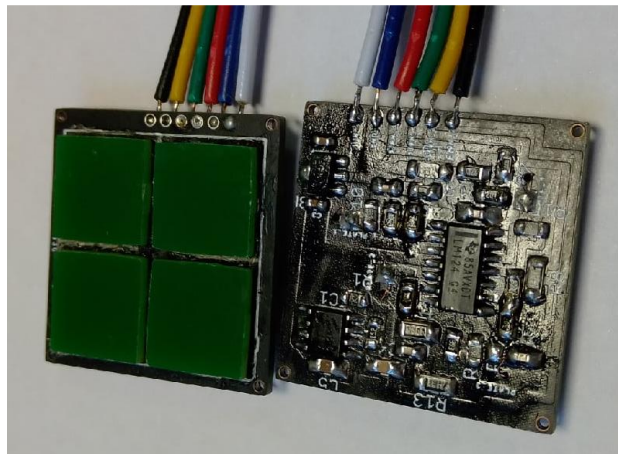


Рисунок 12 – Матричный датчик давления и приближения из 4 ячеек

Представленные выше на рисунках 6-11 зависимости выходного сигнала с ячеек матрицы при приближении объекта и оказываемом давлении были получены именно на данном прототипе.

Для использования в основании захвата манипулятора, установленного на робота «АНТАРЕС» [4, 5], была разработана матрица из 6 ячеек (рисунок 13) с размером ячейки 12 x 12 мм. Помимо матриц различной конфигурации

также разработаны прототипы единичных датчиков давления и приближения (рисунок 13), с размером 10 x 10 мм, для применения в схватах манипуляторов.

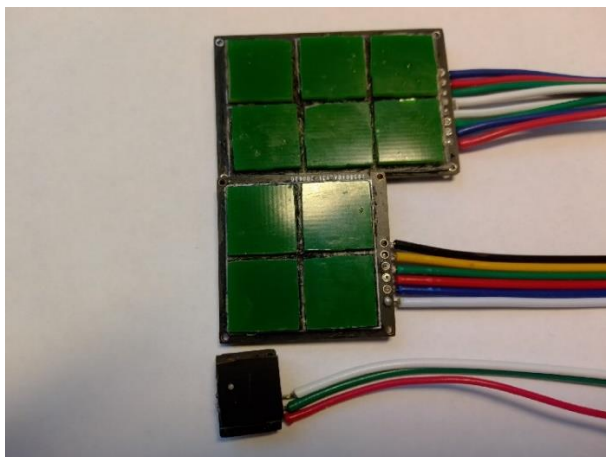


Рисунок 13 – Матрица датчиков давления и приближения из 6 ячеек в сравнении с матрицей из 4 ячеек и единичным датчиком

Возможности масштабирования разработанных схемотехнических и конструктивных решений комбинированного датчика представлены на примере матрицы из 64 ячеек размером 4 x 16. Представленный на рисунке 14 датчик помимо интерфейсной схемы включает в себя 16 битный АЦП, мультиплексоры и микроконтроллер ARM архитектуры, что позволяет получать сигналы со всех ячеек по UART интерфейсу.

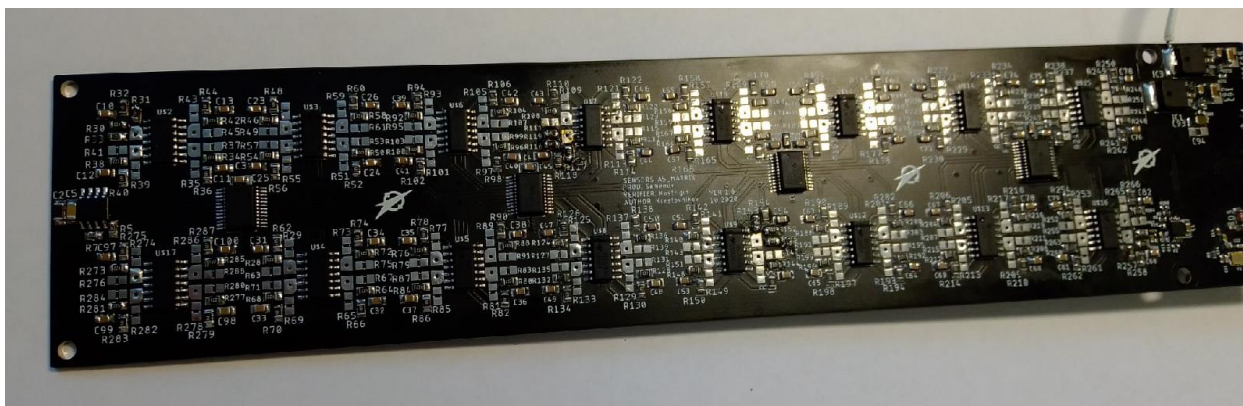


Рисунок 14 – Матрица датчиков давления и приближения на 64 ячейки

Прототип матрицы из 64 ячеек предназначен для эксплуатации в составе киберфизического интеллектуального пространства для определения местоположения объектов в закрытых и плохо обозреваемых другими системами местах.

В рамках выполнения проекта с использованием матрицы, представленной на рисунке 12, были проведены эксперименты по распознаванию материала объекта и измерению расстояния до него. Образцы из различных материалов были закреплены на испытательном стенде тремя способами – без углового смещения (рисунок 15а), с угловым смещением 45° (рисунок 15б) и в перекрытии одной ячейки (рисунок 15в).

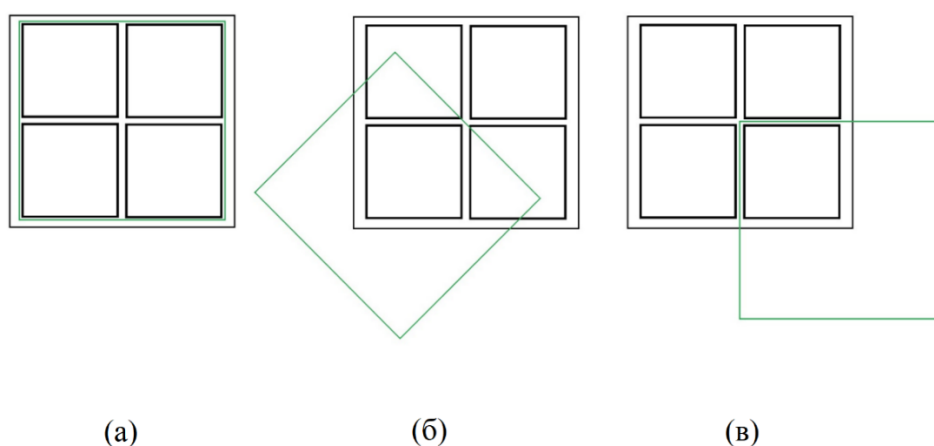


Рисунок 15 – Размещение образца над матрицей датчиков: а – без углового смещения, б – с угловым смещением 45° , в – перекрытие одной ячейки

Объекты эксперимента представляли собой пластины $50\text{мм} \times 50\text{мм}$ из следующих материалов: алюминий, дерево, медь, пластик, картон, углеволокно, вспененный полиэтилен, ткань и стекло.

По результатам проведенных экспериментов были зарегистрированы определенные отличия между кривыми сигналов, получаемых с матричного датчика для различных типов материалов. Наиболее явное различие наблюдается между кривыми проводников и диэлектриков. Исходя из данных наблюдений, для решения задачи распознавания материала объекта все объекты были разделены на две группы (проводники и диэлектрики). Целесообразность объединения материалов по электропроводности обусловлена также тем, что в реальных условиях взаимодействие РТС может происходить с совершенно различными материалами, на которых модель машинного обучения не была обучена заранее.

В связи с существенными различиями в характере графиков напряжения, полученных для различных типов материалов, не было определено обобщающее параметрическое выражение с конечным набором известных параметров, которое бы обеспечивало требуемую точность аппроксимации зависимости напряжения, генерируемого датчиком, от расстояния до объекта. Поэтому в ходе работы над проектом, были использованы методы машинного обучения и разработана специализированная модель определения типа

материала объекта и расстояния до него на основе методов машинного обучения.

Для построения соответствующей модели на основе разработанных технических решений, были апробированы следующие методы машинного обучения: метод опорных векторов (МОВ) (1), дерево решений (2), наивный байесовский классификатор (НБК) (3), случайный лес (4), логистическая регрессия (5), метод k-ближайших соседей (6), градиентный бустинг (7), модель нейронной сети (Keras, функция активации ReLU) (8).

Распознавание расстояния и материала производилось с помощью языка программирования Python 3.8 с использованием библиотек NumPy (высокоуровневые математические функции и многомерные массивы), pandas (структуры данных), scikit-learn (алгоритмы классификации, регрессии и кластеризации), XGBoost (метод градиентного бустинга), TensorFlow спецификации Keras API (библиотека для нейросетевого моделирования).

Далее представлена основная информация о параметрах, с использованием которых были реализованы методы машинного обучения. Для метода опорных векторов параметр регуляризации C (сила регуляризации обратно пропорциональна этому параметру) равен 1.0, выбрано ядро с радиальной базовой функцией (RBF), коэффициент для ядра gamma равен 0,33. В реализации дерева решений для измерения качества разделения в наборе данных используется критерий Джини, максимальная длина дерева не указана, минимальное количество выборок, необходимое для разделения внутреннего узла дерева – 2. Априорные вероятности классов для наивного байесовского классификатора не указаны, доля наибольшей дисперсии всех функций, которая добавляется к дисперсии для стабильности расчетов, равна $1e-9$.

Всего в методе случайный лес использовалось сто деревьев. Для функции оценки качества разделения в наборе данных применялся критерий Джини. Максимальная глубина дерева не указана, минимальное количество выборок, необходимое для разделения внутреннего узла – 2. В качестве регуляризации в логистической регрессии выбрана l2-регуляризация, допуск для критерия остановки – $1e-4$, параметр регуляризации – 1.0. Количество соседей в методе k-ближайших соседей равно трем. Все точки для каждого соседа в методе k-ближайших соседей имеют одинаковый вес.

В градиентном бустинге для мультиклассового разделения используется функция Softprob, для бинарного – логистическая. Максимальная глубина дерева – 6. Соотношение подвыборок обучающей выборки равно 1. Gbtree используется в качестве алгоритма предиктора, априорная вероятность прогноза для всех случаев равна 0,5.

Параметры модели нейронной сети Keras: три слоя, функция активации для первых двух слоев – ReLu, для третьего слоя – Softmax (при мультиклассовой классификации) и сигмовидная функция (при бинарной классификации), функция потерь – категориальная перекрестная энтропия (при мультиклассовой классификации) и двоичная перекрестная энтропия (при бинарной классификации), алгоритм оптимизации – алгоритм Адама,

всего 400 эпох, размер пакета (количество обучающих примеров за один проход вперед/назад) – 10.

Перед подачей данных в модели машинного обучения производилось их нормирование. Поправки для всех ячеек матричного датчика рассчитаны в соответствии с представленным ранее подходом.

Обучающая и тестовая выборки состояли из ста приближений объектов от 10 до 0 мм для каждой выборки. Всего для каждого материала и типа перекрытия производилось три или четыре, приближения от 10 до 0 мм. Пример формирования обучающей и тестовой выборок представлен на рисунке 16.

	Uout ADC1	Uout ADC 2	Uout ADC 3	Uout ADC 4	L, mm	Type of material
	19693	19761	19752	19787	0	0
	19692	19760	19750	19786	0	0
	19691	19758	19749	19784	0	0
	19691	19757	19747	19783	0	0
	19691	19756	19745	19780	0	0
	19689	19756	19745	19778	0	0
	19671	19701	19699	19709	0	1
	19672	19702	19699	19709	0	1
	19671	19701	19699	19708	0	1
	19671	19700	19698	19709	0	1
	19671	19700	19698	19707	0	1
	19670	19700	19698	19707	0	1

обучающая выборка													
feature 0	feature 1	feature 2	feature 3	feature 4	feature 5	feature 6	feature 7	feature 8	feature 9	feature 10	feature 11	L, mm	Type of material
19693	19761	19752	19787	19692	19760	19750	19786	19691	19758	19749	19784	0	0
19671	19701	19699	19709	19672	19702	19699	19709	19671	19701	19699	19708	0	1

тестовая выборка													
feature 0	feature 1	feature 2	feature 3	feature 4	feature 5	feature 6	feature 7	feature 8	feature 9	feature 10	feature 11	L, mm	Type of material
19691	19757	19747	19783	19691	19756	19745	19780	19689	19756	19745	19778	0	0
19671	19700	19698	19709	19671	19700	19698	19707	19670	19700	19698	19707	0	1

Рисунок 16 – Процесс формирования обучающей и тестовой выборок

В столбцах «Uout ADC 1 – Uout ADC 4» указаны выходные значения АЦП, измеряющего выходной сигнал каждой из четырех ячеек. В столбце «L, mm» указано расстояние до объекта, округленное до миллиметров. В «Type of material» в числовом виде представлены классы материала объекта, где 1 – диэлектрик, 0 – проводник. Выходные значения, полученные от каждой из ячеек комбинированного датчика, объединялись по три отсчета. Так, вектор признаков на основе данных от четырех ячеек состоял из двенадцати признаков (столбцы «feature 0 - feature 11»). Шаг записи данных составлял 0,016 мм, однако, поскольку требовалось осуществлять оценку расстояния до объектов с точностью до миллиметра, то для задачи оценки расстояния было сформировано одиннадцать целевых классов, соответствующих

целочисленному расстоянию до объекта в мм. При определении материала объекта (диэлектрик или проводник) соответствующая задача рассматривалась как задача бинарной классификации.

Эксперимент по распознаванию расстояния до объекта и его материала осуществлялся для трех перекрытий датчиков и двух типов материалов (диэлектриков и проводников). В первую очередь, производилось оценка качества решения задачи распознавания типа материала объекта с использованием выбранных методов машинного обучения. В таблице 1 представлены результаты распознавания материала объекта.

Таблица 1 – Результаты распознавания типа материала объекта

ММО	Доля правильных ответов, %	Точность, %	Полнота, %
НБК	70,4	67,3	61,3
Логистическая регрессия	67,2	72,1	52,3
Дерево решений	92,6	91,9	91,7
МОВ	96,2	96,7	97,8
Случайный лес	96,7	96,6	91,6
Метод k-ближайших соседей	97,1	96,9	96,6
Градиентный бустинг	85,7	85,3	82,1
Нейронная сеть Keras TensorFlow	86,2	86,7	82,1

Из таблицы 1 следует, что наиболее высокое качество решения задачи определения типа материала объекта среди апробированных методов машинного обучения демонстрируют метод k-ближайших соседей, случайный лес и МОВ.

В таблице 2 представлены результирующие оценки качества определения расстояния до объекта рассматриваемыми методами машинного обучения.

Таблица 2 – Результаты распознавания расстояния до объекта

ММО	Доля правильных ответов, %	Точность, %	Полнота, %
НБК	24,4	27,1	24,9
Логистическая регрессия	14,9	9,5	13,8
Дерево решений	62,9	61,9	61,7
МОВ	76,1	69,2	70,3
Случайный лес	70,9	72,8	73,5
Метод k-ближайших соседей	64,3	66,7	61,0
Градиентный бустинг	62,6	66,5	61,0

Нейронная сеть Keras TensorFlow	41,6	47,7	41,8
---------------------------------	------	------	------

По результатам из таблицы 2 видно, что случайный лес продемонстрировал наиболее высокую точность решения поставленной задачи. Самыми неэффективными методами, в обоих экспериментах, оказались НБК и логистическая регрессия.

Чтобы улучшить результаты определения расстояния до объекта было решено переобучить модель случайного леса, дополнив обучающую выборку информацией о типе материала, полученной с использованием модели k-ближайших соседей. Таким образом, на первом этапе производится определение типа материала посредством метода k-ближайших соседей, а затем распознанный класс материала и сигналы с датчиков подаются на вход модели на основе метода случайного леса. На выходе разработанной модели определяется расстояние до объекта в диапазоне от 1 до 10 мм.

Чтобы улучшить результаты определения расстояния до объекта, было решено объединить полученные модели распознавания материала и расстояния в единую модель. Объединённая модель по данным емкостных датчиков производит распознавание материала объекта, после этого данные о распознанном типе материала добавляются дополнительным признаком к изначальному вектору признаков (состоящему из значений напряжений от емкостных датчиков). Далее, по новому вектору признаков реализуется определение расстояния до объекта методами машинного обучения.

Таблица 3 – Результаты распознавания расстояния до объекта по значениям напряжения и типу материала

ММО	Доля правильных ответов, %	Точность, %	Полнота, %
НБК	24,3	30,9	24,6
Логистическая регрессия	20,6	18,8	13,1
Дерево решений	65,7	64,0	63,7
МОВ	77,0	70,12	71,1
Случайный лес	77,6	78,7	74,8
Метод k-ближайших соседей	76,3	78,7	73,0
Градиентный бустинг	54,3	58,5	53,3
Нейронная сеть Keras TensorFlow	49,7	53,4	48,9

Как видно из таблицы 3, алгоритм случайного леса, как и в прошлом эксперименте, обладает наиболее точной классифицирующей способностью относительно других методов. На рисунке 17 представлена матрица ошибок данного метода.

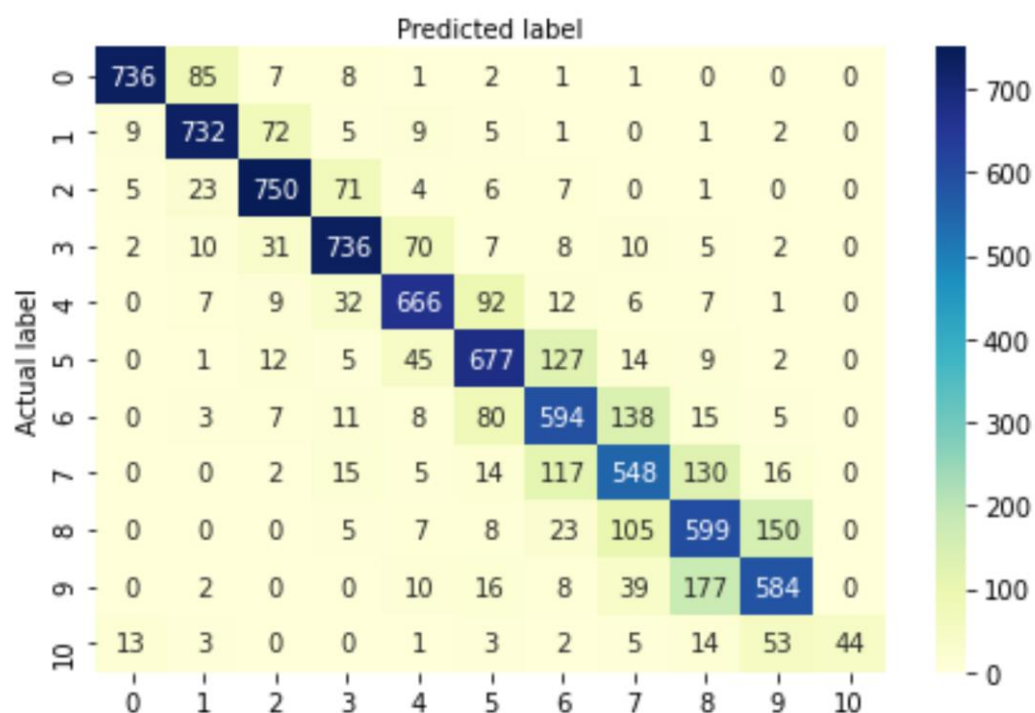


Рисунок 17 – Матрица ошибок метода случайный лес

По результатам проведенного исследования наиболее точным методом распознавания типа объекта является метод k-ближайших соседей (96,9%). Наиболее точным методом определения расстояния до объекта по данным выходного сигнала емкостных датчиков и материалу объекта является случайный лес (78,7%).

2. Подход к осуществлению захвата и дальнейшей манипуляции объектами интеллектуального пространства посредством робототехнических средств с использованием методов машинного обучения и разработанного метода получения виртуальных трехмерных представлений объектов

В рамках настоящего проекта отдельная группа работ была посвящена разработке метода захвата и дальнейшего манипулирования целевым объектом (ЦО) при совместном применении комбинированных датчиков, технического зрения и машинного обучения, что требует решения следующих основных задач:

1. Восстановление карты глубины сцены посредством искусственной нейронной сети, по кадрам, полученным с камеры.
2. Получение нейросетевым способом точки захвата объекта, соотнесенной с геометрическим центром схвата, ее положения и ориентации в опорной системе координат.
3. Построение траектории перемещения рабочего органа манипулятора робота к данной точке с заданной точностью.
4. Решение прямой и обратной задач кинематики для обеспечения системы управления перемещением звеньев манипулятора информацией о положении и ориентации схвата манипулятора.
5. Осуществить перемещение схвата манипулятора до достижения окрестности точки захвата γ , соответствующей идентификации сигналов о приближении к целевому объекту. Затем произвести корректирование ориентации схвата манипулятора, основываясь на сигналах с ячеек комбинированных датчиков давления и приближения, установленных на каждом пальце схвата.

В качестве средств реализации первых двух пунктов выступают методы и подходы технического зрения, используемые для получения кадров сцены с целью реконструкции карты глубины сцены, а также нейросетевые модели, с помощью которых может быть реализован поиск точки захвата ЦО.

Многие существующие нейросетевые методы определения точек захвата объекта основаны на анализе информации о положении и пространственных характеристиках окружающих объектов, получаемой при помощи специальных датчиков глубины, таких как LIDAR и Kinect. Сканирующие лидары и RGB-D камеры формируют карту глубины сцены, на основе которой и определяются положение и характеристики захватываемых объектов. Однако такая аппаратура обладает крайне высокой стоимостью в сравнении с бытовыми видео- и фотокамерами. Подобного недостатка лишены методы, применяющие нейронные сети [6], которые способны восстановить карты глубины сцены по 2-D изображению, полученному с обыкновенной монокулярной камеры. По результатам проведенного анализа [7] среди моделей нейронных сетей, осуществляющих построение карты глубины, была выбрана ResNet-50, дополненная Upsampling слоями,

поскольку она обладает меньшим количеством обучающих параметров и, следовательно, требует меньшего количества данных для обучения. Кроме того, данная модель характеризуется наименьшей величиной СКО, а значит, обладает наибольшей точностью предсказания выходной карты глубины, в сравнении с другими рассмотренными решениями.

Для захвата ЦО схватом манипулятора необходимо решить следующий комплекс задач технического зрения: 1) детектирование ЦО; 2) локализация ЦО на сцене; 3) определение точек захвата ЦО; 4) отслеживание траектории движения исполнительного механизма; 5) определение соответствия точек рабочей поверхности исполнительного механизма точкам схвата на ЦО; 6) детектирование успешного захвата ЦО. В анализе [7] были рассмотрены методы решения задачи определения точек захвата объекта (ТЗО), обеспечивающие возможность надежного захвата для последующего манипулирования ЦО.

ТЗО представляют собой центр сжатия объекта посредством рабочего органа манипулятора. При реализации методов определения ТЗО основная сложность заключается в наличии некоторой функциональной зависимости между положением ТЗО и такими свойствами ЦО, как: размер, геометрические особенности, свойства материала, упругость, жесткость. Неполнота данных о свойствах ЦО, а также о виде и параметрах соответствующей зависимости накладывает значительные ограничения на перспективы разработки универсальных математических моделей, обеспечивающих определение ТЗО. Таким образом, при решении задачи определения ТЗО целесообразным представляется применение нейросетевых моделей и иных средств машинного обучения для обеспечения универсальности подходов к определению ТЗО.

По результатам проведенного анализа методов и подходов к определению ТЗО для проведения сравнительной оценки были выбраны решения Grasp pose detection (GPD), 6-DOF GraspNet, Visual Pushing and Grasping (VPG), характеризующиеся наибольшими вероятностями успешного захвата ЦО.

Сравнительная оценка выбранных подходов к определению ТЗО производилась на основе тестового набора данных, включающего в себя 33000 изображений различных сцен, смоделированных в программной среде gazebo. Каждому изображению I_i из тестового набора данных соответствует набор значений параметров P_i : 1) тип ЦО; 2) размер ЦО; 3) расстояние между объективом фиксирующей видеокамеры и ЦО сцены; 4) уровень освещенности сцены; 5) угол наблюдения сцены по вертикальной оси. Для всех тестовых сцен, в качестве стандартного был выбран набор значений параметров $P_{base} = \{l_{base} = 3, d_{base} = 0,75, lum_{base} = 100\%, \alpha_{base} = 36^\circ\}$, данный набор значений параметров не зависит от типа ЦО.

Применение рассматриваемых подходов к определению ТЗО к данному набору карт глубины позволило сформировать оценки качества работы соответствующих решений в зависимости от параметров сцены и

характеристик ЦО. Для оценки качества работы данных решений использовалась вероятность успешного захвата S – доля успешных попыток захвата ЦО при использовании предлагаемых данными методами ТЗО в рамках моделирования соответствующих процессов.

Полученные оценки S для каждого из рассматриваемых методов представлены на рисунке 18. Эксперименты проводились со стандартными значениями параметров ($\mathbf{P}_i = \mathbf{P}_{\text{base}}$) ЦО.

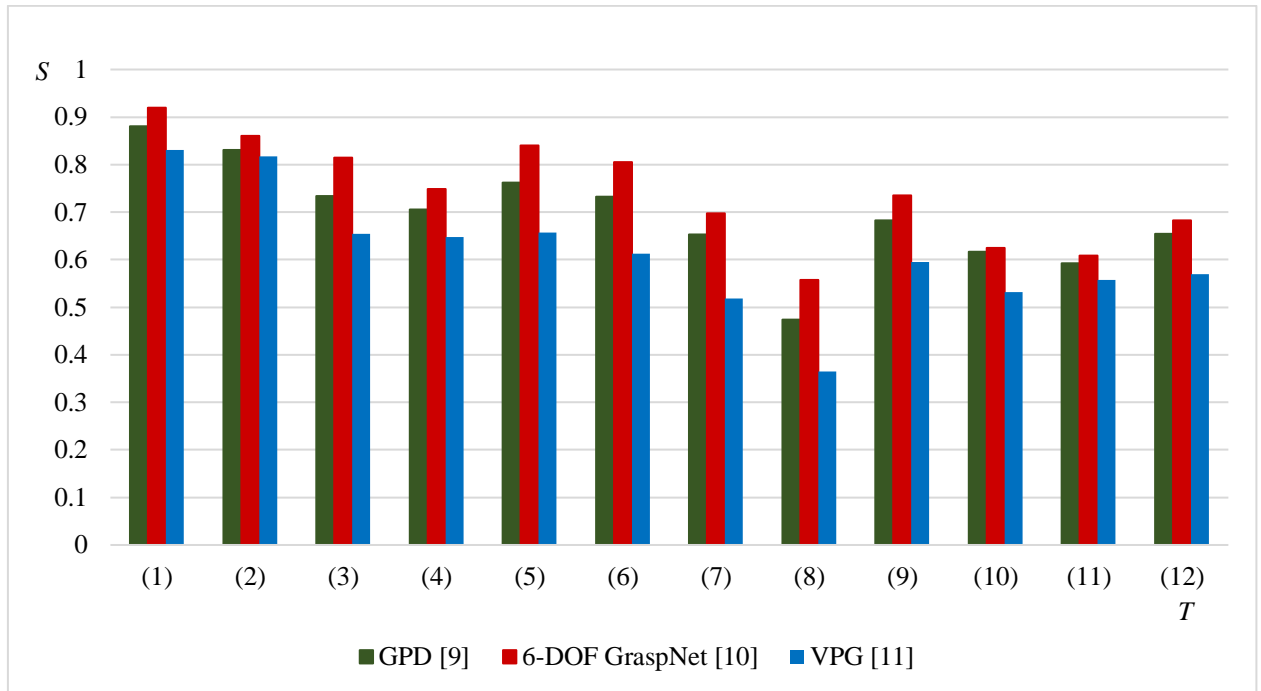


Рисунок 18 – Оценка качества работы исследуемых методов в зависимости от типа ЦО T , в стандартных условиях эксперимента ($\forall \mathbf{P}_i = \mathbf{P}_{\text{base}}$)

Анализ полученных оценок показал, что наилучшие результаты для всех типов ЦО демонстрирует подход 6-DOF GraspNet. Усредненная по типам объектов оценка вероятности успешности захвата для данного решения составила 0,741. Несколько худшее качество было установлено у метода GPD, полученная усредненная оценка в данном случае составила 0,690. Наихудшие результаты в рамках данного эксперимента продемонстрировал подход VPG, усредненная оценка S для данного метода составила 0,613.

При исследовании влияния размеров ЦО l на качество работы рассматриваемых решений набор тестовых изображений содержал варьирование размеров ЦО каждого типа, остальные параметры принимали стандартные значения из набора \mathbf{P}_{base} . Результаты проведенной серии экспериментов для каждой категории размера ЦО представлены на рисунке 19.



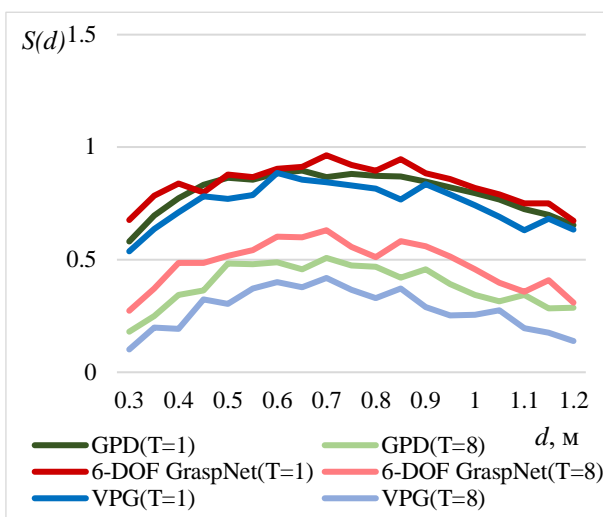
Рисунок 19 – Оценка влияния размеров ЦО l на качество работы исследуемых методов: а) $l = 1$; б) $l = 2$; в) $l = 3$; г) $l = 4$; д) $l = 5$; е) усредненная по T оценка вероятности успешного захвата ЦО в зависимости от l

По графикам, представленным на рисунке 19, было отмечено, что для всех категорий размера ЦО наилучшие результаты по-прежнему в среднем демонстрирует подход 6-DOF GraspNet, чуть худшие значения S

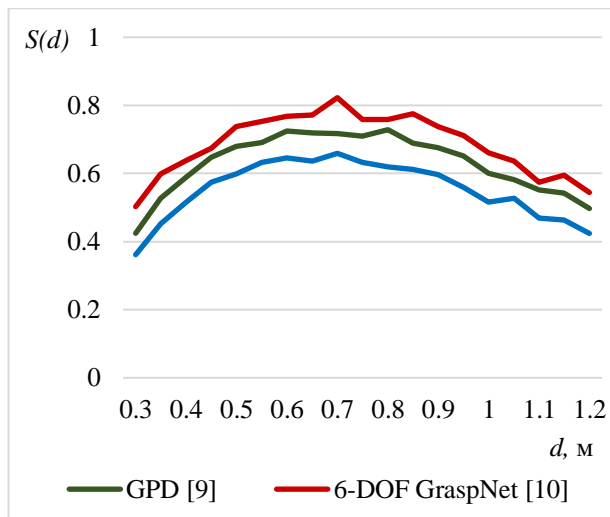
демонстрирует подход GPD, а самые низкие – метод VPG. Наблюдаемые зависимости вероятности успешного захвата от типа ЦО имели схожий вид для всех рассмотренных категорий размера ЦО. Во всех рассмотренных случаях наилучшие результаты были характерны для объектов в форме коробки и цилиндрической банки, а наихудшие – для объектов типа вазы. При переходе к меньшим и к большим размерам ЦО наблюдалось снижение данных усредненных оценок, при этом минимальные значения усредненных оценок вероятности успешности захвата были получены для категории $l = 1$ и составили, соответственно, 0,63; 0,682; 0,515. Увеличение размеров ЦО до определенного предела не оказало значительного влияния на качество восстановления карты глубины, значения усредненной оценки вероятности успешности захвата для $l = 5$ составили, соответственно, 0,679; 0,728; 0,584.

На рисунке 20 представлены полученные зависимости вероятности успешного захвата в зависимости от d , lum и α для классов объектов типа коробка и ваза: а) вероятность успешного захвата S ЦО в зависимости от d ; б) усредненные по T оценки вероятности успешного захвата ЦО в зависимости от d ; в) вероятность успешного захвата S для ЦО в зависимости от lum ; г) усредненные по T оценки вероятности успешного захвата ЦО в зависимости от lum ; д) вероятность успешного захвата S для ЦО в зависимости от α ; е) усредненные по T оценки вероятности успешного захвата ЦО в зависимости от α . При варьировании каждого параметра остальным параметрам были присвоены стандартные значения из набора \mathbf{P}_{base} .

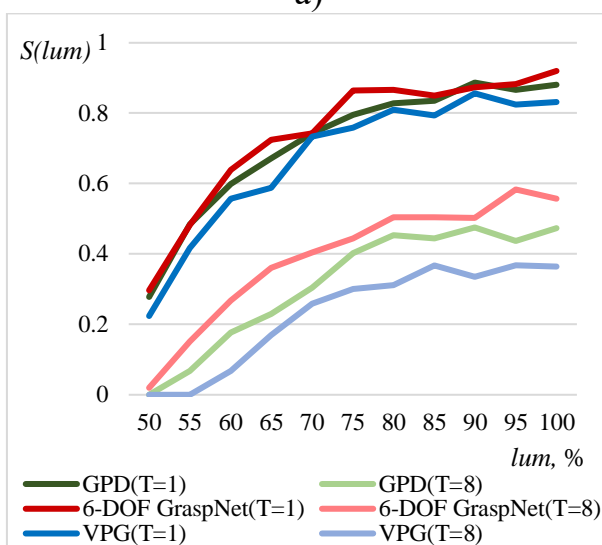
При анализе полученных результатов на рисунке 20а был сделан вывод, что зависимости обладают схожим характером независимо от используемого метода определения ТЗО и типа ЦО T : при значительном отклонении от стандартного значения параметра d в большую и в меньшую сторону наблюдалось устойчивое снижение качества работы всех рассматриваемых методов на всех типах ЦО. Усредненные по T оценки S , представленные на рисунке 3б, указывали на то, что оптимальные S для всех методов достигаются при значениях параметра d в диапазоне от 0,6 до 0,8 м. При дальнейшем увеличении d у всех методов наблюдалось плавное устойчивое снижение вероятности успешного захвата ЦО. В тоже время при уменьшении d до значений менее 0,5 м также наблюдался значительный рост темпа падения качества работы сравниваемых методов. На всем рассматриваемом диапазоне d метод 6-DOF GraspNet продемонстрировал наилучшие усредненные показатели качества работы, результирующая оценка S_{Td} вероятности успешного захвата усредненная по T и d для данного метода составила 0,685, несколько более низкие результаты демонстрирует решение GPD $S_{Td} = 0,629$, а самые слабые показатели вновь соответствуют для решения $S_{Td} = 0,552$.



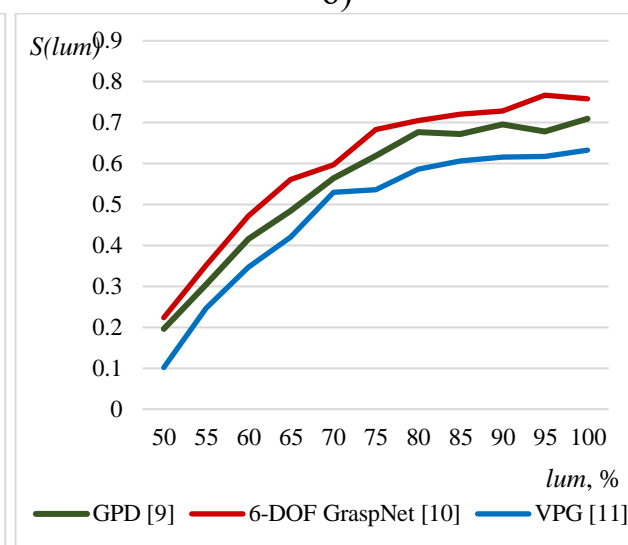
а)



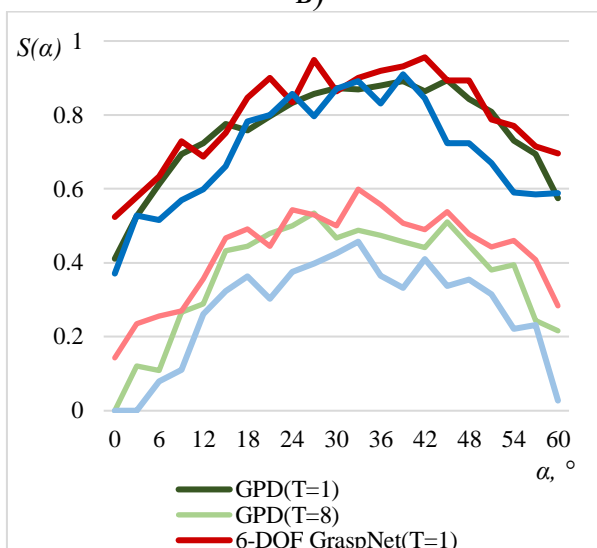
б)



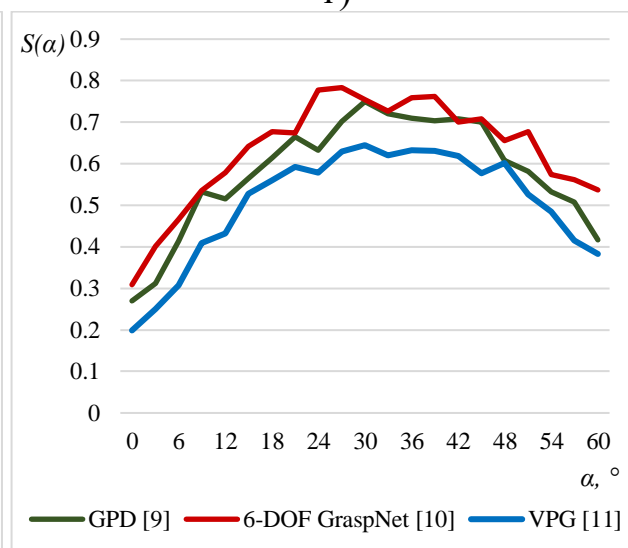
в)



г)



д)



е)

Рисунок 20 – Оценка влияния d , lum и α на качество работы исследуемых методов

По результатам исследования зависимостей вероятности успешного захвата от уровня освещенности сцены lum для ЦО типа коробка и ваза (рисунок 20в), а также аналогичных зависимостей, усредненных в разрезе каждого метода по типам ЦО T (рисунок 20г), был сделан вывод о схожести данных зависимостей независимо от используемого метода определения ТЗО и типов ЦО. При снижении уровня освещенности, начиная с некоторого момента, оценки вероятностей успешного захвата существенно снижаются, в частности, при понижении уровня освещенности с 60% до 55% усредненные по T оценки вероятности успешного захвата на всех методах дополнительно снизились более чем на 35%. Более того, для ЦО типа ваза оценки вероятности успешного захвата для всех методов при уровне освещенности уже в 50% обращались в ноль.

Аналогичным образом была исследована зависимость оценки вероятности успешного захвата ЦО от α . Полученные значения данных оценок для рассматриваемых типов ЦО в разрезе каждого рассматриваемого метода представлены на рисунке 3д, а соответствующие усредненные по T зависимости представлены на рисунке 20е. Было отмечено, что полученные зависимости характеризуются несколько более высокой вариабельностью результатов при изменении α в большую и в меньшую сторону относительно его стандартного значения. Тем не менее, во всех представленных случаях наблюдались явные снижения оценок S при отклонении α от стандартного значения в большую и меньшую стороны. На всем рассматриваемом диапазоне α метод 6-DOF GraspNet продемонстрировал наилучшие усредненные показатели качества работы, несколько более низкие результаты демонстрирует решение GPD, а самые низкие показатели вновь характерны для решения VPG.

Исходя из результатов проведенного сравнительного анализа, для решения задачи получения точки захвата объекта выбран метод 6-DOF GraspNet, который демонстрирует наилучшие показатели по доле успешных попыток захвата ЦО для подавляющего большинства комбинаций рассмотренных в работе параметров сцены. Работа данного метода происходит следующим образом. На первом этапе извлекается из полученной на вход карты глубины облако точек сцены, затем оно подвергается обработке: происходит сжатие и удаление шума из облака точек путем вокселизации, а также оптимизации на предмет наличия статистических выбросов. Далее на основе обработанного облака точек с использованием нейронной сети Grasp Sampler Network генерируются возможные положения точек захвата объектов на сцене. Полученные данные о возможных положениях точек захвата поступают на вход нейронной сети Evaluator Network, которая обучена прогнозировать вероятность успешного захвата. На заключительном этапе осуществляется оптимизация полученных результатов с использованием стохастического градиентного спуска.

Относительно полученной точки захвата определяются необходимые целевые положения и ориентация рабочего органа относительно опорной

системы отсчета манипулятора. Эта информация является входными данными блока генерации траектории разработанного алгоритма захвата объекта с применением комбинированных матричных датчиков давления и приближения, который представлен на рисунке 21.

В алгоритме приняты следующие обозначения:

1. ε – заданное отклонение между сигналами ячеек матрицы, установленной на одном из пальцев схвата. Данная величина соответствует точности ориентации схвата относительно объекта манипулирования.
2. μ – заданное отклонение между сигналами ячеек матриц датчиков, установленных на разных пальцах схвата, соответствующее точности по положению схвата относительно объекта манипулирования.
3. λ – заданное отклонение сигналов с ячеек матричных датчиков в момент касания объекта схватом манипулятора.
4. σ – отклонение сигналов с ячеек матричных датчиков, соответствующее заданному значению давления, которое необходимо достигнуть схвату манипулятора до начала манипулирования объектом.

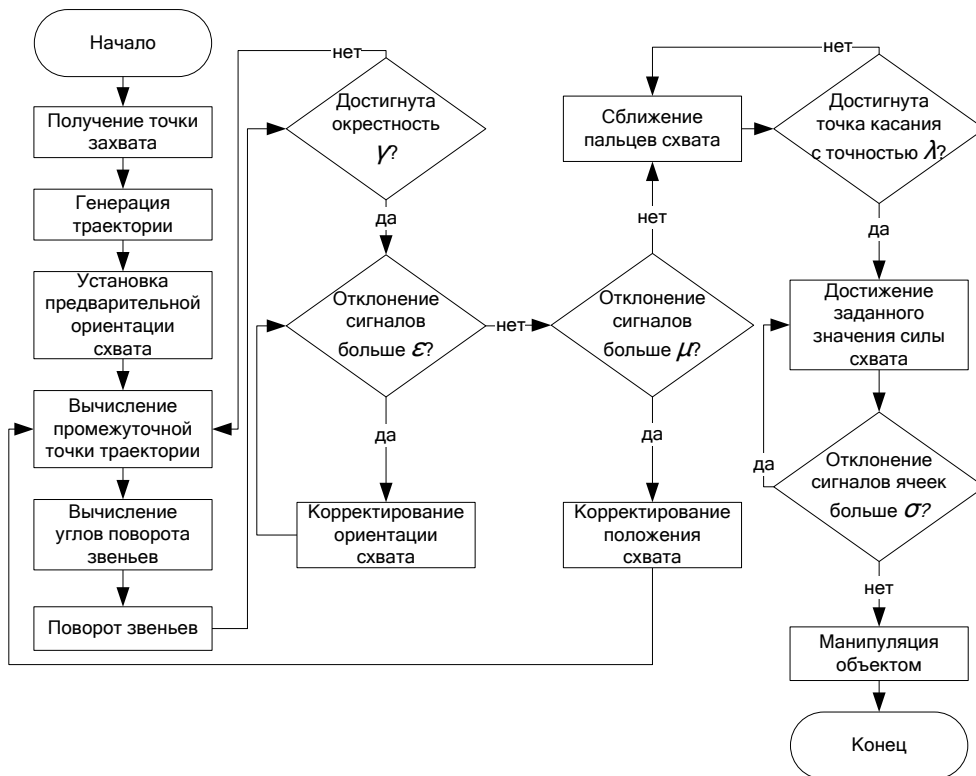


Рисунок 21 – Алгоритм работы манипулятора с установленными матрицами датчиков давления и приближения на схвате

Точка захвата объекта определяет «ориентир» для перемещения рабочего органа манипулятора и является целевой точкой, участвующей в генерации траектории перемещения схвата. В качестве планировщика траектории используется метод RRT-Connect [8], выбранный в результате ранее проведенного подробного анализа. После установки предварительной ориентации схвата происходит вычисление промежуточной точки траектории.

Затем решается обратная задача кинематики, чтобы вычислить необходимые углы поворота для перемещения рабочего органа манипулятора к промежуточным (узловым) точкам сгенерированной траектории.

При перемещении звеньев по заданной траектории происходит проверка достижения рабочим органом манипулятора окрестности точки захвата γ . Проверка осуществляется по сигналам с комбинированных датчиков давления и приближения, и дальнейшие действия направлены на достижение заданных отклонений ε и μ между сигналами ячеек матричных датчиков.

После достижения рабочим органом окрестности точки захвата целевого объекта, происходит регулирование ориентации и положения схвата по сигналам с ячеек матричных датчиков. При уровне сигналов, которые соответствуют идентификации приближения объекта, возможны следующие случаи:

1. Отклонение между сигналами ячеек матрицы одного пальца больше ε , что говорит о некорректной ориентации схвата относительно целевого объекта. Тогда производится выравнивание схвата манипулятора относительно продольной оси последнего звена по сигналам с датчика до тех пор, пока отклонение между сигналами ячеек матрицы датчиков не станет меньше ε .

2. Отклонение между сигналами ячеек матриц, расположенных на различных пальцах схвата манипулятора, больше μ . Это является признаком того, что один из пальцев схвата находится ближе к объекту, чем другой. В этом случае осуществляется корректировка положения рабочего органа манипулятора до тех пор, пока отклонение сигналов с ячеек матричных датчиков не будет меньше, чем μ .

Алгоритм работы с применением матричных датчиков давления и приближения предполагает ориентирование схвата относительно точки захвата так, чтобы она находилась посередине между пальцами с точностью ε и μ . После достижения заданной точности по положению и ориентации происходит итеративное сближение пальцев манипулятора, при этом проверяется соответствие уровней выходных сигналов ячеек матричных датчиков диапазону точки касания с точностью λ . При попадании уровней сигналов в данный диапазон алгоритм управления наращивает силу схвата до тех пор, пока не будет достигнуто определенное значение силы давления с отклонением сигналов с ячеек матричных датчиков σ .

Для точного позиционирования звеньев манипулятора необходимо выполнить его математическое описание с точки зрения кинематики, основанное на таких параметрах, как ориентация и положение звеньев манипулятора. Данное описание позволит решать прямую и обратную задачи кинематики.

Кинематическая схема руки робота изображена на рисунке 22. Точка α является геометрическим центром схвата манипулятора. Решение задачи позиционирования манипулятора направленно на достижение совпадения точки α с целевой точкой захвата объекта.

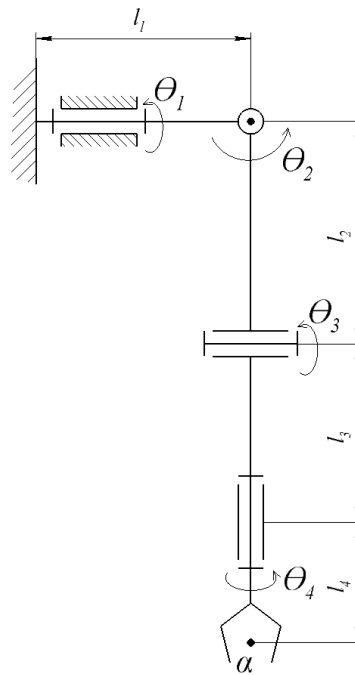


Рисунок 22 – Кинематическая схема руки-манипулятора

Управление рукой построено на основе регулирования угла поворота трех первых сочленений. Последнее сочленение отвечает за ориентацию схвата в начале алгоритма, когда требуется установить предварительную ориентацию в соответствии с информацией, полученной от системы технического зрения, а также после достижения манипулятором целевой точки. В этом случае управление ориентацией схвата происходит с учетом информации, получаемой от комбинированных датчиков давления и приближения. Последнее сочленение поворачивается относительно продольной оси звена на 2π , поэтому далее рассматривается упрощенная кинематическая схема, приведенная на рисунке 23а. На рисунке 23б также приведена привязка систем отсчета к звеньям, с помощью которой определяются кинематические уравнения в соответствии с методикой Денавита-Хартенберга. При этом не показана ось Y каждой системы координат, поскольку данные оси не влияют на используемые в методике параметры: длина звена, угол скручивания, смещение звена и угол в сочленении.

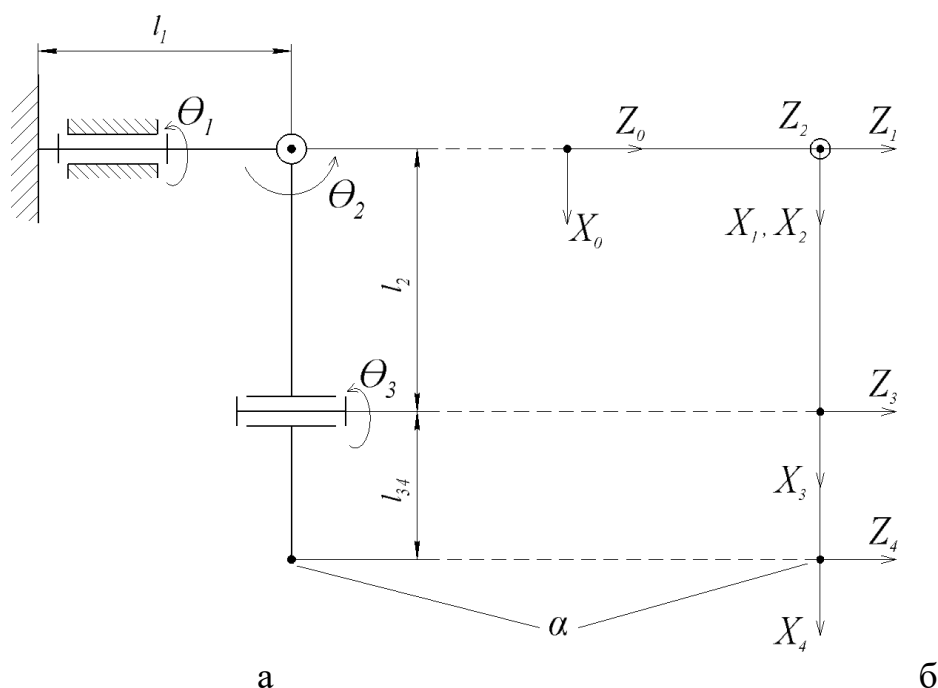


Рисунок 23 – Схематичные изображения манипулятора: а – упрощенная кинематическая схема манипулятора; б – привязка систем координат к звеньям манипулятора

Описание параметров манипулятора в соответствии с методом Денавита-Хартенберга приведено в таблице 4.

Таблица 4 – Параметры Денавита-Хартенберга руки робота «АНТАРЕС»

i	a_{i-1}	β_{i-1}	d_i	θ_i
1	0	0	l_1	θ_1
2	0	$\pi/2$	0	θ_2
3	l_2	$-\pi/2$	0	θ_3
4	l_{34}	0	0	0

Расстоянию между первым и вторым сочленениями соответствует величина l_1 , между вторым и третьим – l_2 . Расстояние от третьего сочленения до точки α объединяется в одну величину $l_{34} = l_3 + l_4$, что необходимо для работы описанного выше алгоритма. Углом поворота первого звена является величина θ_1 , второго и третьего звеньев, соответственно, θ_2 и θ_3 , последнее звено вращается относительно продольной оси четвертого звена на угол θ_4 .

Для определения описания привязанной к схвату системы отсчета {4} относительно опорной системы {0} необходимо использовать однородные преобразования. Данные преобразования представляют собой матрицы 4×4 и показывают, какую ориентацию и положение имеет каждая система i относительно предыдущей системы $i-1$. Чтобы определить ориентацию и положение системы {4} по отношению к опорной, необходимо получить матрицу, которая равна произведению матриц преобразования в порядке описания систем отсчета, то есть с системы {0} до системы {4}. Матрица однородных преобразований в общем виде записывается следующим образом:

$${}^{i-1}_iT = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 & a_{i-1} \\ \sin(\theta_i)\cos(\beta_{i-1}) & \cos(\theta_i)\cos(\beta_{i-1}) & -\sin(\beta_{i-1}) & -\sin(\beta_{i-1})d_i \\ \sin(\theta_i)\sin(\beta_{i-1}) & \cos(\theta_i)\sin(\beta_{i-1}) & \cos(\beta_{i-1}) & \cos(\beta_{i-1})d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (8)$$

Тогда описание системы $\{1\}$ относительно опорной $\{0\}$ в виде матрицы преобразования определяется как

$${}^0_1T = \begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 & 0 \\ \sin(\theta_1) & \cos(\theta_1) & 0 & 0 \\ 0 & 0 & 1 & l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (9)$$

Описание системы $\{2\}$ относительно системы $\{1\}$ в виде матрицы преобразования:

$${}^1_2T = \begin{bmatrix} \cos(\theta_2) & -\sin(\theta_2) & 0 & 0 \\ 0 & 0 & -1 & 0 \\ \sin(\theta_2) & \cos(\theta_2) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (10)$$

Описание системы $\{3\}$ относительно системы $\{2\}$ в виде матрицы преобразования:

$${}^2_3T = \begin{bmatrix} \cos(\theta_3) & -\sin(\theta_3) & 0 & l_2 \\ 0 & 0 & 1 & 0 \\ -\sin(\theta_3) & -\cos(\theta_3) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (11)$$

Описание системы $\{4\}$ относительно системы $\{3\}$ в виде матрицы преобразования:

$${}^3_4T = \begin{bmatrix} 1 & 0 & 0 & l_{34} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (12)$$

Описание системы $\{4\}$ относительно опорной $\{0\}$ определяется в виде произведения:

$${}^0_4T = {}^0_1T \cdot {}^1_2T \cdot {}^2_3T \cdot {}^3_4T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & r_{14} \\ r_{21} & r_{22} & r_{23} & r_{24} \\ r_{31} & r_{32} & r_{33} & r_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (13)$$

где

$$\begin{aligned} r_{11} &= \cos(\theta_1)\cos(\theta_2)\cos(\theta_3) - \sin(\theta_1)\sin(\theta_3), \\ r_{12} &= -\cos(\theta_3)\sin(\theta_1) - \cos(\theta_1)\cos(\theta_2)\sin(\theta_3), \\ r_{13} &= -\cos(\theta_1)\sin(\theta_2), \\ r_{14} &= l_2 \cos(\theta_1)\cos(\theta_2) - l_{34}(\sin(\theta_1)\sin(\theta_3) - \cos(\theta_1)\cos(\theta_2)\cos(\theta_3)), \\ r_{21} &= \cos(\theta_1)\sin(\theta_3) + \cos(\theta_2)\cos(\theta_3)\sin(\theta_1), \end{aligned}$$

$$\begin{aligned}
r_{22} &= \cos(\theta_1) \cos(\theta_3) - \cos(\theta_2) \sin(\theta_1) \sin(\theta_3), \\
r_{23} &= -\sin(\theta_1) \sin(\theta_2), \\
r_{24} &= l_2 \cos(\theta_2) \sin(\theta_1) + l_{34} (\cos(\theta_1) \sin(\theta_3) + \cos(\theta_2) \cos(\theta_3) \sin(\theta_1)), \\
r_{31} &= \cos(\theta_3) \sin(\theta_2), \\
r_{32} &= -\sin(\theta_2) \sin(\theta_3), \\
r_{33} &= \cos(\theta_2), \\
r_{34} &= l_1 + l_2 \sin(\theta_2) + l_{34} \cos(\theta_3) \sin(\theta_2).
\end{aligned}$$

В результате преобразований получаются кинематические уравнения, которые используются при решении прямой и обратной задач кинематики. Решение прямой задачи необходимо для определения ориентации и положения систем отсчета по известным значениям углов в сочленениях, в то время как обратная задача кинематики требует определения этих углов при известных желаемых координатах для перемещения схвата манипулятора.

Компонентам r_{12} , r_{24} , r_{34} матрицы 0T соответствуют координаты x , y , z системы отсчета $\{4\}$ в координатах опорной системы $\{0\}$. Путем подстановки величин углов в сочленения решается прямая задача кинематики – определяются координаты x , y , z .

Обратная задача кинематики решается геометрически со следующими ограничениями на диапазон вариации углов в сочленениях (обобщенных координат):

$$\theta_1 \in [-\frac{\pi}{9}; 1.38\pi), \theta_2 \in [0; \pi), \theta_3 \in [0; \pi). \quad (14)$$

На рисунках 24 а и 24 б в упрощенном представлении демонстрируются виды манипулятора в двух плоскостях – XY и XZ .

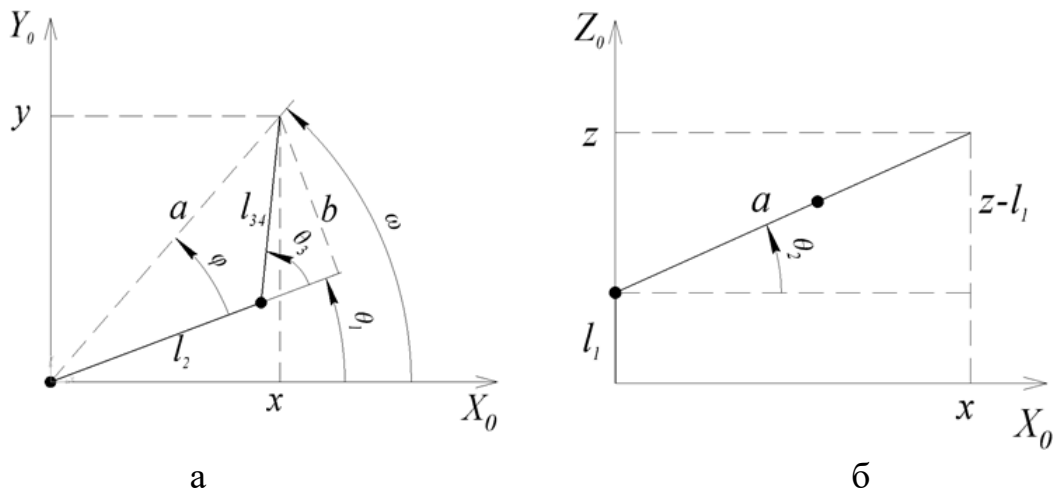


Рисунок 24 – Изображения манипулятора: а – в плоскости XY ; б – в плоскости XZ

Угол между стороной a и звеном с длиной l_2 в соответствии с теоремой косинусов вычисляется по следующему выражению:

$$\varphi = \tan^{-1} \left(\frac{\pm \sqrt{1 - \cos^2(\varphi)}}{\cos(\varphi)} \right), \quad (15)$$

где $\cos(\varphi) = \frac{a^2 + l_2^2 - l_{34}^2}{2al_2}$, $a = \sqrt{x^2 + y^2 + (z - l_1)^2}$.

Знак в числителе (15) определяется положением целевой точки в опорной системе координат.

Синус угла θ_3 вычисляется через величину b :

$$\sin(\theta_3) = \frac{b}{l_{34}}, \quad (16)$$

где $b = a \cdot \sin(\varphi)$.

Угол θ_3 определяется через функцию арктангенса двух аргументов:

$$\theta_3 = \tan^{-1} \left(\frac{\sin(\theta_3)}{\pm \sqrt{1 - \sin^2(\theta_3)}} \right). \quad (17)$$

На выбор знака в знаменателе (17) влияет координатная четверть опорной системы отсчета, в которой находится целевая точка.

Синус угла θ_2 выражается из r_{34} :

$$\sin(\theta_2) = \frac{z - l_1}{l_2 + l_{34} \cdot \cos(\theta_3)}. \quad (18)$$

С учетом выражения (18) рассчитывается величина угла θ_2 по формуле (19):

$$\theta_2 = \begin{cases} \tan^{-1} \left(\frac{\sin(\theta_2)}{\cos(\theta_2)} \right), & \text{при } \text{sign}(x \cdot y) = +1, \\ \pi - \tan^{-1} \left(\frac{\sin(\theta_2)}{\cos(\theta_2)} \right), & \text{остальное,} \end{cases} \quad (19)$$

где $\cos(\theta_2) = \pm \sqrt{1 - \sin^2(\theta_2)}$.

Знак в (20) выбирается аналогично случаю с углом θ_3 .

Из-за $\sin^2(\theta_2) = 1$ решение (20) для угла θ_2 будет иметь деление на ноль. Тогда при условии выполнения ограничения, что θ_2 строго меньше π , принимается $\theta_2 = \pi/2$.

Угол θ_1 находится в зависимости от знака x и y :

$$\theta_1 = \begin{cases} \omega - \varphi', & \text{при } y \geq 0 \text{ и } \forall \text{sign}(x), \\ \pi - \omega + \varphi', & \text{остальное,} \end{cases} \quad (20)$$

где $\omega = \tan^{-1} \left(\frac{y}{x} \right)$, $\varphi' = \tan^{-1} \left(\frac{\sin(\varphi')}{\sqrt{1 - \sin^2(\varphi')}} \right)$ – угол наклона проекции отрезка с

длиной a на плоскость X_0Y_0 , $\sin(\varphi') = \frac{b}{\sqrt{x^2 + y^2}}$.

Решение обратной задачи кинематики необходимо для определения таких углов поворота в сочленениях, при которых рабочий орган достигнет целевой точки захвата объекта. Однако при этом следует контролировать

промежуточные значения углов, т.е. выбрать вид закона управления, по которому будет происходить изменение обобщенных координат. Для этого используется планировщик траекторий, который, в зависимости от присутствия ограничений на перемещение, осуществляет построение траектории изменения углов в сочленениях. Формируемая траектория разбивается на узловые точки, с помощью которых происходит вычисление промежуточных значений углов согласно полученным решениям прямой и обратной задач кинематики. При этом для каждого сочленения строится гладкая функция, которая будет проходить через N узловых точек. В качестве такой функции может быть выбрана линейная функция с параболическими переходами. Использование данной формы траектории обусловлено тем, чтобы скорость на линейном участке перехода изменялась плавно, т.е. с постоянным ускорением, при этом траектория получается с непрерывным положением и скоростью.

Если рассматривать соседние точки траектории j, k, l (см. рисунок 25), то скорость изменения угла в сочленении в промежуточных точках будет равна:

$$\dot{\theta}_{jk} = \frac{\theta_k - \theta_j}{t_{djk}}, \quad (21)$$

где t_{djk} – общая длительность сегмента, который соединяет точки j и k .

Ускорение на переходном участке вычисляется по формуле (22).

$$\ddot{\theta}_k = \text{sign}(\dot{\theta}_{kl} - \dot{\theta}_{jk}) |\ddot{\theta}_k|, \quad (22)$$

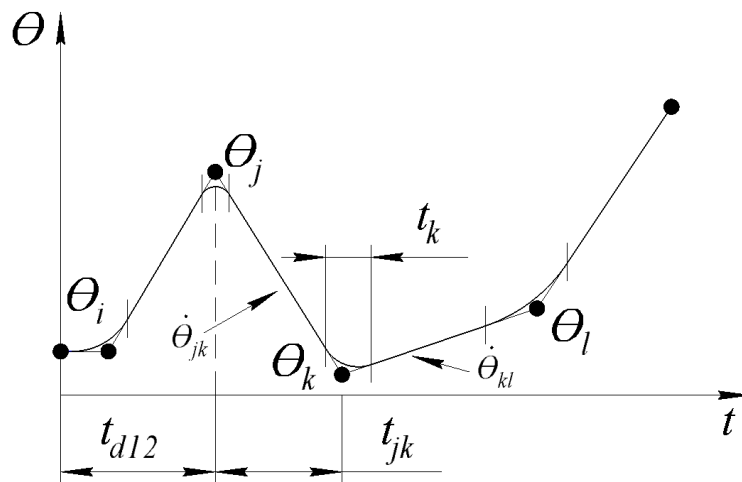


Рисунок 25 – Многосегментная линейная траектория с параболическими переходами

Ускорению на переходном участке соответствует длительность перехода, определяемая по выражению:

$$t_k = \frac{\dot{\theta}_{kl} - \dot{\theta}_{jk}}{\ddot{\theta}_k}. \quad (23)$$

Длительность линейного участка t_{jk} рассчитывается так:

$$t_{jk} = t_{djk} - \frac{1}{2}t_j - \frac{1}{2}t_k. \quad (24)$$

Выражения для первого и последнего фрагментов траектории выглядят иначе, поскольку при определении длительности можно учесть один из участков перехода, расположенный, соответственно, на конце первого фрагмента и в начале последнего фрагмента траектории. Длительность перехода на конце первого фрагмента траектории определяется в соответствии с выражением:

$$t_1 = t_{d12} - \sqrt{t_{d12}^2 - \frac{2(\theta_2 - \theta_1)}{\ddot{\theta}_1}}. \quad (25)$$

Скорость на линейном участке первого фрагмента рассчитывается в соответствии с формулой:

$$\dot{\theta}_{12} = \frac{\theta_2 - \theta_1}{t_{d12} - \frac{1}{2}t_1}. \quad (26)$$

Время на линейном участке первого фрагмента можно определить по выражению:

$$t_{12} = t_{d12} - t_1 - \frac{1}{2}t_2. \quad (27)$$

Похожие выражения используются для последнего фрагмента, границам которого соответствуют точки с индексами $N-1$ и N . Длительность перехода на конце последнего фрагмента траектории определяется в соответствии с выражением:

$$t_N = t_{d(N-1)N} - \sqrt{t_{d(N-1)N}^2 - \frac{2(\theta_N - \theta_{N-1})}{\ddot{\theta}_N}}. \quad (28)$$

Скорость на линейном участке последнего фрагмента рассчитывается в соответствии с формулой:

$$\dot{\theta}_{(N-1)N} = \frac{\theta_N - \theta_{N-1}}{t_{d(N-1)N} - \frac{1}{2}t_N}. \quad (29)$$

Время на линейном участке последнего фрагмента вычисляется по выражению:

$$t_{(N-1)N} = t_{d(N-1)N} - t_N - \frac{1}{2}t_{N-1}. \quad (30)$$

Разработанный алгоритм позволяет по полученной с помощью технического зрения и искусственных нейронных сетей информации о точке захвата построить траекторию перемещения схвата манипулятора к целевой точке захвата с заданной точностью ε и μ . Использование полученного аналитического решения прямой и обратной задач кинематики обеспечивает вычисление необходимых углов поворота каждого звена по промежуточным точкам сгенерированной траектории. Внедрение в пальцы схвата манипулятора комбинированных матричных датчиков давления и приближения позволяет корректировать ориентацию и положение схвата

манипулятора с заданными точностями при приближении его к целевому объекту и при захвате объекта с учетом заданных отклонений λ и σ .

Матрица комбинированных датчиков давления и приближения из 4 ячеек, представленная на рисунке 12 была установлена в каждый палец двупалого захвата робота «АНТАРЕС» (рисунок 26).

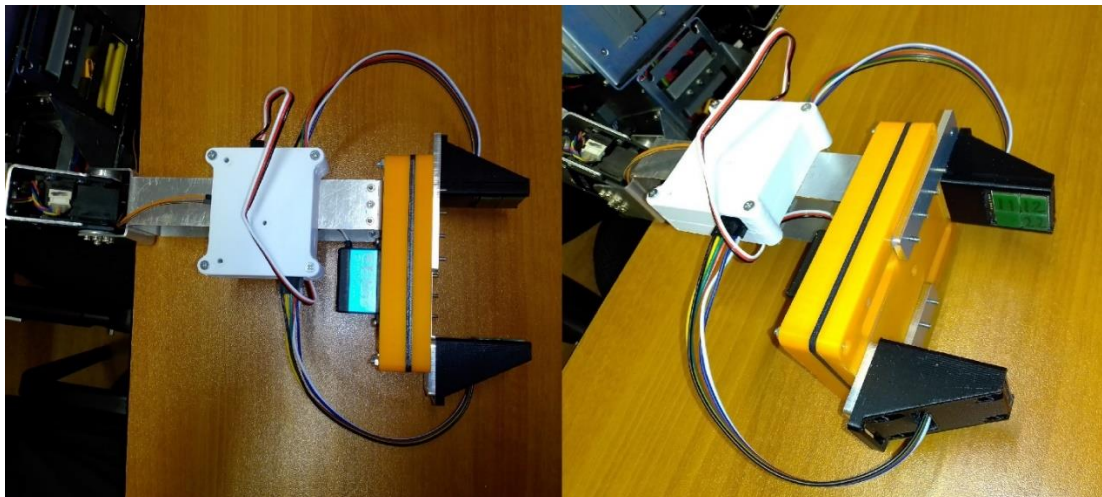


Рисунок 26 – Матрица датчиков давления и приближения в манипуляторе робота «АНТАРЕС»

Проведен ряд экспериментов по захвату объектов из проводящих и диэлектрических материалов, с различным расположением объекта относительно схвата манипулятора. В первом эксперименте объект находился в геометрическом центре схвата (рисунок 27 а) и полностью перекрывал площадь всех ячеек матричных датчиков. Второй эксперимент предполагал частичное перекрытие площади матрицы при котором объект был смещен относительно геометрического центра манипулятора (рисунок 27 б).

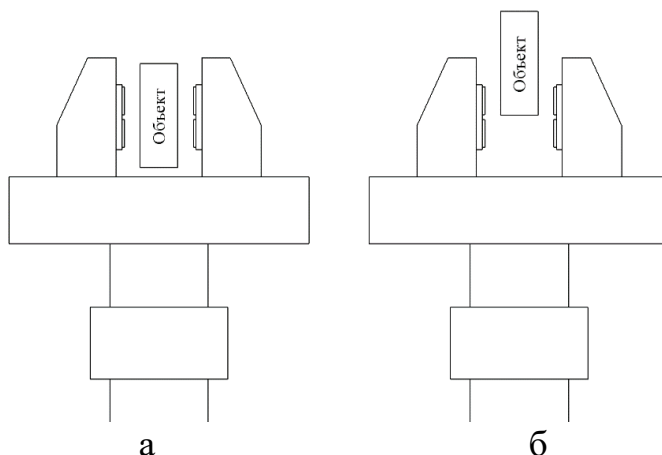


Рисунок 27 – а) объект находится в геометрическом центре схвата, б) объект смещен относительно геометрического центра схвата

На рисунке 28 представлен график выходных сигналов с ячеек каждой матрицы без нормирования при захвате объекта из алюминия, находящегося в геометрическом центре манипулятора.

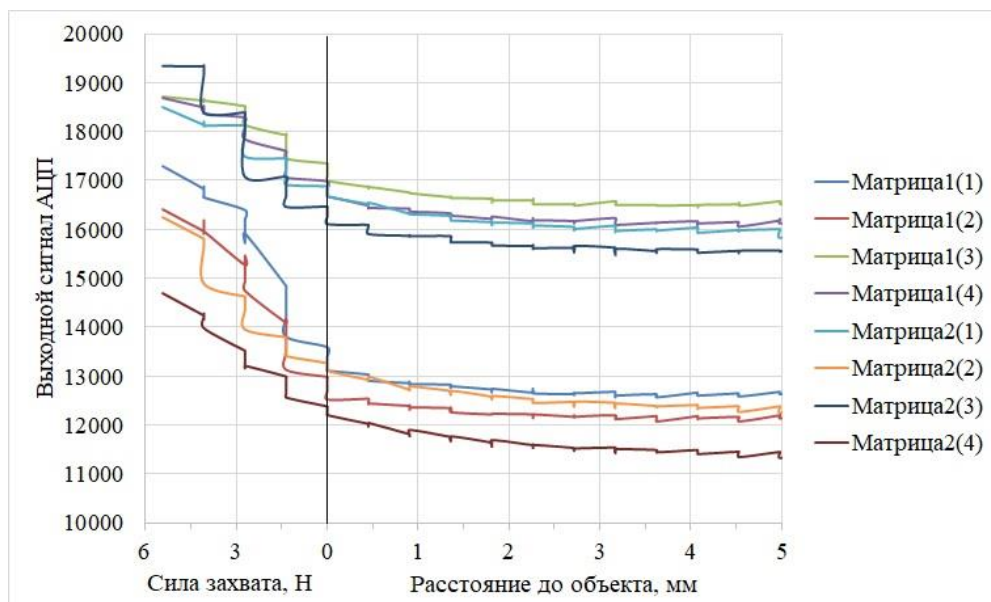


Рисунок 28 – Выходные сигналы с каждой ячейки матриц при захвате объекта из алюминия

Точка «0» на горизонтальной оси соответствует касанию объекта пальцами манипулятора. Слева от точки касания горизонтальная ось размечена в единицах силы давления, приложенного к объекту со стороны манипулятора, а справа в единицах расстояния между объектом и пальцем захвата. Кривые имеют ярко выраженный ступенчатый характер, так как захват объекта с применением обратной связи с комбинированных датчиков является итеративным процессом, а сервопривод, примененный в схвате манипулятора, позволяет сдвигать пальцы дискретно с шагом 0,453 мм. На рисунке 29 представлен график зависимостей для объекта из ABS пластика, расположенного таким же образом.

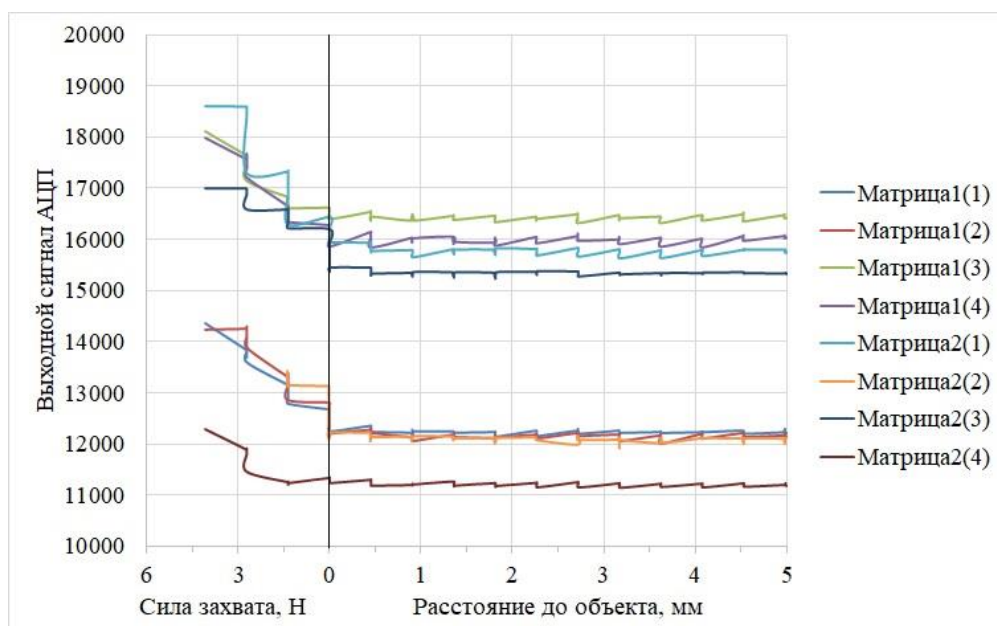


Рисунок 29 – Выходные сигналы с каждой ячейки матриц при захвате объекта из ABS пластика

При захвате объекта из ABS пластика можно видеть значительно меньшие изменения в сигналах ячеек при приближении к объекту, чем для проводящего объекта из алюминия.

Согласно результатам проведенных экспериментов, результаты которых представлены выше на рисунках 28 и 29, установленные в манипулятор матрицы позволяют фиксировать и определять приближение к тестовому объекту на расстоянии до 5мм и измерять силу давления до 6 Н. Данные параметры возможно корректировать, изменяя конструктивные элементы ПЕП.

Результаты экспериментов по захвату объекта, смещенного относительно геометрического центра схвата манипулятора представлены на рисунках 30 и 31.

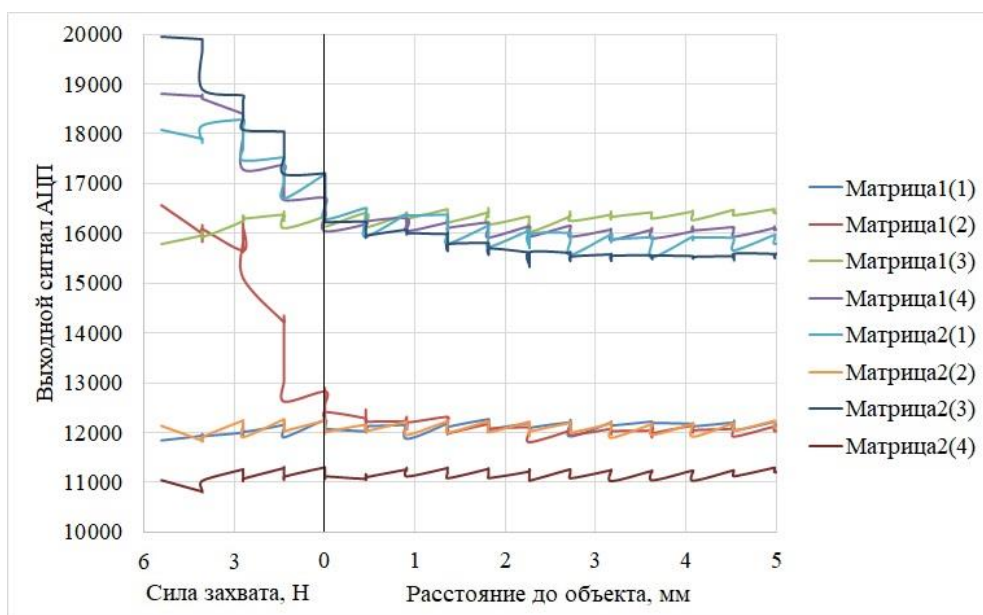


Рисунок 30 – Выходные сигналы с матриц при захвате объекта со смещением из алюминия

Из рисунка 30 видно, что возможно однозначно определить приближение к проводящему объекту и его положение относительно схвата манипулятора. Ячейки матрицы на против которых располагался объект имеют значительные изменения в выходном сигнале при приближении к нему. После касания с объектом однозначно определяются точки контакта и возможно измерить силы давления в них. Ячейки матрицы, площадь которых не перекрывалась объектом не имеют закономерных изменений в выходном сигнале, что позволяет делать вывод о том, что объекта захвата на против них не было.

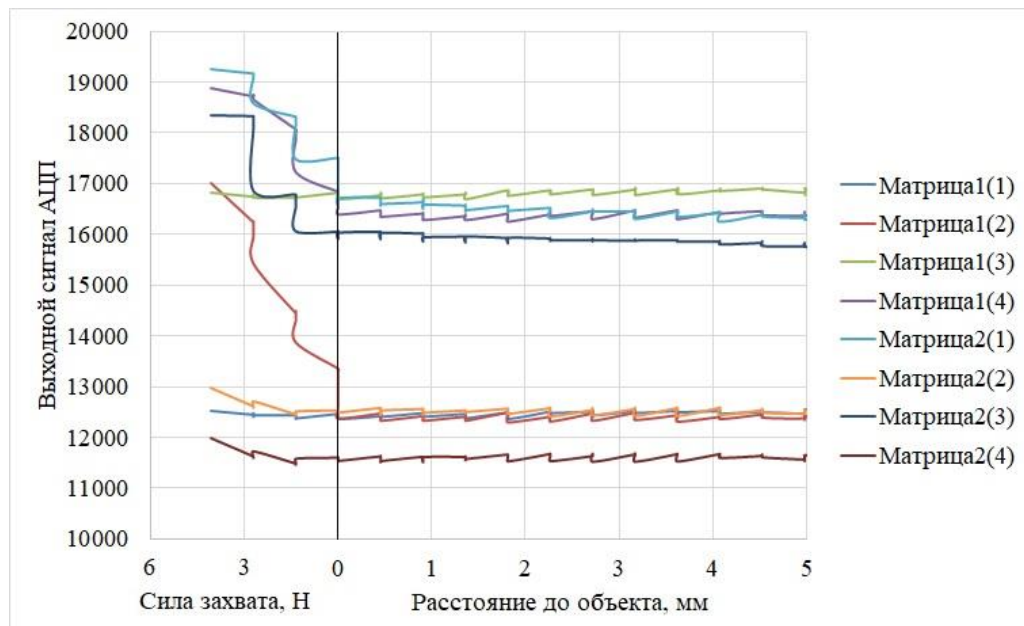


Рисунок 31– Выходные сигналы с матриц при захвате объекта со смещением из ABS пластика

Реакция ячеек на приближение к диэлектрическому объекту значительно меньше, чем при приближении к проводящему, но также позволяет однозначно определить на против каких ячеек матрицы находится захватываемый объект. Диапазон изменений в сигналах ячеек матрицы после касания с объектом из ABS пластика схож с результатами при захвате объекта из алюминия.

Сравнивая графики частичного и полного перекрытия площади матрицы объектом при его захвате, возможно сделать вывод, что разработанный комбинированный матричный датчик давления и приближения позволяет однозначно определять положение объекта из проводящих и диэлектрических материалов относительно схвата манипулятора до касания объекта и в момент захвата. Матрица датчиков позволяет определить точки касания с объектом и измерить контактные силы в каждой из них. Информация о силе давления схвата позволяет регулировать её, и предотвращать выскальзывание или деформацию объекта манипулирования.

В рамках проекта также было выполнено внедрение комбинированных датчиков давления и приближения в стопы педипуляторов антропоморфного робота, что потребовало разработки специализированного алгоритма походки, который был бы способен учитывать информацию с данных сенсоров при расчете параметров механического движения. На рисунке 32 представлен разработанный алгоритм перемещения педипулятора на один шаг малой длины относительно исходного состояния, соответствующего устойчивому положению робота на двух педипуляторах. В алгоритме приняты следующие обозначения:

1. φ – заданная точность, соответствующая допустимому отклонению сигналов с матричных датчиков по приближению стопы педипулятора к поверхности перемещения робота;

2. θ – заданная точность, которая равна допустимому отклонению расчетного значения точки нулевого момента (ТНМ) от желаемого.

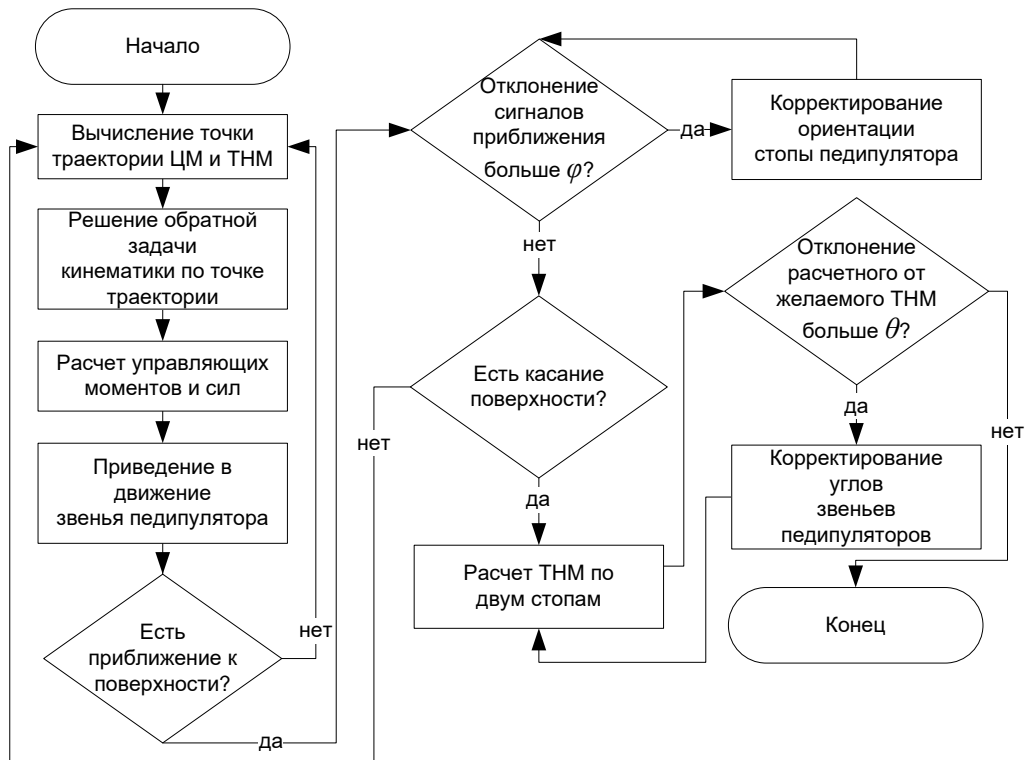


Рисунок 32 – Алгоритм выполнения шага одним педипулятором

Обеспечение движения робота за счет педипуляторов предполагает наличие информации о желаемых значениях углов звеньев, положения центра масс робота и точке нулевого момента для каждого цикла управления. Данная информация необходима для вычисления траектории перемещения центра масс (ЦМ) робота. Полагается, что в ЦМ концентрируется масса робота, что позволяет моделировать его динамику как динамику инвертированного маятника.

Для измерения положения ЦМ и ТНМ в рамках настоящего алгоритма вводится опорная система отсчета, начало координат которой находится на поверхности перемещения робота, ее ось Z расположена вертикально. В момент времени, когда один из педипуляторов робота оторван от поверхности перемещения (рисунок 33а), опорная система привязана к той стопе педипулятора, который служит опорой робота, с осью X , направленной вдоль стопы. Когда робот устойчиво стоит на педипуляторах, начало опорной системы отсчета соответствует точке, находящейся между стопами, при этом ось X имеет промежуточную ориентацию между ориентациями стоп (рисунок 33б). Положение ЦМ и ТНМ определяется и управляется относительно этой системы координат. Основным источником соответствующей информации для выполнения вычислений являются данные с ячеек комбинированных матриц датчиков давления и приближения, установленных на каждой стопе педипулятора.

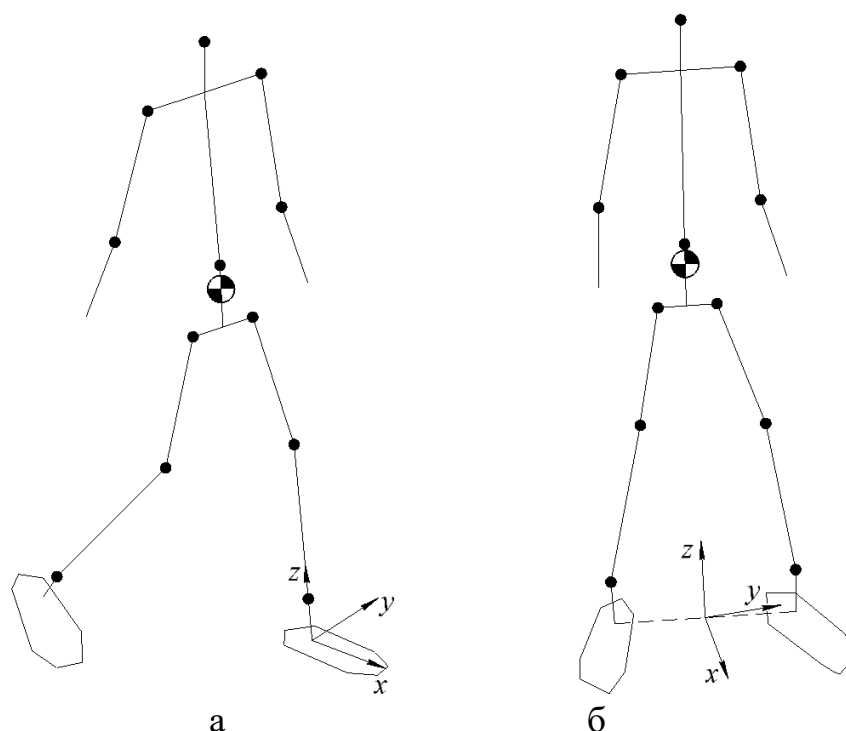


Рисунок 33 – Привязка систем отсчета для измерения положения ЦМ и ТНМ:
а – в процессе выполнения шага; б – исходное состояние

Следующим шагом разработанного алгоритма является вычисление точки траектории перемещения робота для отслеживания опорной системы координат при ходьбе робота. Затем в соответствии с известными аналитическими выражениями решается обратная задача кинематики с целью определения необходимых углов звеньев, при которых стопа педипулятора достигнет желаемой точки траектории. По полученным значениям углов рассчитываются управляющие воздействия для каждого сочленения педипулятора, звенья которого приводятся в движение исполнительными механизмами в виде сервоприводов. Управляющее воздействие вырабатывается регулятором на основании датчиков обратной связи (гироскопы по положению и скорости, датчик ускорения), сигналы с которых используются также при решении прямой задачи кинематики. В результате производимых вычислений и преобразований звенья педипуляторов приводятся в движение.

После отрыва педипулятора от опорной поверхности происходит его дальнейшее перемещение на заданный шаг до тех пор, пока не появится сигнал с комбинированных матричных датчиков о приближении к опорной поверхности. Если отклонение сигналов о приближении больше φ , происходит корректирование ориентации и положения стопы педипулятора, в противном случае управление перемещением звеньев будет длиться до момента касания стопы поверхности. При этом происходит определение ТНМ по показаниям давления с ячеек датчиков, после чего осуществляется проверка на соответствие ТНМ заданному положению с точностью θ . Если достигнутое положение ТНМ не соответствует заданному, происходит корректировка

углов звеньев педипуляторов. В случае соответствия заданного и достигнутого положения ТНМ шаг считается выполненным, а алгоритм завершенным.

Звенья каждого педипулятора перемещаются в соответствии с кинематической схемой, показанной на рисунке 34 (указаны выбранные положительные направления).

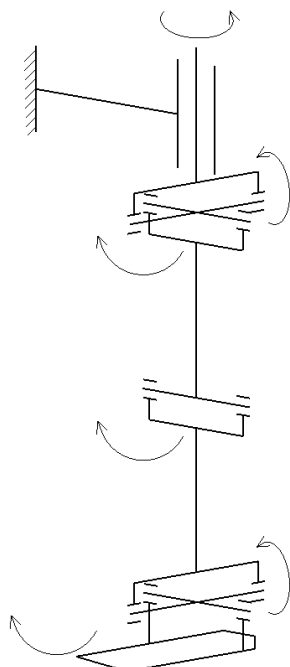


Рисунок 34 – Кинематическая схема педипулятора робота «АНТАРЕС»

В каждую стопу педипулятора антропоморфного шагающего робота «АНТАРЕС» [9, 10] были установлены разработанные комбинированные датчики давления и приближения, как показано на рисунке 35.

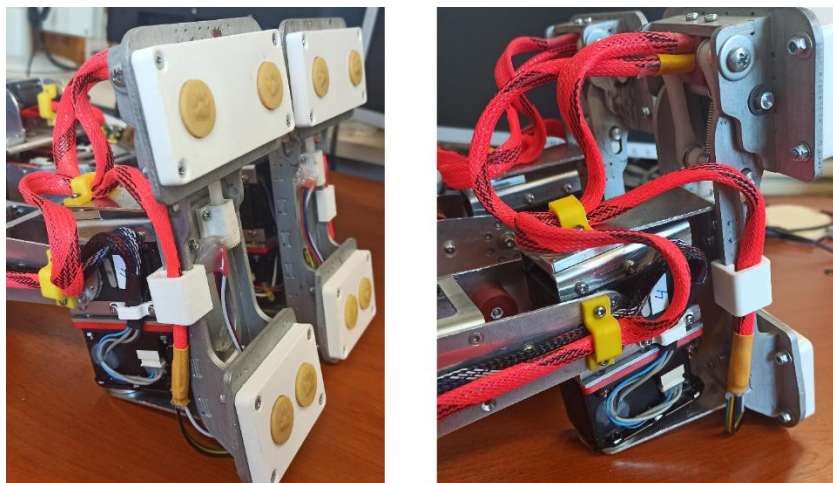


Рисунок 35 – Датчики давления и приближения в стопах робота «АНТАРЕС»

Стопа робота может сгибаться на угол от 0° до $39,5^\circ$. Стопа имеет четыре опорных точки, образующие трапецию, в каждой из которых установлен комбинированный датчик. Подобно человеческой стопе, передний отдел

стопа робота имеет большую ширину относительно задней, что положительно влияет на устойчивость робота. Считывание аналоговых сигналов с датчиков осуществляется посредством 16-битного АЦП с I2C интерфейсом (ADS1115). На рисунке 36 показан реализованный модуль матрицы сенсоров для установки в стопу робота и визуализация данных, получаемых с него.

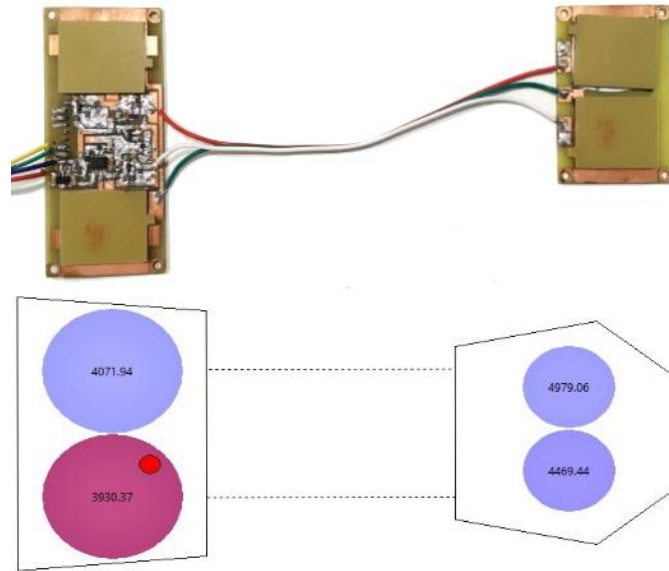


Рисунок 36 – Внешний вид и визуализация данных изготовленного сенсорного устройства стопы

Визуализация была проведена с помощью разработанного программного обеспечения, которое визуализирует полученные данные о величине приложенного давления к датчикам и расчетные точки центра приложенного к стопе давления. Для расчетов использовался геометрический метод в соответствии с формулами (31) и (32):

$$X_c = \frac{\sum_{i=1}^4 X_i P_i}{\sum_{i=1}^4 P_i}, \quad (31)$$

где X_i – абсцисса точки приложения давления на каждую опору стопы; P_i – сила давления, прикладываемого к каждой опоре стопы,

$$Y_c = \frac{\sum_{i=1}^4 Y_i P_i}{\sum_{i=1}^4 P_i}. \quad (32)$$

Были проведены серии экспериментов с определением центра давления стопы робота. Силовое воздействие на датчик осуществлялось через элементы конструкции, поглощающие ударные механические нагрузки. Была получена зависимость величины выходного сигнала датчика от приложенной к датчику силы, которая представлена на рисунке 37.

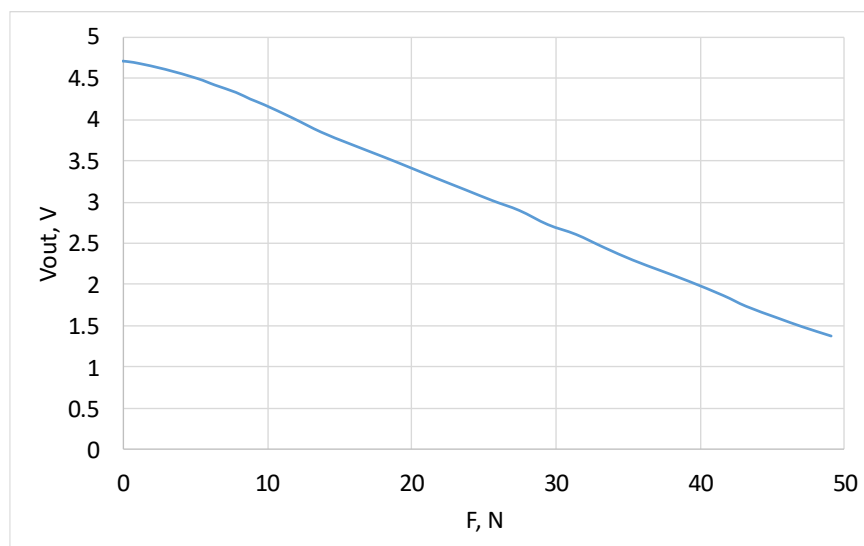


Рисунок 37 – Зависимость выходного сигнала датчика от приложенного к датчику давления

Полученная зависимость имеет линейный характер с незначительным отклонением от линейности в диапазоне нагрузок от 0 до 8 Н. Данное отклонение обусловлено наличием паразитных емкостей в экспериментальном образце между проводами, соединяющими интерфейсную схему с матрицами датчиков, которые соизмеримы по величине с емкостью комбинированных датчиков при малой величине приложенной силы давления. Диапазон нагрузок от 9 до 50 Н позволяет переводить выходной сигнал датчика в приложенное к датчику давление с помощью линейного уравнения вида:

$$y = kx + b. \quad (33)$$

Одним из основных параметров, характеризующих сенсорное устройство, является его чувствительность. Согласно характеристике, представленной на рисунке 37, расчетная чувствительность датчика составляет 0,072 В/Н.

Выполненное в рамках реализации проекта внедрение комбинированных датчиков давления и приближения в стопы антропоморфного робота позволило значительно улучшить качество походки и поддержания равновесия. Робот способен поддерживать равновесие в случае, когда шаг осуществляется на предварительно неизвестную наклонную поверхность. Проведены эксперименты, в которых робот совершал 20 отдельных шагов по двум типам наклонных поверхностей с углом наклона 5 и 10 градусов. Первый тип поверхности представлял собой искусственный газон с высотой ворса 10мм, вторая поверхность – ровный деревянный помост. На основе проведенных экспериментов, было выявлено, что при установке датчиков в основных точках опоры стопы количество ситуаций, связанных с потерей баланса и последующими падениями робота «Антареса», снизилось на 31,2%. Таких показателей удалось достигнуть за счет применения емкостного первичного преобразователя, который позволяет получать высокую чувствительность и низкое значение циклического дрейфа, что

необходимо для точного расчёта ТНМ при каждом шаге. Проведенное циклическое нагружение матриц с деформируемой проставкой из силикона в течении 10000 циклов показало средний дрейф значений АЦП 1,39 %, что позволяет эксплуатировать датчик для силомоментного очувствления АРС.

Установленная в схват манипулятора матрица датчиков позволяет определить точки касания с объектом и измерить контактные силы в каждой из них. Проведены эксперименты по захвату объектов, имеющих форму цилиндра, прямоугольного параллелепипеда и сферы. Объекты были выполнены из различных материалов: условно не деформируемые с твердостью более 50 единиц по Шору и легко деформируемые с твердостью материала менее 10 единиц по Шору. Обратная связь по силе давления позволяет осуществлять захват легко деформируемых объектов без нарушения их формы при предварительном задании допустимой силы схвата. Информация о силе давления схвата позволила минимизировать количество случаев выскальзывания объекта при манипулировании до 0,89 %. Данные о приближении к объекту пальцев схвата манипулятора позволяют успешно производить захват объекта на основе целевой точки в 97 % случаев.

Единовременное тактильное и силомоментное очувствление робототехнической системы позволяет улучшить функционирование роботов в окружающей среде, что особенно актуально при взаимодействии в киберфизических средах с различными объектами и человеком.

3. Подход к распределению задач между сервисными роботами и средствами киберфизического интеллектуального пространства при оценивании пространственных характеристик объектов, с которыми производится манипуляции и взаимодействие в процессе обслуживания пользователей

В рамках проекта для решения задачи оценки пространственных характеристик объектов и получения их трехмерных представлений было предложено комплексное решение, основанное на распределении задач между сервисными роботами и компонентами киберфизического интеллектуального пространства (КФИ). Соответствующее решение включает в себя два ключевых компонента: подход к определению пространственного положения и размеров объектов сцены по изображению с использованием методов машинного обучения и алгоритм получения трехмерных представлений и оценки пространственных характеристик объектов сцены с использованием сервисных робототехнических средств (РС).

В соответствии с результатами проведенного анализа связанных методов и подходов в рамках настоящего проекта был предложен авторский метод решения задачи оценки пространственных положений и характеристических свойств объектов сцены по изображениям. В рамках разработанного подхода источником изображений служит камера RealSense D435 [11], закрепленная вертикально на потолке некоторого помещения таким образом, чтобы оптическая ось объектива данной камеры располагалась в горизонтальной плоскости, и при этом стены помещения не попадали в кадр.

Использование предлагаемого подхода предполагает реализацию ряда предварительных шагов, выполняемых однократно в процессе развертывания соответствующего решения в исследуемом помещении. В рамках данного предварительного этапа осуществляется устранение эффекта дисторсии, устранение перспективных искажений, а также производится оценка исследуемой сцены и отдельных параметров видеокамеры. С данной целью, в первую очередь, осуществляется первоначальная калибровка камеры на основе метода [12]: калибровка устраняет эффект дисторсии (радиальное и тангенциальное искажения) и перспективные искажения, что приводит к повышению точности преобразования координат изображения в координаты реального мира. По результатам применения соответствующих методов определяется основная матрица камеры, содержащая фокусные расстояния и оптические центры, коэффициент искажения, вектора поворота и смещения. Все эти данные сохраняются для последующего использования в процессе функционирования соответствующего решения.

Следующим шагом в рамках предварительного этапа является оценка релевантных углов обзора видеокамеры по горизонтальной (*Alpha*) и вертикальной (*Beta*) оси изображения соответственно. Поскольку применение методов устранения эффекта дисторсии приводит к потере граничных частей изображения, использование угловых полей объектива видеокамеры в

качестве углов обзора становится невозможным. Таким образом, определение данных углов осуществляется экспериментально, исходя из оценки размера видимого сегмента плоской поверхности, фиксируемого камерой, при известном расстоянии между объективом камеры и данной поверхностью, ориентированной перпендикулярно оптической оси видеокамеры.

Последним шагом в рамках предварительного этапа является оценка исследуемой сцены и отдельных параметров видеокамеры. Оценка исследуемой сцены осуществляется экспериментально: при известных координатах центра сцены (x_c, y_c) , расположенного строго под камерой, измеряется расстояние от пола до объектива камеры по вертикали H . В таком случае координаты центра сцены будут иметь следующий вид:

$$Center = (x_c, y_c, H).$$

Центр изображения совпадает с центром сцены с точностью до координаты z . Таким образом, для центра изображения $c_{im} = (x_{c_{im}}, y_{c_{im}})$ справедливо следующее выражение:

$$x_{c_{im}} = x_c;$$

$$y_{c_{im}} = y_c.$$

Таким образом, на центральном пикселе получаемого с камеры изображения отображается центр исследуемой сцены. Оценка отдельных параметров исследуемой сцены также подразумевает определение того, какому участку в пространстве соответствовал бы каждый пиксель изображения в случае, если на изображении присутствовал бы только пол. Рассмотрим данный вопрос более подробно. Пусть есть пространство с тремя объектами на нем, как показано ниже на рисунке 38.

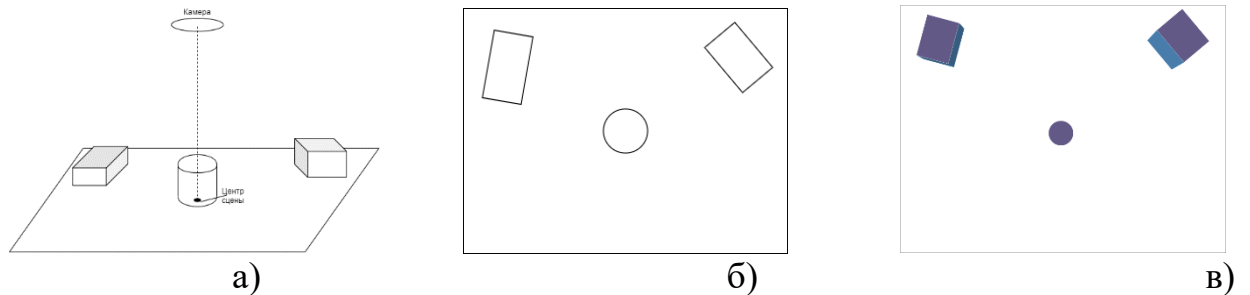


Рисунок 38 – а) Иллюстрация сцены с тремя объектами; б) Иллюстрация проекции данной сцены на плоскость XY ; в) Иллюстрация изображения, снятого камерой, находящейся над сценой и установленной перпендикулярно поверхности пола

Как можно заметить на иллюстрациях выше, существуют значительные отличия между проекцией сцены и снимком камеры, обусловленные тем, что снимок камеры имеет статически зафиксированную точку обзора и при фиксации на нем объектов, смещенных по осям x и y , они наблюдаются под некоторым углом. Соответственно, на изображении, полученном камерой, которая закреплена перпендикулярно к поверхности пола, просматриваются не только верхние, но и боковые грани объектов сцены.

С учетом представленных выше особенностей изображений трехмерных сцен был предложен следующий подход к оценке пространственных положений и характеристических свойств объектов сцены. Разработанный подход включает в себя следующие основные этапы:

1. Детектирование объектов на исследуемом изображении:
 - a. Передача полученного изображения с камеры RealSense D435 [11] в нейросетевую модель Mask R-CNN [13];
 - b. Определение количества объектов на изображении и их принадлежности к определенному классу объектов;
 - c. Выделение боксов на изображении, соответствующих задетектированным объектам;
 - d. Определение для каждого бокса пяти опорных точек: вершин и центра бокса;
2. Определение пространственных областей сцены, соответствующих пикселям исследуемого изображения, полученного с камеры RealSense D435;
3. Определение высоты, размеров и пространственных координат объектов, задетектированных на изображении.

Как можно заметить, предложенный подход содержит три ключевых этапа, каждый из которых направлен на решение отдельной группы подзадач. Рассмотрим каждый из этапов предложенного подхода более детально.

3.1 Детектирование объектов на исследуемом изображении

На первом этапе изображение, полученное с камеры RealSense D435, передается в нейросетевую модель Mask R-CNN, которая производит поиск координат объектов на изображении, определяет количество объектов и их принадлежность к определенному классу предметов. Inception блоки, которые используются в данной нейронной сети, представляют собой аналоги блоков в нейронной сети Inception-V1, где карта признаков, полученная на предыдущем слое, параллельно обрабатывается четырьмя различными ветвями. Первая ветвь блока Inception представляет собой свертку с фильтром размером $(1 \times 1 \times 1)$, вторая и третья ветви содержат по два последовательных сверточных слоя с фильтрами размером $(1 \times 1 \times 1)$ и $(3 \times 3 \times 3)$ соответственно, а третья ветвь содержит два последовательных слоя – Max-Pool с полем размером $(3 \times 3 \times 3)$ и сверточный слой размером $(1 \times 1 \times 1)$. Далее формируется результирующая карта признаков с помощью конкатенации карт признаков, полученных с использованием всех четырех ветвей модуля Inception.

Первые слои модели Mask R-CNN представляют собой чередующиеся трехмерные сверточные и пулинговые слои, за которыми следует последовательность чередующихся блоков Inception и пулинговых слоев. Выход данной нейронной сети представляет собой трехмерный сверточный слой размера $(1 \times 1 \times 1)$. При этом выходная карта признаков этого слоя «вытягивается» в вектор, размер которого соответствует числу рассматриваемых классов.

Нейронная сеть данной архитектуры способна обрабатывать фрагменты видео различной длительности, каждый кадр является цветным трехканальным изображением цветового пространства RGB. Обработывая изображение, нейронная сеть возвращает количество найденных объектов и набор массивов (боксы, классы, маски, степень соответствия). Боксы и маски представляют собой искомый набор точек на изображении, принадлежащих объекту, а классы и коэффициент соответствия позволяют определить, насколько точно каждый объект соответствует конкретному классу. В рамках разработанного подхода для последующего определения ключевых точек объектов на изображениях было принято решение использовать боксы, поскольку с их помощью можно получить всю требуемую информацию об исследуемых объектах, кроме того, использование боксов положительным образом сказывается на вычислительной сложности предложенного подхода. Боксы представляют собой массивы из четырех крайних точек, соответствующих прямоугольным областям, в которых находятся объекты. По результатам детектирования боксов на изображении для каждого из боксов формируется набор опорных точек, состоящий из координат вершин и геометрического центра данного бокса. Таким образом, результатом применения нейросетевой модели Mask R-CNN [14] к исследуемому изображению является массив O , состоящий из множества наборов опорных точек боксов объектов $Points_j$ и названий классов Cls_j , соответствующих задетектированным объектам Ob_j .

3.2 Определение пространственных областей сцены, соответствующих пикселям исследуемого изображения, полученного с камеры RealSense D435

Рассмотрим разработанный алгоритм определения пространственных областей сцены, соответствующих пикселям исследуемого изображения. Пусть есть изображение пола некоторой сцены, на которой отсутствуют объекты (рис. 39.а). Центральный пиксель изображения P_c , расположенный непосредственно под камерой, в пространстве соответствует некоторому квадрату в центре сцены Fig_c с площадью S_c . Однако, если взять не центральный пиксель, а некоторый произвольный пиксель изображения P_i , то ситуация будет несколько иной:

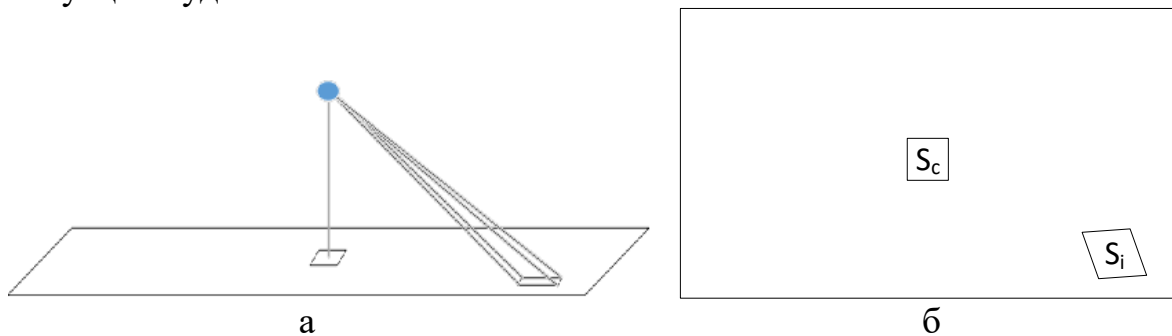


Рисунок 39 – а) Иллюстрация сцены без объектов; б) Иллюстрация проекции данной сцены на плоскость XY

В данном случае (рис. 39.б), пикселю P_i в пространстве будет соответствовать некоторый четырехугольник Fig_i с площадью S_i , причем $S_c \neq S_i$. Данное явление связано с тем, что участки сцены, смещенные от центра сцены, камера фиксирует под некоторым углом, и тем самым на каждый пиксель изображения приходится больший участок пространства. Рассмотрим более подробно решение задачи соотнесения пикселей изображения и соответствующих им пространственных участков в оговоренных выше условиях: на изображении сцены присутствует только пол, камера зафиксирована перпендикулярно плоскости пола.

Пусть размер исследуемого изображения в пикселях по горизонтальным и вертикальным осям равен Img_{pix_x} и Img_{pix_y} соответственно. В таком случае индексы центрального пикселя P_c на изображении по данным осям могут быть определены следующим образом:

$$P_c(x; y) = \frac{Img_{pix_x}}{2}; \frac{Img_{pix_y}}{2}.$$

Как было сказано ранее, центральный пиксель изображения P_c в пространстве соответствует некоторому квадрату в центре сцены Fig_c (рис. 40).

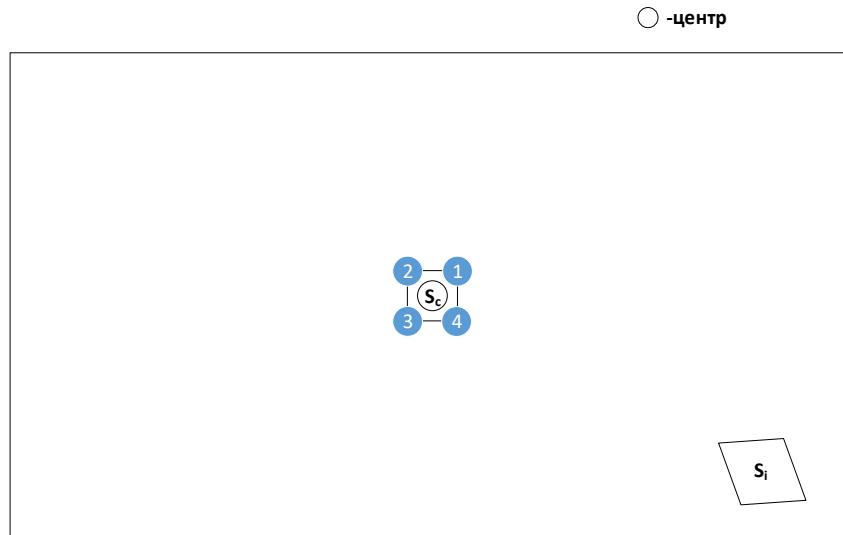


Рисунок 40 – Иллюстрация области пространства, соответствующей центральному пикселю сцены, в проекции на плоскость XY

Таким образом, каждая из сторон данного квадрата, отклонена от нормали к плоскости XY на некоторый плоский угол в плоскостях XZ или YZ. Величина данного углового отклонения может быть определена следующим образом:

$$ang_dev_x_{P_c} = \left[-\frac{Alpha}{Img_{pix_x}*2}; \frac{Alpha}{Img_{pix_x}*2} \right] \quad (34)$$

$$ang_dev_y_{P_c} = \left[-\frac{Betta}{Img_{pix_y}*2}; \frac{Betta}{Img_{pix_y}*2} \right] \quad (35)$$

где $ang_dev_x_{P_c}$ – это угловое отклонение по оси x , $ang_dev_y_{P_c}$ – угловое отклонение по оси y .

Соответственно, каждая из вершин данного квадрата отклонена от нормали к плоскости XY на некоторые плоские углы в плоскостях XZ и YZ. Следовательно, координаты вершин данной области пространства, соответствующей центральному пикселю исследуемого изображения, могут быть определены следующим образом:

$$\begin{aligned} 1_c: & \left(tg \left(\frac{Alpha}{img_{pix_x} * 2} \right) * H; tg \left(\frac{Betta}{img_{pix_y} * 2} \right) * H \right); \\ 2_c: & \left(tg \left(-\frac{Alpha}{img_{pix_x} * 2} \right) * H; tg \left(\frac{Betta}{img_{pix_y} * 2} \right) * H \right); \\ 3_c: & \left(tg \left(-\frac{Alpha}{img_{pix_x} * 2} \right) * H; tg \left(-\frac{Betta}{img_{pix_y} * 2} \right) * H \right); \\ 4_c: & \left(tg \left(\frac{Alpha}{img_{pix_x} * 2} \right) * H; tg \left(-\frac{Betta}{img_{pix_y} * 2} \right) * H \right). \end{aligned}$$

В общем случае, для произвольного пикселя изображения P_i с индексами по горизонтальной и вертикальной осям i_{pix_x} и i_{pix_y} соответственно, угловое отклонение ang_dev по данным осям задается как:

$$\begin{aligned} ang_dev_{x_{P_i}} &= \left[-\frac{Alpha}{img_{pix_x} * 2} + \left(i_{pix_x} - \frac{img_{pix_x}}{2} \right) * \frac{Alpha}{img_{pix_x}}; \frac{Alpha}{img_{pix_x} * 2} + \left(i_{pix_x} - \frac{img_{pix_x}}{2} \right) * \frac{Alpha}{img_{pix_x}} \right]; \\ ang_dev_{y_{P_i}} &= \left[-\frac{Betta}{img_{pix_y} * 2} + \left(i_{pix_y} - \frac{img_{pix_y}}{2} \right) * \frac{Betta}{img_{pix_y}}; \frac{Betta}{img_{pix_y} * 2} + \left(i_{pix_y} - \frac{img_{pix_y}}{2} \right) * \frac{Betta}{img_{pix_y}} \right]. \end{aligned}$$

Положения вершин пространственного участка сцены, охватываемого данным пикселем P_i соответственно можно определить следующим образом:

$$\begin{aligned} 1_i: & \left(tg \left(\frac{Alpha}{img_{pix_x} * 2} + \left(i_{pix_x} - \frac{img_{pix_x}}{2} \right) * \frac{Alpha}{img_{pix_x}} \right) * H; tg \left(\frac{Betta}{img_{pix_y} * 2} + \left(i_{pix_y} - \frac{img_{pix_y}}{2} \right) * \frac{Betta}{img_{pix_y}} \right) * H \right) \\ 2_i: & \left(tg \left(-\frac{Alpha}{img_{pix_x} * 2} + \left(i_{pix_x} - \frac{img_{pix_x}}{2} \right) * \frac{Alpha}{img_{pix_x}} \right) * H; tg \left(\frac{Betta}{img_{pix_y} * 2} + \left(i_{pix_y} - \frac{img_{pix_y}}{2} \right) * \frac{Betta}{img_{pix_y}} \right) * H \right) \\ 3_i: & \left(tg \left(-\frac{Alpha}{img_{pix_x} * 2} + \left(i_{pix_x} - \frac{img_{pix_x}}{2} \right) * \frac{Alpha}{img_{pix_x}} \right) * H; tg \left(-\frac{Betta}{img_{pix_y} * 2} + \left(i_{pix_y} - \frac{img_{pix_y}}{2} \right) * \frac{Betta}{img_{pix_y}} \right) * H \right) \\ 4_i: & \left(tg \left(\frac{Alpha}{img_{pix_x} * 2} + \left(i_{pix_x} - \frac{img_{pix_x}}{2} \right) * \frac{Alpha}{img_{pix_x}} \right) * H; tg \left(-\frac{Betta}{img_{pix_y} * 2} + \left(i_{pix_y} - \frac{img_{pix_y}}{2} \right) * \frac{Betta}{img_{pix_y}} \right) * H \right) \end{aligned}$$

В соответствии с представленными выше выражениями для каждого целевого пикселя P_i на изображении могут быть найдены вершины участка пространства, который соответствует данному пикселю. Для каждого такого участка Fig_i можно достаточно легко оценить его центр C_i как среднее арифметическое координат его вершин:

$$C_i = \left(\frac{1}{4} \sum_{k=1}^4 k_{i_x}; \frac{1}{4} \sum_{k=1}^4 k_{i_y} \right)$$

Таким образом, по итогам данного этапа формируется массив центров пространственных участков сцены для пикселей, соответствующих опорным точкам боксов объектов из наборов $Points_j$, входящих в состав массива O .

3.3 Определение высоты, размеров и пространственных координат объектов, задетектированных на изображении

В рамках заключительного этапа разработанного подхода осуществляется определение высоты, размеров и пространственных координат задетектированных на изображении объектов Ob_j . На вход данному этапу поступает информация о полученном на втором этапе массиве центров пространственных участков сцены C_i для пикселей P_i , соответствующих опорным точкам боксов объектов из наборов $Points_j$. Для каждого такого центра проводится расчет расстояния от объектива камеры до данного участка пространства в соответствии со следующим выражением:

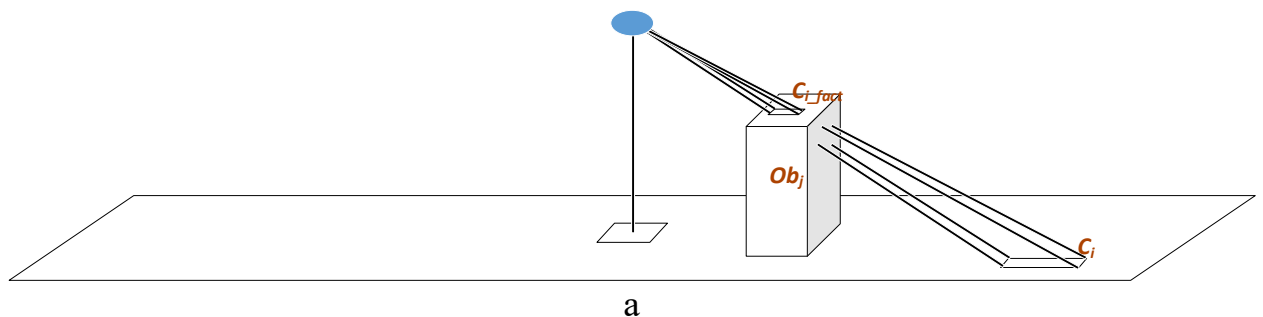
$$D(C_i)_{floor} = \left(\sqrt{C_{ix}^2 + C_{iy}^2 + H^2} \right).$$

Таким образом, мы получаем ожидаемые расстояния $D(C_i)_{floor}$, которые должны были бы соответствовать каждому рассматриваемому пикселю изображения P_i в случае, если бы сцена представляла собой ровный пол, на котором бы отсутствовали какие-либо объекты.

Далее для пикселей P_i , соответствующих опорным точкам боксов объектов Ob_j , с использованием камеры RealSense D435 определяются расстояния $D(C_i)_{fact}$ до центров фактических пространственных участков сцены, ассоциированных с пикселями P_i (рис. 41.а). Теперь мы можем вычесть из набора ожидаемых расстояний $D(C)_{floor}$, фактические расстояния $D(C)_{fact}$:

$$D(C)_{delta} = D(C)_{floor} - D(C)_{fact}.$$

Таким образом, мы получаем разностный набор $D(C)_{delta}$. Каждый элемент $D(C_i)_{delta}$ данного набора представляет собой некоторое расстояние B , которое равно разнице между тем расстоянием, которое было бы между камерой и пространственным участком пола в соответствующем пикселе P_i , если бы с данным пикселем был ассоциирован участок пола (расстояние $A+B$), и фактическим расстоянием между пространственным участком сцены, отображаемом на данном пикселе, и камерой – расстояние A (рис. 41.б).



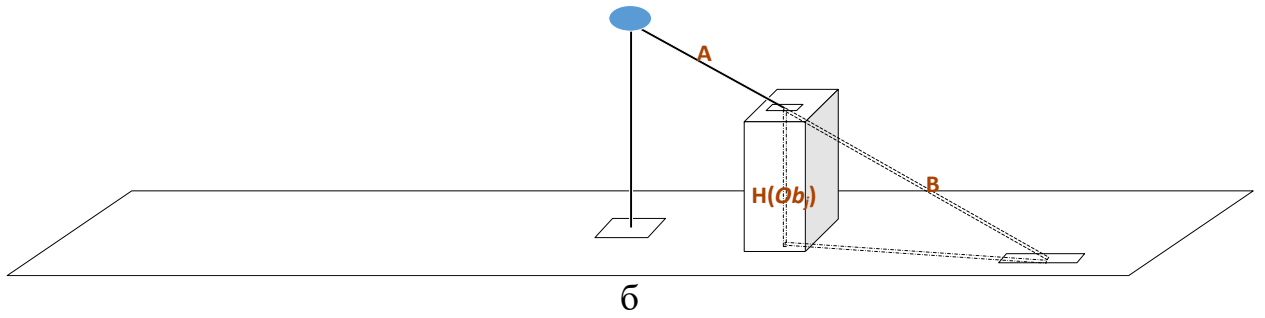


Рисунок 41 – а) Иллюстрация типовой сцены с некоторым произвольным объектом; б) Иллюстрация типовой сцены с указанием исследуемых расстояний

Важно отметить, что в отсутствии погрешностей все элементы набора $D(C)_{delta}$ должны быть больше либо равны 0. Далее, с использованием определенных выше величин, могут быть найдены высоты объектов Ob_j , задетектированных на исследуемой сцене. Вычисление высоты каждого объекта Ob_j осуществляется в два этапа. На первом этапе для каждого целевого пикселя изображения P_{ji} , соответствующего одной из опорных точек $Points_{ji}$ из набора $Points_j$ (набора опорных точек бокса объекта Ob_j) осуществляется определение вертикального смещения пространственного участка C_{ji} относительно пола в соответствии со следующим выражением:

$$H(C_{ji}) = \frac{H}{A + B} * B = \frac{H}{D(C_{ji})_{floor}} * D(C_{ji})_{delta}$$

Таким образом, формируются оценки высоты объекта Ob_j по опорным точкам бокса данного объекта. На втором этапе формируется результирующая оценка высоты объекта $H(Ob_j)$ в соответствии со следующим выражением:

$$H(Ob_j) = \sum_{i=1}^5 H(C_{ji}) * w_i,$$

где i – индекс опорной точки бокса объекта Ob_j , w – весовые коэффициенты опорных точек боксов, равные 0.7 для опорных точек, ассоциированных с геометрическими центрами боксов, и 0.075 – для других опорных точек. Таким образом, по результатам данных расчетов для каждого задетектированного на изображении объекта Ob_j формируется окончательный вывод о его высоте.

После того, как в отношении каждого объекта Ob_j на сцене была сформирована результирующая оценка его высоты $H(Ob_j)$, определение двух других величин, характеризующих линейный размер каждого из объектов Ob_j , осуществляется следующим образом:

1. В первую очередь, определяются пространственные координаты опорных точек бокса объекта Ob_j :

$$Points_{ji} = (C_{jix}, C_{jiy}, H(Ob_j)), i = [1 \dots m],$$

где C_j – массив центров пространственных участков сцены для пикселей, соответствующих опорным точкам бокса объекта из набора $Points_j$, $H(Ob_j)$ – результирующая оценка высоты объекта, $m = 5$ – число опорных точек бокса данного объекта.

2. Оценка длины объекта Ob_j , таким образом, может быть определена как расстояние между двумя смежными опорными точками бокса данного объекта. В работе оценка длины объекта осуществлялась в соответствии со следующим выражением:

$$L(Ob_j) = |Points_{j2} - Points_{j1}|$$

3. Аналогичным образом может быть определена оценка ширины объекта Ob_j . В работе оценка ширины объекта осуществлялось в соответствии со следующим выражением:

$$W(Ob_j) = |Points_{j3} - Points_{j2}|$$

Для оценки местоположения объектов за начало системы отсчета было принято решение взять положение объектива фиксирующей видеокамеры, а в качестве оценок местоположения использовать оценки пространственного положения центров данных объектов. Для каждого из задетектированных на изображении объектов пространственное положение его центра было определено в соответствии со следующим выражением:

$$C(Ob_j) = (C_{j5x}, C_{j5y}, H - \frac{H(Ob_j)}{2}),$$

где C_{j5} – положение центра пространственного участка, соответствующего опорной точке бокса объекта, ассоциированной с геометрическим центром бокса данного объекта, H – расстояние от объектива фиксирующей видеокамеры до пола исследуемого помещения по вертикальной оси, $H(Ob_j)$ – полученная на предыдущем этапе оценка высоты объекта Ob_j .

Таким образом, по результатам выполнения представленных выше операций в отношении каждого задетектированного на изображении объекта Ob_j ($\forall Ob_j \in O$), могут быть получены оценочные значения трех ключевых характеристик размера данных объектов (высота, длина и ширина), а также получены оценки местоположений данных объектов в системе координат, связанной с объективом фиксирующей видеокамеры.

Обобщенная алгоритмическая модель разработанного подхода к оценке пространственных положений и характеристических свойств объектов сцены по изображениям представлена ниже на рисунке 42.

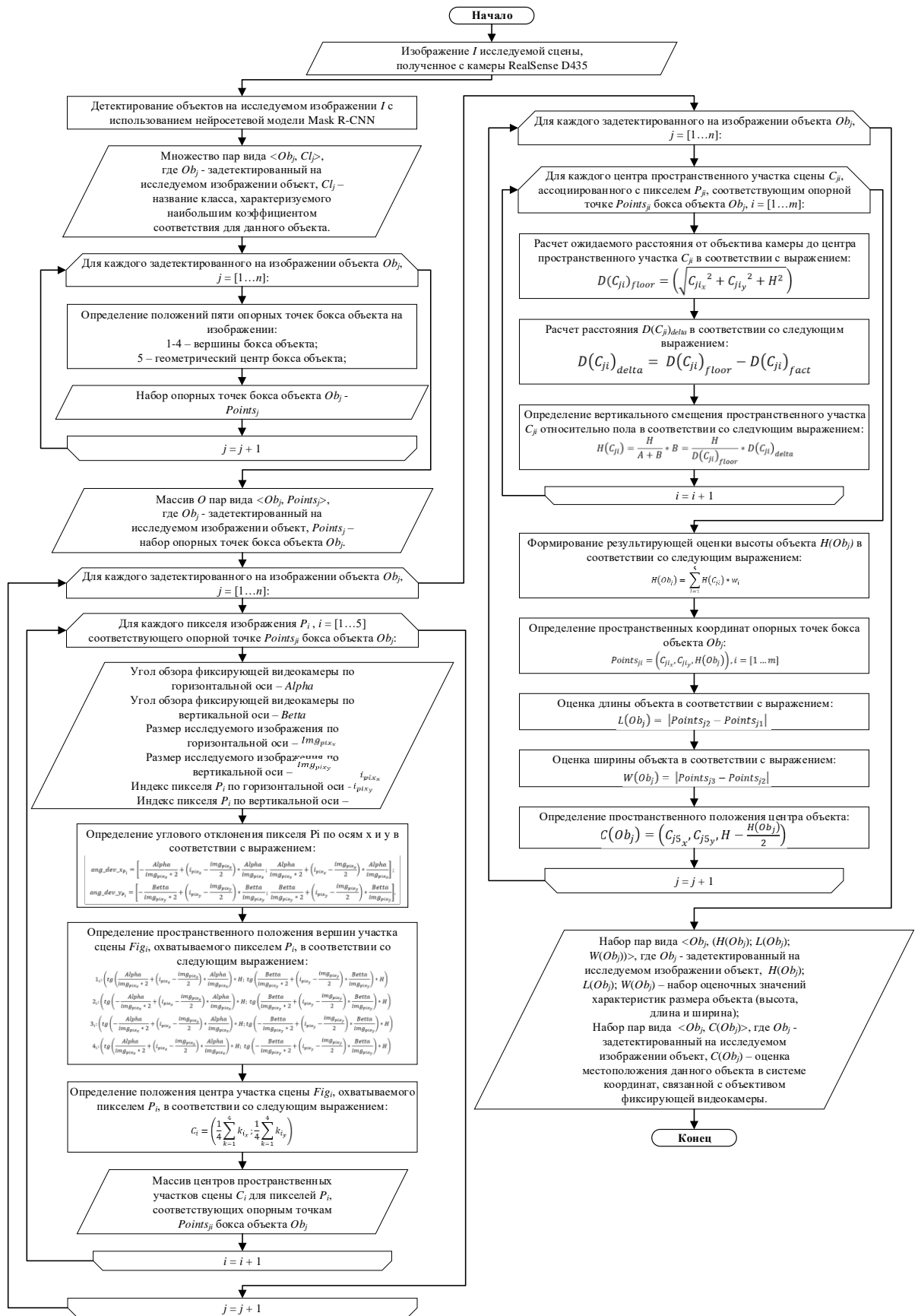


Рисунок 42 – Обобщенная алгоритмическая модель разработанного подхода к оценке пространственных положений и характеристических свойств объектов сцены

Ниже на рисунке 43 представлены поэтапные результаты работы разработанного подхода к определению пространственного положения и размеров объектов сцены на некотором изображении.

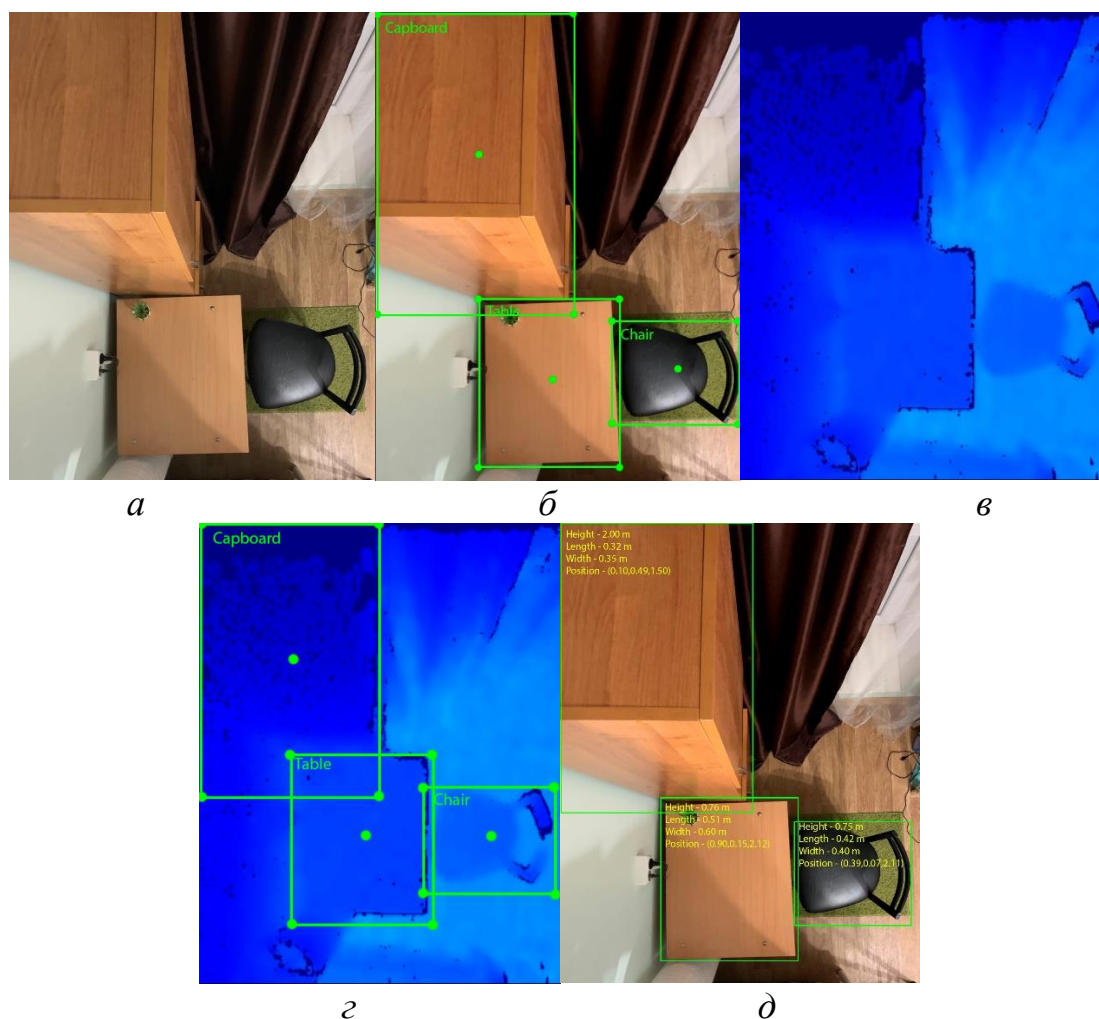


Рисунок 43 – Поэтапные результаты работы предложенного подхода на некотором исследуемом участке сцены: а) RGB-изображение сцены с 2-3 простыми объектами на сцене; б) результат работы Mask R-CNN с выделенными боксами и подписанными классами объектов; в) карта глубины исследуемой сцены, полученная при помощи камеры RealSense D435; г) карта глубины исследуемой сцены с выделенными боксами и подписанными классами объектов; д) результирующее RGB-изображение с подписанными параметрами объектов

На рисунке 43.а представлено RGB-изображение исследуемой сцены. Согласно разработанному подходу, в первую очередь, осуществляются детектирование и классификация объектов с использованием нейросетевой модели Mask R-CNN (рисунок 43.б). На данном кадре было задетектировано три объекта: стул, стол и шкаф. Далее, в соответствии с разработанным алгоритмом, для заданного участка сцены строятся карты глубины с использованием камеры RealSense D435 (рисунок 43.в). На рисунке 43.г представлена карта глубины исследуемой сцены с выделенными боксами и

подписанными классами объектов. Результаты определения пространственного положения и размеров задетектированных объектов представлены на рисунке 43.д.

Апробация и оценка качества разработанного подхода к определению пространственного положения и размеров объектов сцены по изображению проводилась на основании тестового набора данных, включающего в себя 12000 изображений различных сцен с офисными помещениями. Данный набор данных был сформирован с использованием камеры глубины RealSense D435. Каждому изображению I_i , входящему в состав тестового набора данных, соответствует некоторый набор значений параметров P_i . Описание данных параметров представлено ниже:

а. Тип целевого объекта T . Включает в себя 4 различных класса объектов: шкафы (1), стулья (2), столы (3), растения (4).

б. Размер целевого объекта R . В качестве параметра, характеризующего размер объектов, предлагается выбрать среднее значение длин проекций объекта на каждую из осей координат l (м). Таким образом, каждый целевой объект будет отнесен к одной из следующих категорий, исходя из значения l : 1 – $[0, 0.5]$, 2 – $[0.5, 1]$, 3 – $[1, 2]$.

с. Уровень освещенности сцены L . Тестовый набор данных включает в себя изображения, сформированные при различных уровнях освещенности сцены: 50%, 70% и 100%, где за 100% взят нормативный уровень освещенности для офисных помещений [15].

Таким образом, тестовый набор данных был разделен на три подгруппы изображений, отличающихся по уровню освещенности сцены: 50%, 70% и 100%. Каждая выделенная подгруппа изображений включает в себя 4000 изображений, содержащих объекты следующих типов и размеров: 600 столов размера 2; 150 растений размера 1; 50 растений размера 2; 8000 стульев размера 1; 4000 стульев размера 2 и 400 шкафов размера 3. В результате применения нейросетевой модели Mask R-CNN для подгруппы изображений с освещенностью сцены 100% было задетектировано и корректно классифицировано следующее количество объектов на сцене:

Таблица 5 – Доля задетектированных и корректно классифицированных объектов на сцене в условиях 100% освещенности

Тип целевого объекта T	Размер целевого объекта R	Доля задетектированных объектов от общего числа объектов	Доля корректно классифицированных объектов от числа задетектированных
Стол	2	89%	60%
Растение	1	88%	49%
Растение	2	88%	48%
Стул	1	90%	50%
Стул	2	89%	50%
Шкаф	3	90%	80%

На основе полученных данных для всех трех групп изображений, были определены результирующие доли задетектированных и корректно классифицированных объектов для каждой выделенной подгруппы с заданным уровнем освещенности сцены. Полученные данные представлены в таблице 6.

Таблица 6 – Доля задетектированных и корректно классифицированных объектов с использованием нейросетевой модели Mask R-CNN

Тип целевого объекта T	Размер целевого объекта R	Доля задетектированных и корректно классифицированных объектов		
		При освещенности сцены 100%	При освещенности сцены 70%	При освещенности сцены 50%
Стол	2	0,53	0,37	0,26
Растение	1	0,43	0,30	0,22
Растение	2	0,42	0,29	0,21
Стул	1	0,45	0,32	0,22
Стул	2	0,45	0,32	0,23
Шкаф	3	0,72	0,51	0,36
Средняя доля		0,50	0,35	0,25

Из представленных выше данных можно сделать вывод о том, что точность детектирования и классификации объектов с использованием нейросетевой модели Mask R-CNN существенным образом зависит от уровня освещенности сцены и снижается при понижении уровня освещенности. Важно также отметить, что класс объекта влияет на качество классификации, в частности рассчитанные показатели имеют более низкие значения для объектов следующих классов: растение и стул.

Для осуществления непосредственной оценки качества работы разработанного подхода к определению пространственного положения и размеров объектов сцены по изображению были сформированы различные количественные оценки точности работы предложенного решения в зависимости от параметров сцены и характеристик целевых объектов. Соответствующие показатели были определены следующим образом:

1. В качестве показателя точности определения размеров объектов сцены была взята относительная погрешность, определявшаяся в соответствии со следующим выражением:

$$\frac{|a_{real} - a|}{\left(\frac{a_{real} + a}{2}\right)},$$

где a_{real} – реальное значение размера объекта; a – величина, рассчитанная на основе данных, полученных с использованием разработанного подхода к определению пространственного положения и размеров объектов сцены по изображению. Величины a_{real} и a непосредственно определялись как

усредненные значения длин проекций объекта на каждую из осей координат. Таким образом, чем ниже значение относительной погрешности, тем выше успешность определения размеров объектов сцены с использованием разработанного подхода.

Ниже на рисунке 44 представлена диаграмма полученных значений относительной погрешности определения размеров для каждого типа объектов для подгруппы изображений с освещенностью сцены, равной 100%.

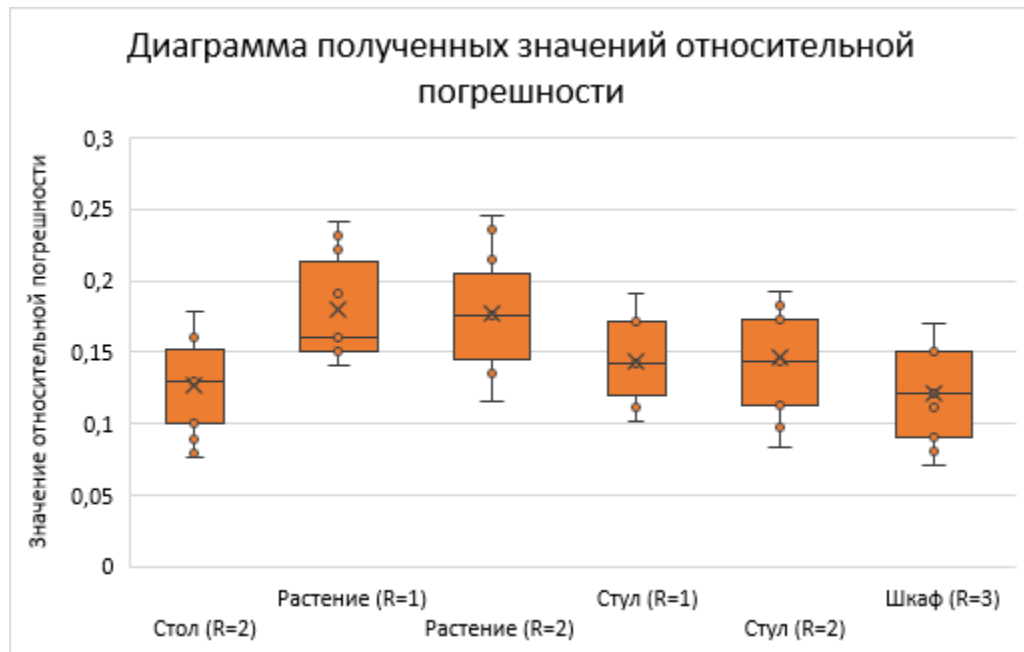


Рисунок 44 – Диаграмма полученных значений относительной погрешности определения размеров для каждого типа объектов

На рисунке выше представлена диаграмма ящик с усами (box plot), отражающая распределение полученных значений относительной погрешности определения размеров для каждого типа объектов. На данной диаграмме границами ящиков служат первый и третий квартили (25-й и 75-й процентили соответственно). Оценка средней величины ошибки отображена в виде крестика в середине ящиков, выбросы отображаются на графике в виде точек. Концы усов – края статистически значимой выборки (без выбросов), определяющиеся в соответствии со следующими выражениями:

$$X_1 = Q_1 - k(Q_3 - Q_1), X_2 = Q_3 + k(Q_3 - Q_1),$$

где X_1 – нижняя граница уса, X_2 – верхняя граница уса, Q_1 – первый квартиль, Q_3 – третий квартиль, k – коэффициент, наиболее часто употребляемое значение которого равно 1,5.

Из данных, представленных на рисунке 44, можно заключить, что разработанный подход демонстрирует наилучшие результаты по определению размеров для объектов классов шкаф (R=3) и стол (R=2). Значения средней величины ошибки для объектов данных классов составили соответственно 0.1207 и 0.1296. Несколько худшее качество определения размеров объектов разработанный подход показывает для таких классов как стул (R=1,2),

полученные значения средней величины ошибки для объектов данных классов составили соответственно 0.1416 и 0.1429. Наихудшие результаты по определению размеров в рамках данного эксперимента разработанный подход продемонстрировал для объектов классов растение (R=1,2). Предполагается, что подобные результаты связаны с геометрическими особенностями данных объектов. Важно также отметить, что для объектов одинакового класса, но разного размера, в среднем не наблюдается существенное изменение значений относительной погрешности. Таким образом, можно сделать вывод, что до определенного предела фактические размеры объектов не влияют на качество определения размеров объектов в контексте разработанного подхода. Однако, так же следует отметить, что для объектов крайне малого размера, данный вывод не будет справедлив, поскольку в таком случае, существенно большую роль начнут играть ошибки построения карт глубины, а также иные погрешности разработанного подхода.

Ниже на рисунке 45 представлены усредненные значения относительной погрешности определения размеров объектов по изображениям при разных уровнях освещенности сцены.

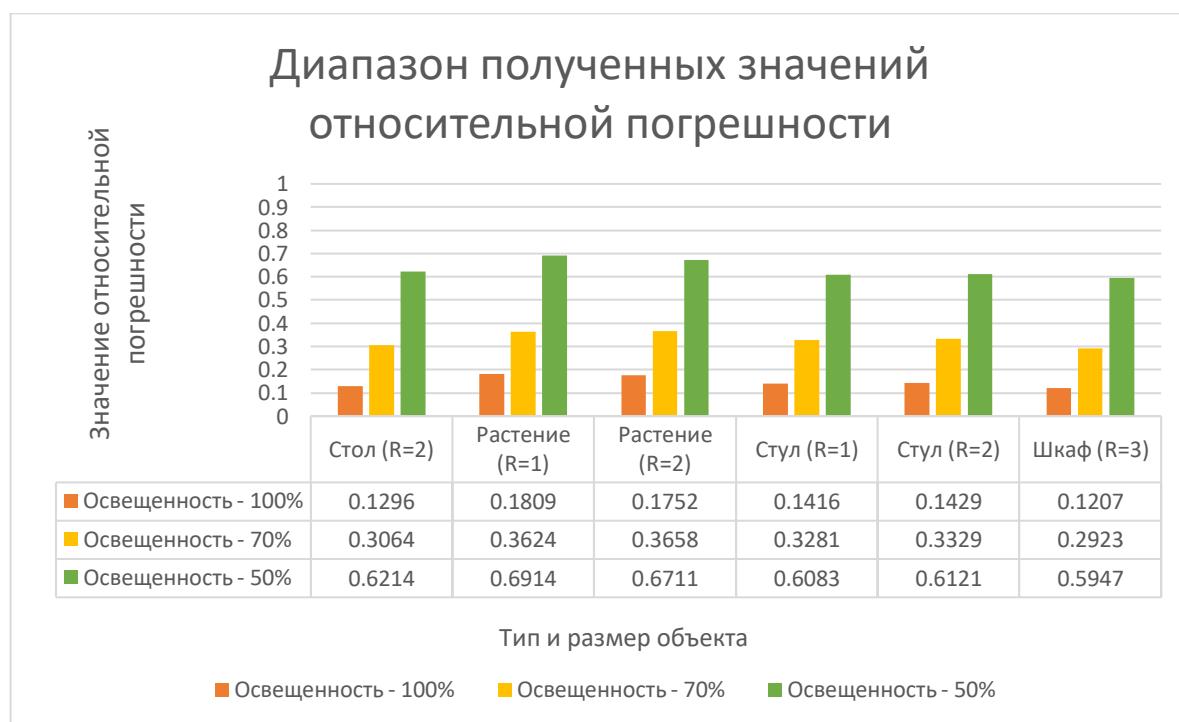


Рисунок 45 – Диапазон полученных значений относительной погрешности определения размеров для всех уровней освещенности сцены

Как можно заметить по представленным выше данным, для всех типов объектов, независимо от их размера, наблюдается значительный рост относительной погрешности определения размеров данных объектов при снижении уровня освещенности сцены. При этом наблюдается возрастающий темп роста данной погрешности при переходе от 100% к 70% и от 70% к 50% уровням освещенности соответственно. В частности, рост относительной погрешности при переходе от 100% к 70% освещенности составляет в среднем

7,5%, а при переходе от 70% к 50% составляет в среднем 9,5% при 1% снижении уровня освещенности. Данные зависимости носят схожий характер для всех рассматриваемых классов объектов независимо от их размера. Усредненные по классам значения относительной погрешности для сцен с различными уровнями освещенности равны 0.1449, 0.3313 и 0.6332 для 100%, 70% и 50% уровней освещенности соответственно.

Полученные результаты могут быть объяснены тем, что при значительном уровне освещенности снижается точность детектирования объектов, кроме того, возникают дополнительные погрешности при построении карт глубины, что существенным образом сказывается на качестве определения размеров объектов. Данная ситуация наблюдается при применении разработанного подхода к определению пространственного положения и размеров объектов сцены на изображениях с уровнем освещенности в 50%. Важно также отметить, что снижение уровня освещенности крайне негативно сказывается на качестве детектирования и классификации объектов в результате чего, большая часть представленных на сцене объектов остается незамеченной или же некорректно классифицированной.

Таким образом, на основе полученных результатов можно заключить, что разработанный подход демонстрирует приемлемое качество определения размеров объектов для изображений сцен с уровнями освещенности в 100% и 70%.

2. В качестве показателя точности определения пространственного положения объектов была взята такая метрика как величина относительного отклонения, которая определяется согласно выражению:

$$Err = \frac{|r_{real} - r|}{l},$$

где r_{real} – реальное положение объекта; r – положение объекта, определенное с помощью разработанного подхода; l – усредненное значение длин проекций объекта на каждую из осей координат. Таким образом, чем ниже значение относительного отклонения, тем выше точность определения пространственного местоположения объектов сцены с использованием разработанного подхода.

Ниже на рисунке 46 представлен диапазон полученных значений относительного отклонения при определении пространственного положения объектов для подгруппы изображений с освещенностью сцены 100%.

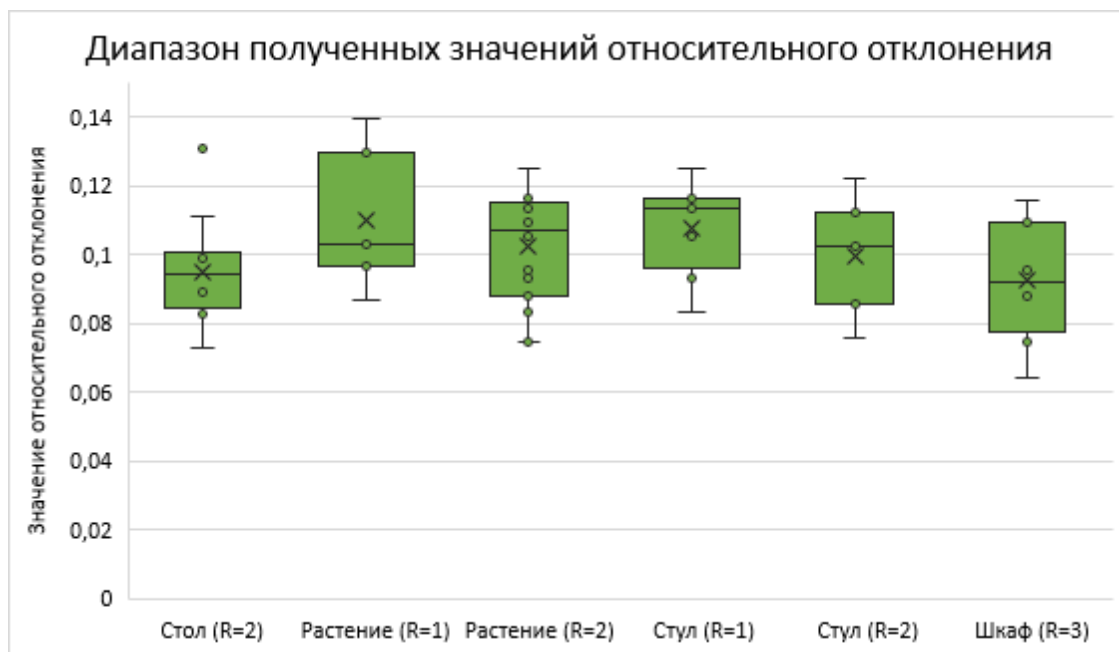


Рисунок 46 – Диапазон полученных значений относительного отклонения при определении пространственных положений объектов

Из данных, представленных на рисунке выше, можно заметить, что разработанный подход демонстрирует наилучшие усредненные значения относительного отклонения для объектов классов шкаф (R=3) и стол (R=2). Усредненные значения относительного отклонения для объектов данных классов составили соответственно 0.0919 и 0.0932. Несколько худшие значения относительного отклонения демонстрирует разработанный подход для объектов классов стул и растение (R=2), полученные значения составили соответственно 0.1001 и 0.1039. Наихудшие результаты по определению пространственного положения объектов в рамках данного эксперимента разработанный подход продемонстрировал для объектов классов растение и стул (R=1), усредненные значения относительного отклонения для данных классов составили соответственно 0.1098 и 0.1071. Отметим, что наименьшие усредненные значения относительного отклонения определения пространственных положений наблюдаются для категорий объектов размеров $R = 3$ и $R = 2$. При переходе к меньшим размерам целевых объектов ($R = 1$) наблюдается увеличение усредненных значений относительного отклонения. Данный рост может быть объяснен снижением точности выделения сегмента объекта на изображении при уменьшении размера исследуемого объекта, а также более высокой удельной долей ошибки, возникающей в процессе восстановления карт глубины. Следует также отметить, что оценка дисперсии значений относительного отклонения для объектов класса растение (R=1,2) значительно выше, а для объектов класса стол (R=2) – ниже, чем в среднем по выборкам. Такое распределение может быть объяснено геометрическими особенностями данных объектов, в частности, высокой сложностью формы растений и простотой формы стола соответственно.

Ниже на рисунке 47 представлен диапазон полученных значений относительного отклонения при определении пространственного положения объектов при различных уровнях освещенности сцены.

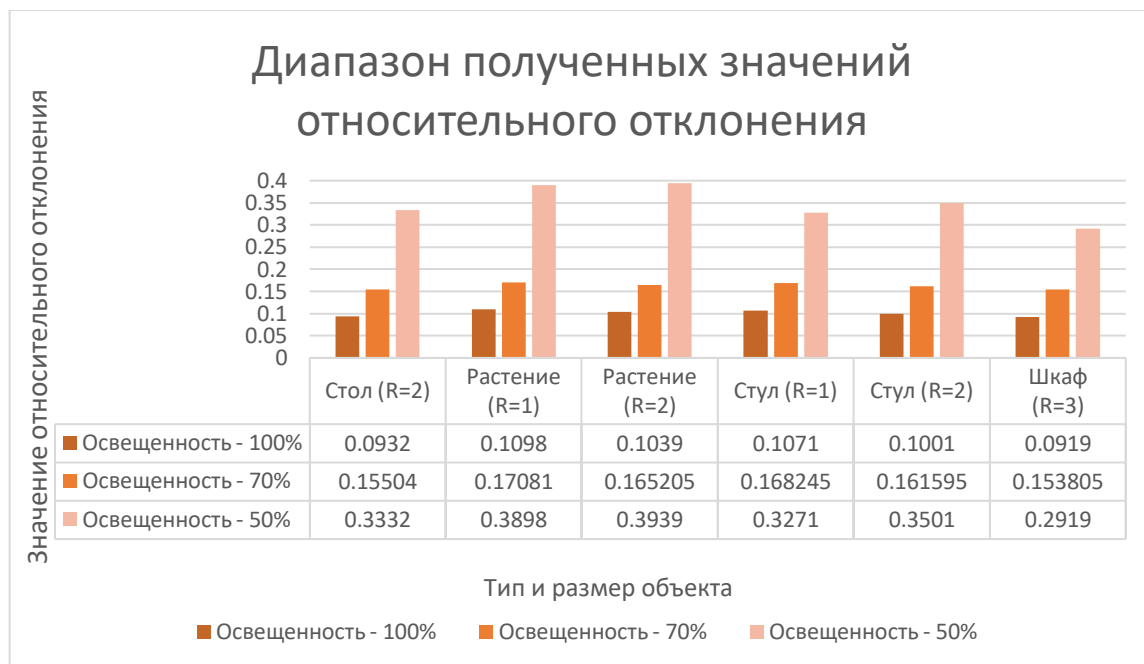


Рисунок 47 – Диапазон полученных значений относительной погрешности определения размеров объектов для различных уровней освещенности сцены

Из данных, представленных на рисунке выше, можно заметить, что разработанный подход демонстрирует наиболее высокую точность определения пространственного положения объектов для изображений со 100% и 70% уровнями освещенности. Усредненные по классам значения относительного отклонения для сцен с уровнями освещенности 100% и 70% равны 0,1010 и 0.1624 соответственно. При снижении освещенности наблюдается критическое снижение качества определения пространственного положения объектов на сцене, и уже при 50% уровне освещенности подход становится непригоден к практическому применению. Усредненное по классам значение относительного отклонения для сцен с уровнем освещенности 50% равно 0.3477. На основании полученных результатов можно заключить, что разработанный подход демонстрирует довольно высокую точность определения пространственного положения объектов для изображений со 100% и 70% уровнями освещенности.

По результатам апробации предложенного подхода к определению пространственного положения и размеров объектов сцены по изображению на тестовом наборе из 12000 изображений предложенное решение показало приемлемый уровень точности определения размеров и пространственного положения объектов для изображений с уровнями освещенности 100% и 70%.

Усредненные по классам значения относительной погрешности определения размеров объектов для соответствующих наборов изображений составили соответственно 0.1449 и 0.3313, а усредненные по классам значения

относительного отклонения при определении пространственного положения объектов для данных наборов изображений составили – 0,1010 и 0.1711.

3.4. Алгоритм получения трехмерных представлений и оценки пространственных характеристик объектов сцены с использованием сервисных РС

В рамках выполнения проекта также был разработан алгоритм получения трехмерных представлений и оценки пространственных характеристик объектов сцены с использованием сервисных РС, направленный на уточнение полученных ранее (с использованием подхода, представленного выше) оценок пространственных характеристик целевых объектов, а также определение формы данных объектов. Рассмотрим ключевые шаги данного алгоритма:

1. Запрашиваются актуальные данные в отношении объекта Ob_i для которого требуется выполнить оценку его характеристик с использованием РС:

$$Ob_i = \langle ID_i, Cl_i, R_i, Pos_i \rangle,$$

где ID_i – уникальный идентификатор объекта в КФП; Cl_i – класс объекта ($Cl_i \in Cl, Cl = \{Cl_j \mid j = \overline{1, n}\}$); R_i – линейный размер объекта, представляющий собой вектор, компоненты которого соответствуют оценкам длин проекций объекта Ob_i на оси координат трехмерного пространства; Pos_i – пространственное положение объекта, определенное как оценка положения центроида данного объекта.

2. Запрашивается информация в отношении актуального положения сервисного РС RS_i , которому была поставлена соответствующая задача по оценке пространственных характеристик объекта Ob_i :

$$RS_i = \langle RSID_i, RSPos_i \rangle,$$

где $RSID_i$ – уникальный идентификатор сервисного РС; $RSPos_i$ – пространственное положение данного РС, представленное вектором, описывающим положение данного устройства относительно некоторой системы отсчета в КФП.

3. В зависимости от класса целевого объекта Ob_i , определяется радиус региона интереса $RReg_i$. В общем случае регион интереса Reg_i представляет собой окрестность некоторого радиуса $RReg_i$ вокруг предполагаемого местоположения объекта, оценка которого получена средствами КФП. Радиус $RReg_i$ определяется в соответствии со следующим выражением:

$$RReg_i = \gamma * M(\{|R(Ob_k)| \mid Cl(Ob_k) == Cl(Ob_i), k \in (1, \dots, K)\}),$$

где K – число зарегистрированных на территории КФП объектов.

Представленное выше выражение представляет собой медианную оценку линейных размеров всех объектов класса, к которому принадлежит целевой объект Ob_i , помноженную на некоторый коэффициент γ . Конечное выражение для γ определяется в рамках первичной настройки системы мониторинга и зависит от габаритов РС

RS_i , реализующего выполнение задачи по оценке характеристик, и углов обзора фиксирующего устройства данного РС.

4. Определяем первичное представление региона интереса Reg_i как окружность в плоскости XY радиуса $RReg_i$ с центром в точке, соответствующей оценке пространственного положения целевого объекта $Pos(Ob_i)$. Совмещаем первичное представление региона интереса Reg_i с двумерной картой территории КФП – Map . К результирующему представлению региона интереса $FReg_i$ относятся лишь такие области A_j , которые одновременно принадлежат первичному представлению региона интереса, а также двумерной карте территории КФП:

$$FReg_i = \{A_j | A_j \in Reg_i, A_j \in Map\}.$$

Таким образом, из результирующего представления региона интереса $FReg_i$ исключаются все области, находящиеся вне территории КФП.

5. На следующем шаге настоящего алгоритма с использованием подхода к глобальному планированию движения и оценке местности для наземных роботов в универсальных средах [16] осуществляется формирование маршрута движения сервисного РС RS_i и последующее перемещение данного РС по полученному маршруту до региона интереса $FReg_i$.
6. После того, как регион интереса $FReg_i$ был достигнут, с использованием прямого монокулярного SLAM-алгоритма LSD-SLAM [17] осуществляется исследование результирующего региона интереса $FReg_i$. По результатам применения LSD-SLAM формируется трехмерная карта $PCloud_i$ участка сцены, соответствующего двумерному региону интереса $FReg_i$, в форме облака точек трехмерного евклидова пространства. Таким образом, полученное трехмерное представление исследуемого участка сцены может быть представлено в следующем виде:

$$PCloud_i = \{P_{ij} | j = \overline{1, N}\},$$

где N – число точек, принадлежащих облаку точек, являющемуся трехмерным представлением исследуемого участка сцены; P_{ij} – точка в трехмерном евклидовом пространстве, ассоциированная с исследуемым участком сцены.

7. Набор кадров видеоряда, полученных с фиксирующего устройства РС RS_i , которые были использованы в процессе формирования облака точек $PCloud_i$ обозначим как Img_i :

$$Img_i = \{Img_{im} | m \in (1, \dots, M)\},$$

где M – число уникальных кадров видеоряда, использованных в процессе формирования облака точек $PCloud_i$.

8. Для каждого кадра Img_{im} из набора Img_i известна позиция $RSPos_m$ и ориентация $RSrot_m$ РС RS_i , в которых оно находилось в момент фиксации такого кадра. Таким образом каждый кадр Img_{im} однозначно

ассоциирован с некоторым пространственным положением РС, которое описывается позицией РС $RSPos_m$ и ориентацией РС $RSrot_m$:

$$Img_{im} \rightarrow RSPos_m, Img_{im} \in Img_i, m \in (1, \dots, M);$$

$$Img_{im} \rightarrow RSrot_m, Img_{im} \in Img_i, m \in (1, \dots, M).$$

9. В отношении каждого пространственного положения РС RS_i , характеризуемого позицией $RSPos_m$ и ориентацией $RSrot_m$, можно однозначно определить соответствующую позицию $CPos_m$ и ориентацию $Crot_m$ фиксирующей видеокамеры, установленной на данном РС. Таким образом, каждый кадр Img_{im} является однозначно ассоциированным с положением фиксирующей видеокамеры данного РС, описываемым позицией $CPos_m$ и ориентацией $Crot_m$.
10. На следующем шаге в рамках настоящего алгоритма для каждого кадра Img_{im} из набора Img_i с использованием нейросетевой модели Mask R-CNN [13] осуществляется сегментация соответствующего изображения с целью выделения на нем сегментов целевого объекта Ob_i . Результирующее сегментированное изображение обозначим как $SegImg_{im}$, а выделенные нейросетевой моделью сегменты как Seg_{imw} :

$$SegImg_{im} \rightarrow \{Seg_{imw} | Seg_{imw} \in SegImg_{im}, w \in \overline{1, W}\},$$

где W – число уникальных сегментов, выделенных на изображении $SegImg_{im}$.

Каждый сегмент изображения Seg_{imw} , в свою очередь, представляет собой некоторый набор пикселей данного изображения, ассоциированный с некоторым распознанным нейросетевой моделью объектом:

$$Seg_{imv} = \{Pix_{imu} | u = 1, \dots, U\},$$

где U – число пикселей изображения $SegImg_{im}$, принадлежащих сегменту Seg_{imw} .

11. Далее, для каждого сегментированного изображения $SegImg_{im}$, однозначно ассоциированного с кадром Img_{im} , зная параметры и пространственное положение фиксирующего устройства ($CPos_m, Crot_m$), ассоциированное с кадром Img_{im} , можно определить область на трехмерной карте исследуемого участка сцены, охватываемую данным фиксирующим устройством, и соответственно сформировать отображение пикселей Pix_{imq} сегментированного изображения $SegImg_{im}$ в точки трехмерного участка исследуемой сцены. Таким образом, с каждым пикселем Pix_{imq} ассоциирован некоторый набор точек $PSet_{imq}$ трехмерного участка исследуемой сцены:

$$f: Pix_{imq} \rightarrow PSet_{imq},$$

где Pix_{imq} – некоторый пиксель сегментированного изображения $SegImg_{im}$; $PSet_{imq}$ – набор точек, принадлежащих облаку точек $PCloud_i$, ассоциированный с данным пикселем; q – число уникальных точек в трехмерном пространстве, ассоциированных с пикселем Pix_{imq} .

12. На следующем шаге в отношении каждого сегментированного изображения $SegImg_{im}$ осуществляется поиск сегментов Seg_{imv} ,

соответствующих целевому объекту Ob_i . Некоторый сегмент Seg_{imv} признается связанным с целевым объектом в случае, если выполняется следующее условие:

$$Cl(Seg_{imv}) == Cl(Ob_i), \quad (37)$$

где $Cl(Seg_{imv})$ – класс объекта, с которым, согласно используемой сегментационной нейросетевой модели Mask R-CNN, связан данный сегмент.

Набор сегментов изображения $SegImg_{im}$, для которых справедливо выражение (1) обозначим как $SegSet_{im}$:

$$SegSet_{im} = \{Seg_{imv} | Cl(Seg_{imv}) == Cl(Ob_i), v = \overline{1, V}\},$$

V – число уникальных сегментов на изображении $SegImg_{im}$, для которых справедливо выражение (1).

13. Поскольку каждый сегмент из набора $SegSet_{im}$, представляет собой набор пикселей, на следующем шаге алгоритма осуществляется определение набора $PixSet_{im}$ уникальных пикселей изображения $SegImg_{im}$, которые были признаны связанными с объектом(-ами) целевого класса:

$$PixSet_{im} = Seg_{im1} \cup \dots \cup Seg_{imV}, \forall Seg_{imv} \in SegSet_{im}, v = \overline{1, V}.$$

14. Далее, в отношении каждого полученного набора уникальных пикселей $PixSet_{im}$ определяется результирующий набор уникальных точек в трехмерном пространстве $PSet_{im}$, которые по результатам обработки изображения $SegImg_{im}$, были признаны принадлежащими объекту(-ам) целевого класса:

$$PSet_{im} = \{f(Pix_{im1}) \cup \dots \cup f(Pix_{imC}) | Pix_{imc} \in PixSet_{im}, c = 1, \dots, C\},$$

где C – число уникальных пикселей изображения $SegImg_{im}$, которые были признаны связанными с объектом(-ами) целевого класса.

15. На следующем шаге настоящего алгоритма, в отношении каждой точки P_{ij} трехмерного пространства, принадлежащей облаку точек $PCloud_i$ исследуемого участка сцены, инициализируется частотный показатель $Count_{ij}$. Данный показатель призван отражать число вхождений точки P_{ij} ($P_{ij} \in PCloud_i$) в наборы вида $PSet_{im}$. Определим функцию $G(PSet, P)$ так, что:

$$\begin{cases} G(PSet, P) = 1, P \in PSet \\ 0 \end{cases}.$$

Тогда значение показателя $Count_{ij}$, ассоциированного с некоторой точкой P_{ij} может быть найдено следующим образом:

$$Count_{ij} = \sum_{m=1}^M G(PSet_{im}, P_{ij}).$$

16. Далее осуществляется формирование результирующего набора точек трехмерного пространства $PRezSet_i$, принадлежащих объекту(-ам) целевого класса. С целью повышения устойчивости получаемых результатов точка P_{ij} считается принадлежащей набору $PRezSet_i$, если частотный показатель $Count_{ij}$, ассоциированный с данной точкой превосходит некоторое минимальное значение η :

$$P_{ij} \in PRezSet_i, \text{ если: } \begin{cases} Count_{ij} \geq \eta, P_{ij} \rightarrow Count_{ij} \\ P_{ij} \in PCloud_i, j \in (1, \dots, N) \end{cases} \quad (38)$$

Таким образом, набор $PRezSet_i$ может быть определен следующим образом:

$$PResSet_i = \{P_{il} | P_{il} \in PCloud_i, l = \overline{1, L}\},$$

где L – число уникальных точек в трехмерном пространстве, принадлежащих облаку точек $PCloud_i$, для которых справедливо выражение (2).

17. Поскольку существует ненулевая вероятность того, что в исследуемом регионе интереса $FReg_i$, помимо целевого объекта Ob_i , могут быть и иные объекты аналогичного класса, то точки из набора $PRezSet_i$ не могут быть однозначно интерпретированы, как точки, ассоциированные с целевым объектом. Для решения данной проблемы, а конкретно для идентификации как групп точек, связанных с различными объектами целевого класса, так и определения занимаемых ими областей трехмерного пространства на исследуемом участке сцены воспользуемся гипотезой, что точки одного и того же объекта Ob_a формируют область, плотность точек внутри которой существенно превышает плотность за ее пределами.

Приведенная выше формулировка задачи, во многом совпадает с формулировкой задачи в рамках методов кластеризации по плотности элементов. В данном случае в качестве элементов выборки выступают точки P_{il} , принадлежащие набору $PRezSet_i$, а в качестве потенциальных кластеров – объекты целевого класса Ob_a , расположенные на исследуемом участке сцены.

Для решения приведенной выше задачи в рамках настоящего алгоритма, было принято решение использовать метод кластеризации DBSCAN [18]. Радиус окрестности R и пороговое число точек в окрестности N_{crit} выбираются таким образом, чтобы число выбросов по результатам кластеризации точек из набора $PRezSet_i$ было минимальным.

По результатам кластеризации в отношении каждой точки P_{il} из набора $PRezSet_i$ формируется ассоциированный показатель $ClustID_{il}$, характеризующий идентификатор кластера, к которому относится данная точка.

18. На следующем шаге осуществляется формирование наборов точек $ObPSet_a$, ассоциированных с некоторыми конкретными объектами Ob_a , выделенными на исследуемом участке сцены по результатам кластеризации. Некоторая точка P_{il} из набора $PRezSet_i$ включается в набор $ObPSet_a$, если выполняется следующий набор условий:

$$\begin{cases} ClustID_{il} == ClustID_a \\ P_{il} \rightarrow ClustID_{il} \\ ObPSet_a \rightarrow ClustID_a \end{cases}.$$

По результатам выполнения данного шага в отношении каждого выделенного на исследуемом участке местности объекта целевого класса Ob_a формируется соответствующий набор точек $ObPSet_a$.

19. Далее, осуществляется оценка пространственных положений каждого выделенного на исследуемом участке местности объекта целевого класса Ob_a . Оценка пространственного положения каждого такого объекта выполняется в соответствии со следующим выражением:

$$Pos(Ob_a) = \frac{\sum_{h=1}^{H_a} P_{ih} | P_{ih} \in ObPSet_a}{H_a},$$

где H_a – число точек трехмерного пространства, принадлежащих набору $ObPSet_a$, ассоциированному с объектом Ob_a .

20. На следующем шаге осуществляется идентификация соответствия между целевым объектом Ob_i и некоторым выделенным объектом целевого класса Ob_a . Обозначим набор задетектированных объектов целевого класса как $ObDet$. Тожественным целевому объекту Ob_i признается такой объект Ob_a из набора $ObDet$, для которого представленное ниже выражение принимает минимальное значение:

$$|Pos(Ob_i) - Pos(Ob_a)|.$$

Таким образом, по завершении данного шага, однозначно идентифицируется представление целевого объекта Ob_i – объект Ob_a , выделенный на исследуемом участке сцены. Кроме того, с целевым объектом Ob_i становится ассоциированным набор точек трехмерного пространства $ObPSet_a$.

21. Дальнейшие шаги настоящего алгоритма посвящены уточнению пространственных характеристик целевого объекта Ob_i , на основе информации об ассоциированных с ним точках трехмерного пространства P_{ih} ($P_{ih} \in ObPSet_a, h=\overline{1, H_a}$).

В первую очередь осуществляется корректировка оценки пространственного положения объекта:

$$Pos(Ob_i) = Pos(Ob_a), Ob_a \rightarrow Ob_i.$$

Далее обновляются значения оценки линейных размеров R_i данного объекта:

$$R(Ob_i) = (R_{xi}, R_{yi}, R_{zi}),$$

$$R_{xi} = \left| \frac{\sum_{j=1}^n R_{xminj}}{n} - \frac{\sum_{j=1}^n R_{xmaxj}}{n} \right|;$$

$$R_{yi} = \left| \frac{\sum_{j=1}^n R_{yminj}}{n} - \frac{\sum_{j=1}^n R_{ymaxj}}{n} \right|;$$

$$R_{zi} = \left| \frac{\sum_{j=1}^n R_{zminj}}{n} - \frac{\sum_{j=1}^n R_{zmaxj}}{n} \right|,$$

где R_{xmin} , R_{ymin} , R_{zmin} – наборы из n минимальных значений проекций точек из набора $ObPSet_a$ на оси координат x , y и z соответственно; R_{xmax} ,

$R_{y_{max}}, R_{z_{max}}$ – наборы из n максимальных значений проекций точек из набора $ObPSet_a$ на оси координат x, y и z соответственно.

Форма объекта Ob_i представляет собой конечный неупорядоченный набор точек в трехмерном пространстве, связанных с исследуемым объектом и, в свою очередь, тождественна непосредственно набору точек $ObPSet_a$:

$$F(Ob_i) = ObPSet_a.$$

Таким образом, представленный выше алгоритм за счет применения сервисного РС RS_i , а также ряда методов и подходов к анализу визуальных данных, обеспечивает обновление оценок пространственных характеристик целевого объекта Ob_i , полученных с использованием средств КФП, а также позволяет осуществить определение формы такого объекта.

Результатом работы алгоритма является обновленный набор значений параметров целевого объекта Ob_i : $\langle ID_i, Cl_i, R_i, Pos_i, F_i \rangle$, где R_i и Pos_i – обновленные оценки линейного размера и пространственного положения целевого объекта соответственно, а F_i – форма целевого объекта (набор точек в трехмерном пространстве, связанных с исследуемым объектом). Таким образом, представленный выше алгоритм за счет применения сервисного РС RS_i , а также ряда методов и подходов к анализу визуальных данных, обеспечивает обновление оценок пространственных характеристик целевого объекта Ob_i , полученных с использованием средств КФП, а также позволяет осуществить определение формы такого объекта.

4. Модель мониторинга и управления процессом манипуляций над объектами в киберфизическом интеллектуальном пространстве

В ходе выполнения заявленного в плане работ научного исследования по данному проекту на 2020 год был дополнительно разработан и апробирован метод отслеживания процессов взаимодействия пользователей с объектами на видеопоследовательностях.

Разработка метода отслеживания процессов взаимодействия пользователей с объектами (human-object interaction) по видеопоследовательностям предполагает решение следующей группы задач: детектирование на кадрах видеопоследовательности пар вида «пользователь – объект взаимодействия», идентификация факта начала взаимодействия, трекинг соответствующих пар до завершения процесса взаимодействия.

В настоящее время существует большое количество подходов к решению задачи детектирования объектов на изображении и видеопоследовательности [19, 20]. Одним из наиболее перспективных решений в данной области, согласно исследованиям [13, 21], является предобученная нейронная сеть (NN) Mask R-CNN, которая способна не только осуществлять детектирование объектов на изображении, но и определять классовую принадлежность детектируемых объектов. К типам объектов, которые данная сеть способна идентифицировать, относятся различные категории офисной мебели, иные предметы, а также непосредственно люди. Таким образом, данная нейронная сеть может быть использована в качестве основы для решения задачи по идентификации пользователей и объектов взаимодействия. Однако формирование корректных пар «пользователь-объект взаимодействия» требует применения дополнительных алгоритмических моделей к результатам работы данной нейронной сети.

В контексте задачи детектирования начала взаимодействия пользователя с объектом существует большое количество методов, направленных на идентификацию типов взаимодействия. Авторами [22] предложена модель Interact-Net, определяющая на фотографиях сложных повседневных сцен тип взаимодействия человека с объектами. Модель обучена на сложном наборе данных V-COCO и решает задачу распознавания триплетов (человек, действие, объект), локализуя контуры человека, связанного объекта взаимодействия и идентифицируя выполняемое действие. Аналогичный проект представили авторы статьи [23]. Они предложили модель HOI (Human-Object Interaction) для обнаружения взаимодействия человека с объектом, которая в отличие от сложных сквозных моделей использует внешние признаки от предварительно обученных детекторов объектов и кодирует макет с помощью созданных вручную ограничительных элементов координат (необязательно ключевых точек позы человека). Для определения на изображении 18 ключевых точек каждого человека авторы используют модель Open Pose [24], которая определяет ключевые точки скелета человека на изображении или в видеопотоке. Рассмотренные выше

методы [23, 25] обладают достаточно высокой точностью определения вида взаимодействия человека с объектом на основе внешних показателей, но для их успешного применения необходимо, чтобы пользователь и объект в кадре были видны полностью. Недостатком обоих методов также является их ориентация на работу с одиночными изображениями без возможности работы с видеопотоком. Кроме того, в рамках данных методов предполагается, что на входных данных взаимодействие между пользователем и объектом уже осуществляется, таким образом, данные решения не могут быть использованы для решения задачи детектирования взаимодействия между пользователем и объектом.

Альтернативный подход к идентификации факта начала взаимодействия пользователя с объектом может быть основан на идентификации соприкосновения пользователя с объектом. Для выявления данных ситуаций необходимо осуществлять оценку взаимного расположения объекта и рук пользователя в пространстве, при этом пересечение сегментов рук и объекта на изображении является необходимым, но недостаточным условием совпадения их пространственных положений в связи с отсутствием информации об удаленности объекта от пользователя по оси перпендикулярной плоскости изображения. Данная информация может быть получена, в частности, на основе анализа результатов работы различных методов построения карт глубины на соответствующем изображении. Таким образом, реализация надежной идентификации факта начала взаимодействия пользователя с объектом требует комбинированного решения задач по детектированию рук пользователя, детектированию объекта, а также анализа результатов работы выбранного метода построения карт глубины на соответствующем изображении для установления факта соприкосновения пользователя с объектом.

Большинство решений в области детектирования рук пользователя, сконцентрированы на проблеме определения той или иной позы/жеста руки [26-31], например, в работе [31] исследователи сделали акцент на обучении модели распознаванию языка жестов. Наиболее подходящим решением с точки зрения поставленных в настоящем исследовании задач является модель, предложенная в статье [32], поскольку она ориентирована на выделении сегментов рук пользователя, требует небольшого количества данных для обучения и при этом характеризуется довольно высоким показателем точности детектирования.

Методы построения карт глубины могут быть классифицированы на две группы: аппаратные [33] и методы, основанные на использовании нейросетевых моделей [34-36]. Во избежание лишних затрат на специальную аппаратуру, поддерживающую возможность съемки с собственной картой глубины, была выбрана модель FCRN-DepthPrediction [36]. Данная модель позволяет строить карту глубины по 2D изображению. Преимуществом FCRN-DepthPrediction в сравнении с другими решениями данной группы является небольшое количество параметров обучения (train parameters), из чего следует,

что для обучения требуется меньше данных, чем для аналогичных методов. Архитектура данной сети построена на основе остаточной нейронной сети ResNet-50 [37] и дополняет её Upsampling слоями, которые повышают точность предсказанных карт глубины.

Ожидается, что совместное использование выбранного метода, осуществляющего сегментацию рук пользователя, в совокупности с применением нейронной сети Mask R-CNN, реализующей выделение границ объектов, и нейросетевой модели FCRN-DepthPrediction, обеспечивающей построение карты глубины изображений, позволит достоверно устанавливать факты соприкосновения пользователя с объектом и соответственно обеспечить решение задачи идентификации факта начала взаимодействия пользователя с объектами.

В контексте обеспечения трекинга пар пользователь-объект на видеопоследовательности до завершения процесса взаимодействия пользователя с объектом рассмотрим существующие методы трекинга объектов на видеопоследовательности. На сегодняшний день широкое распространение получили методы, основанные на использовании нейросетевых моделей [38-41]. По результатам проведенного анализа наиболее перспективным представляется решение, предложенное в работе [39]. Авторами была разработана модель, представляющая собой глубокую нейронную сеть, которая принимает в качестве входных данных видеокadres, а возвращает для каждого кадра координаты ограничивающего прямоугольника отслеживаемого объекта. Представленный метод расширяет глубокую сверточную нейронную сеть YOLO [42] в пространственно-временную область с использованием рекуррентных нейронных сетей (ROLO). Авторами была проведена обширная эмпирическая оценка эффективности ROLO в сравнении с 10 существующими решениями задачи трекинга. Сравнительная оценка проводилась на наборе из 30 сложных общедоступных видеопоследовательностей. Результат оценки значений удачных построений за единичный проход (OPE) представленной в работе модели равен 0.458. В целом все представленные выше методы демонстрируют достаточно высокую точность трекинга объектов на видеопоследовательности, однако наибольшей эффективности при сохранении высокого уровня точности обладает метод [39]. Данное решение может быть использовано в режиме реального времени, что положительным образом сказывается на применимости метода к реальным условиям и обуславливает выбор данного решения для обеспечения трекинга пар пользователь-объект взаимодействия на видеопоследовательности.

Таким образом, за счет комбинирования и модернизации рассмотренных выше подходов, методов и моделей в рамках данного проекта разработан метод отслеживания процессов взаимодействия пользователей с объектами по видеопоследовательностям, не требующий применения специализированных аппаратных решений и отличающийся возможностью использования в режиме

реального времени при сохранения высокого уровня точности детектирования и трекинга.

В соответствии с результатами проведенного анализа связанных методов и подходов, для обеспечения отслеживания взаимодействия пользователей с объектами предложен авторский метод решения данной задачи, включающий в себя следующие основные этапы:

1. Детектирование на кадрах видеопоследовательности пар вида пользователь-объект взаимодействия:
 - a. Детектирование пользователей и объектов с использованием предобученной модели нейронной сети Mask R-CNN;
 - b. Определение потенциальных пар пользователь-объект взаимодействия с использованием разработанного алгоритма предварительной идентификации взаимодействия;
2. Идентификация фактов начала взаимодействия:
 - a. Получение данных о пользователе – детектирование и сегментация рук пользователя с использованием нейросетевой модели Yolo v3;
 - b. Формирование карты глубины для кадров видеопоследовательности с помощью нейронной сети FCRN-DepthPrediction;
 - c. Идентификация фактов взаимодействия с использованием разработанного алгоритма, основанного на оценке пространственного положения рук пользователей относительно потенциальных объектов взаимодействия;
3. Анализ процессов взаимодействия пользователей с объектами:
 - a. Трекинг пар пользователь-объект в процессе взаимодействия за счет применения метода отслеживания объектов на основе рекуррентных нейронных сетей Rolo;
 - b. Идентификация завершения процессов взаимодействия;

Как можно заметить, предложенный метод содержит три ключевых этапа, каждый из которых направлен на решение отдельной группы подзадач. Важным требованием к разрабатываемому решению является обеспечение его работоспособности в режиме реального времени. Таким образом, первый этап применяется к поступающим кадрам видеоряда и направлен не только на идентификацию наборов пользователей и объектов на исследуемой сцене, но также и на снижение объема данных, для которых потребуется осуществлять более глубокую и ресурсоемкую обработку. Вторым этапом посвящен идентификации пользователей, взаимодействующих с теми или иными объектами на сцене. Важно подчеркнуть, что потенциально каждый пользователь может взаимодействовать с некоторым набором объектов, что выдвигает дополнительные требования к точности производимой оценки в отношении наличия взаимодействия в рамках каждой конкретной пары пользователь-объект. Третий этап связан с отслеживанием тех пар пользователь-объект, для которых было подтверждено наличие взаимодействия. В рамках данного этапа осуществляется трекинг

соответствующих пар, а также производится периодическая проверка на завершение процессов взаимодействия. Рассмотрим детальнее каждый из этапов предложенного метода.

4.1 Детектирование на кадрах видеопоследовательности пар вида пользователь – объект взаимодействия

Для детектирования пользователей и объектов на кадрах видеоряда в качестве модели применяется предобученная нейронная сеть Mask R-CNN, отличительной особенностью которой является возможность получения как ограничивающих прямоугольников, так и при необходимости масок для детектируемых объектов. Mask R-CNN основана на архитектуре сети ResNet-101 и способна классифицировать 92 типа объектов с точностью 0.9. Для определения объектов на сцене на вход нейронной сети подается одиночный кадр из видеопотока, предварительно очищенный от шумов за счет применения гауссовой фильтрации. Результатом работы нейронной сети является информация об объектах на изображении: координаты ограничивающих прямоугольников, маски, вероятности принадлежности каждого объекта к определенному классу. Важно отметить, что к числу классов, распознаваемых Mask R-CNN, также относится и человек, таким образом, по результатам работы данной нейронной сети выполняется детектирование не только объектов, но и непосредственно пользователей. Далее осуществляется определение потенциальных пар пользователь – объект взаимодействия, данные пары выявляются в соответствии со следующим алгоритмом:

1. Для каждого детектированного на некотором изображении пользователя U_i из набора U_{img} , вычислим центр соответствующего ему ограничивающего прямоугольника:

$$Center_i(Box_i) = \frac{(B_x + A_x, B_y + A_y)}{2}, \{Box_i \in Box_{img} | U_{img} \rightarrow Box_{img}\},$$

где Box_i – ограничивающий прямоугольник, соответствующий детектированному пользователю U_i , A и B – координаты диагональных вершин ограничивающего прямоугольника Box_i , U_{img} – набор детектированных на изображении пользователей.

2. Для каждого детектированного на некотором изображении объекта O_i из набора O_{img} вычислим центр соответствующего ему ограничивающего прямоугольника:

$$OCenter_i(OBox_i) = \frac{(B_x + A_x, B_y + A_y)}{2},$$

$$\{OBox_i \in OBox_{img} | O_{img} \rightarrow OBox_{img}\},$$

где $OBox_i$ – ограничивающий прямоугольник, соответствующий детектированному объекту O_i , A и B – координаты диагональных вершин ограничивающего прямоугольника $OBox_i$, O_{img} – набор детектированных на изображении объектов.

3. Далее для каждого элемента из наборов Box_{img} и $OBox_{img}$ произведем оценку размера соответствующих ограничивающих прямоугольников согласно следующему выражению:

$$Size(Box) = \frac{\|(B_x - A_x, B_y - A_y)\|}{2},$$

где A и B – координаты диагональных вершин соответствующего ограничивающего прямоугольника из наборов Box_{img} или $OBox_{img}$.

4. Для каждого пользователя U_i из набора U_{img} определим расстояния между данным пользователем и объектами $O_j \in O_{img}$, детектированными на исследуемой сцене. В качестве расстояния между пользователем U_i и объектом O_j будем использовать расстояния между центрами соответствующих им ограничивающих прямоугольников:

$$Dist(U_i, O_j) = \|Center_i - OCenter_j\|.$$

5. Определение потенциального набора пар вида пользователь – объект взаимодействия $PairSet_{pot}$ осуществляется в соответствии со следующим выражением:

$$Pair(U_i, O_j) = \begin{cases} 1, & \frac{Dist(U_i, O_j)}{Size(Box_i) + Size(OBox_j)} \leq \beta \\ 0, & \frac{Dist(U_i, O_j)}{Size(Box_i) + Size(OBox_j)} > \beta \end{cases}$$

$Pair(U_i, O_j) \in PairSet_{pot}$

где β – некоторое строго положительное пороговое значение, характеризующее предельную удаленность пользователя U_i от объекта O_j , при которой объект рассматривается в качестве потенциального объекта взаимодействия для данного пользователя.

По результатам работы данного алгоритма формируется набор потенциальных пар пользователь–объект взаимодействия, которые подлежат дальнейшему анализу на предмет фактического наличия взаимодействия.

4.2 Идентификация факта начала взаимодействия

Для идентификации факта взаимодействия пользователя с объектом необходимо предварительно произвести оценку пространственного положения рук каждого пользователя относительно соответствующих потенциальных объектов взаимодействия, что требует предварительного решения задач по детектированию рук пользователя, детектированию объекта, а также построению карты глубины соответствующего изображения.

Для детектирования и сегментации рук пользователя на изображении была выбрана модель YOLO v3, характеризующаяся высокими показателями точности своей работы и относительно низкими требованиями с точки зрения вычислительных ресурсов для обеспечения собственного функционирования в реальном времени. Входными данными модели является кадр

видеопоследовательности, а выходными – сегментированное изображение, с выделенными контурами рук пользователей, а также соответствующими ограничивающими прямоугольниками.

Полученные результаты применения модели масштабируются с целью формирования матрицы соответствия пикселей входного изображения выявленным сегментам рук пользователей. Данная матрица совмещается с результатами детектирования пользователей, полученными на предыдущем этапе, по принципу перекрытия областей с целью обеспечения связности между полученными контурами рук и сформированным ранее набором пользователей U_{img} . Таким образом, каждый выявленный на изображении сегмент руки Arm_k ассоциирован с конкретным пользователем U_i .

Так как детектирование потенциальных объектов взаимодействия было выполнено в рамках предыдущего этапа, далее для оценки относительного положения объектов и рук пользователей в трехмерном пространстве необходимо построить карту глубины текущего кадра видеопоследовательности. По результатам проведенного анализа существующих методов построения карты глубины было принято решение воспользоваться моделью нейронной сети FCRN-DepthPrediction, выходными данными которой является матрица значений глубины. Проведение операции масштабирования данной матрицы до размера исходного изображения позволяет обеспечить соответствие пикселей входного изображения и полученных значений глубины.

Следующий шаг заключается в оценке положения рук каждого пользователя относительно соответствующих потенциальных объектов взаимодействия в рамках определённых на предыдущем этапе пар пользователь – объект взаимодействия. Для каждой пары $P_n = (U_i; O_j)$ из набора потенциальных пар пользователь – объект взаимодействия $PairSet_{pot}$ оценка пространственного положения объектов относительно рук пользователей производится в соответствии со следующим алгоритмом идентификации взаимодействия:

1. Для каждой руки $Arm_k \in Arm_{img}$, детектированной на кадре видеопоследовательности и ассоциированной с некоторым пользователем U_i , определим центр соответствующего ограничивающего прямоугольника:

$$ACenter_k(ABox_k) = \frac{(B_x + A_x, B_y + A_y)}{2},$$

$$\{ABox_k \in ABox_{img} \mid Arm_{img} \rightarrow ABox_{img}\}$$

где $ABox_k$ – ограничивающий прямоугольник, соответствующий детектированной руке Arm_k , A и B – координаты диагональных вершин ограничивающего прямоугольника $ABox_k$, Arm_{img} – набор задетектированных на изображении рук.

2. Определим вектор, характеризующий положение центра объекта O_j относительно центра руки $ACenter_k$, следующим образом:

$$direct_n = OCenter_j - ACenter_k$$

3. В общем случае карту глубины можно рассматривать как дискретно заданную функцию F , определенную на множестве пар (x, y) , где x и y – значения индексов пикселей исследуемого кадра видеопоследовательности по горизонтальной и вертикальной осям соответственно. В качестве характеристики изменения удаленности руки Arm_k и объекта O_j в направлении перпендикулярном плоскости изображения возьмем двустороннюю разностную оценку производной данной функции dF по направлению $direct_n$. Произведем расчет данных оценочных значений на отрезке $[ACenter_k, OCenter_j]$ в соответствии со следующим выражением:

$$\frac{dF}{d(direction_n)}(Point_q) = \frac{F(Point_0 + \gamma \cdot \frac{direction_n}{|direction_n|}) - F(Point_0 - \gamma \cdot \frac{direction_n}{|direction_n|})}{2 \cdot \gamma},$$

где γ – параметр шага, а $Point_q = (x_q, y_q)$ – q -ый пиксель изображения, который принадлежит отрезку $[ACenter_k, OCenter_j]$.

4. Результирующий вывод относительно наличия взаимодействия между пользователем и объектом осуществляется в соответствии со следующим выражением:

$$\forall Point_q \in Points, \left| \frac{dF}{d(direction_n)}(Point_q) \right| \leq \theta$$

где $Points$ – набор пикселей изображения, которые принадлежат отрезку $[ACenter_k, OCenter_j]$, θ – некоторое строго положительное пороговое значение, характеризующее предельное изменение величины удаленности при смещении вдоль отрезка $[ACenter_k, OCenter_j]$ и выполняющее роль критерия идентификации резких изменений значений глубины в данном направлении. В случае, если представленное выше выражение истинно, делается вывод о наличии взаимодействия пользователя с объектом в рамках пары $P_n = (U_i; O_j)$.

Таким образом, по итогам данного этапа для каждой пары P_n из набора $PairSet_{pot}$ делается окончательный вывод о наличии взаимодействия между пользователями и объектами, в результате чего формируется набор пар вида пользователь – объект взаимодействия $PairSet_{res}$.

4.3 Анализ процессов взаимодействия пользователей с объектами

Как было сказано ранее, данный этап связан с отслеживанием тех пар пользователь-объект, для которых было подтверждено наличие взаимодействия. В рамках данного этапа осуществляется трекинг соответствующих пар, а также производится периодическая проверка на завершение процессов взаимодействия. Трекинг пар пользователь-объект в процессе их взаимодействия обеспечивается за счет применения метода отслеживания объектов на основе рекуррентных нейронных сетей Rolo. На вход данному методу подается информация об отслеживаемом объекте – соответствующий ограничивающий прямоугольник, а также поступающие кадры исследуемой видеопоследовательности. Для каждого поданного кадра

данный метод возвращает обновленные параметры ограничивающего прямоугольника для отслеживаемого объекта. Таким образом, обеспечивается отслеживание всех пар P_n из результирующего набора $\text{PairSet}_{\text{res}}$. Проверка на завершение взаимодействия для каждой пары P_n инициируется на каждом j -ом кадре видеопоследовательности и реализуется в соответствии с алгоритмом идентификации взаимодействия, рассмотренном ранее в рамках второго этапа. По результатам выполнения данной проверки определяется актуальный набор взаимодействующих пар пользователь-объект $\text{PairSet}_{\text{final}}$.

Обобщенная алгоритмическая модель разработанного метода отслеживания процессов взаимодействия пользователей с объектами представлена ниже на рисунке 48.

Важно отметить, что представленная выше алгоритмическая модель предполагает анализ лишь каждого m -го кадра видеопоследовательности. При этом доля анализируемых кадров может быть как увеличена для достижения большей точности отслеживания, так и уменьшена с целью минимизации затрачиваемых вычислительных ресурсов. Кроме того, данная алгоритмическая модель учитывает возможность возникновения случаев множественного взаимодействия как некоторого пользователя с группой объектов, так и нескольких пользователей с одним объектом за счет предложенной реализации процесса выявления потенциальных пар пользователь-объект. Ниже на рисунке 49 представлены поэтапные результаты работы сформированной модели на некотором кадре видеопоследовательности.

На рис. 49.а представлен обрабатываемый кадр видеопоследовательности. Согласно разработанной модели, в первую очередь осуществляется детектирование пользователей и объектов с использованием нейросетевой модели Mask R-CNN (рис. 49.б). На данном кадре был детектирован один пользователь и четыре объекта: три чашки и один рюкзак. Далее, в соответствии с разработанным алгоритмом, выполняется поиск потенциальных пар пользователь-объект взаимодействия, результаты работы данного алгоритма отражены на рис. 49.в. Было определено 2 потенциальные пары: пользователь – рюкзак, пользователь – чашка (ограничивающий прямоугольник отображен на рисунке). После того, как потенциальные пары взаимодействующих пользователей и связанных с ними объектов определены, выполняется детектирование и сегментация рук пользователей с использованием нейросетевой модели Yolo v3. На рис. 49.г представлены найденные на данном кадре видеопоследовательности сегменты рук пользователей.

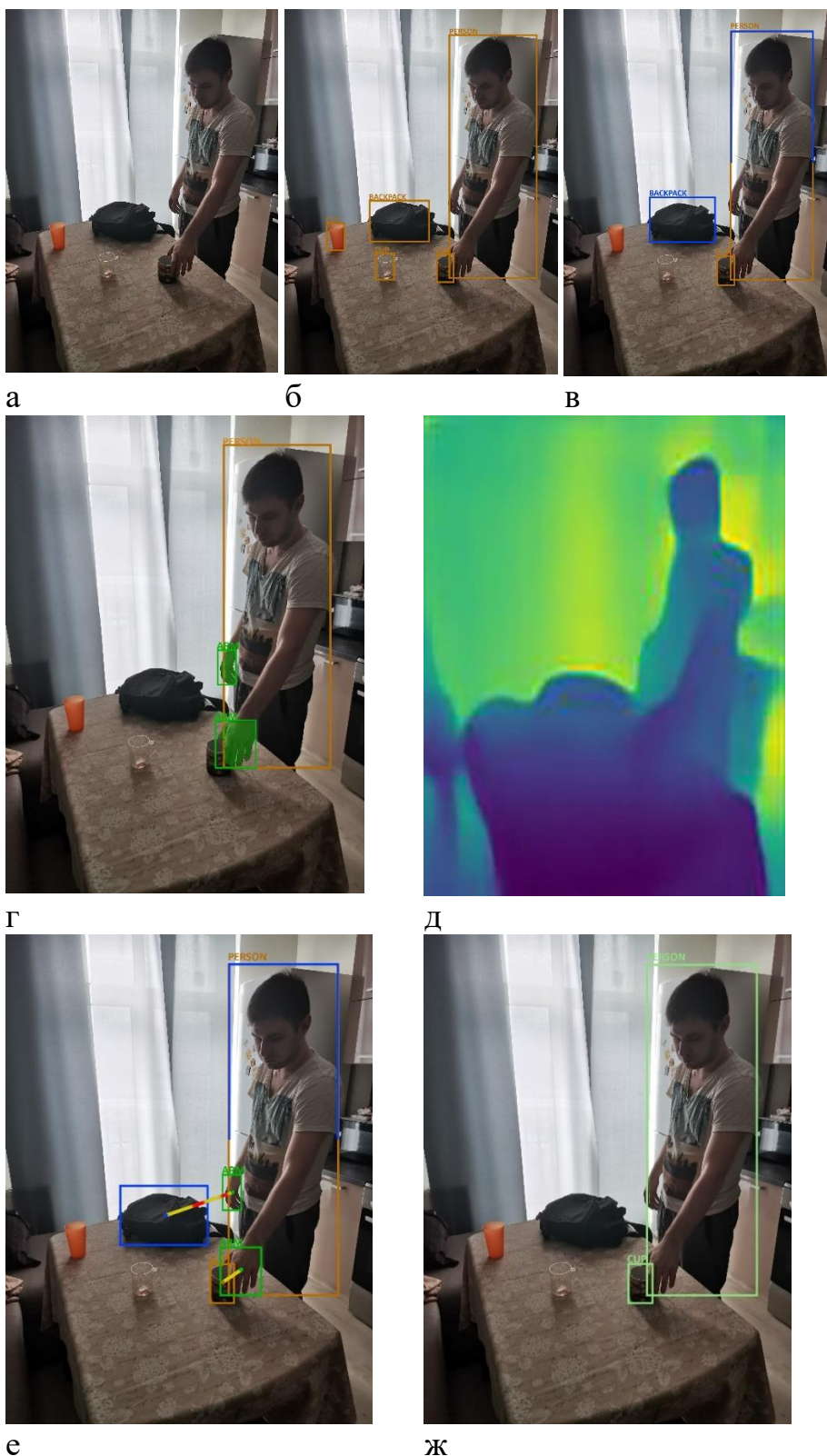


Рисунок 49 – Поэтапные результаты работы предложенной алгоритмической модели на некотором кадре видеопоследовательности: а) исходный кадр видеопоследовательности; б) результаты детектирования пользователей и объектов; в) результаты выявления потенциальных пар пользователь - объект взаимодействия г) результаты детектирования и сегментации рук пользователей; д) результаты формирования карты глубины для данного кадра видеоряда; е) результаты оценки производной функции dF в соответствии с алгоритмом идентификации взаимодействий; ж) результат идентификации взаимодействий между потенциальными парами пользователь – объект

Следующим шагом является построение карты глубины для соответствующего изображения, полученные с помощью нейронной сети FCRN-DepthPrediction результаты представлены на рис. 49. д. Далее производится анализ отдельных участков полученной карты глубины, и, в частности, осуществляется двусторонняя разностная оценка производной функции dF по направлению, связывающему центр руки пользователя и центр соответствующего объекта. Результаты, полученные для данного кадра видеопоследовательности, представлены на рис. 49.е. и отражены соответствующими линиями, при этом цвет линии по умолчанию является желтым, но на участках, где оценка производной функции dF превосходит заданное пороговое значение, цвет линии становится красным. Для потенциальной пары пользователь-рюкзак данная линия имеет красные участки, что, в соответствии с логикой разработанной модели, свидетельствует об отсутствии взаимодействия пользователя с данным объектом. Таким образом, в качестве фактически взаимодействующих пар, моделью выделена лишь пара пользователь-чашка (ограничивающий прямоугольник отображен на рис. 49.ж). Данный результат работы алгоритма идентификации взаимодействия следует считать корректным, так как в действительности взаимодействие пользователя с рюкзаком или иным объектом сцены, помимо чашки, отсутствует.

Далее перейдем к оценке реализации предложенного метода в контексте качества детектирования и отслеживания взаимодействий пользователей с объектами на видеопоследовательностях.

Апробация и оценка качества функционирования разработанного метода отслеживания процессов взаимодействия пользователей с объектами производилась на основе тестового набора данных, включающего в себя 1000 видеопоследовательностей длительностью до 20 секунд. В тестовом наборе данных присутствовали видеопоследовательности как с одним, так и с несколькими пользователями, взаимодействующими с объектами сцены. На каждой тестовой видеопоследовательности представлено как минимум одно взаимодействие одного из следующих типов: один пользователь – один объект, один пользователь – несколько объектов, несколько пользователей – один объект. Объекты взаимодействия представляют собой объекты 15 различных классов, подлежащих детектированию нейронной сетью Mask R-CNN. Тестовый набор данных состоит из видеопоследовательностей, сформированных при различных условиях освещенности: 30%, 50% и 100%, где за 100% взят нормативный уровень освещенности для офисных помещений.

На основе данных, полученных в результате применения разработанного метода к набору тестовых видеопоследовательностей, были сформированы различные количественные оценки точности работы предложенного решения. В частности, были определены усредненные показатели точности (ассурасу, recall, precision) детектирования взаимодействий в разрезе классов объектов

взаимодействия и уровней освещенности. Значения соответствующих показателей были вычислены следующим образом:

1. Для каждой видеопоследовательности на основе анализа результатов применения двух первых этапов разработанного метода были сформированы матрицы ошибок идентификации взаимодействия. Данные матрицы M_{ij} формировались независимо для каждого класса объектов (табл.7).

Таблица 7 – Матрицы ошибок

	Взаимодействие присутствует	Взаимодействие отсутствует
Алгоритм показал наличие взаимодействия	TP_{ij}	FP_{ij}
Алгоритм не показал наличия взаимодействия	FN_{ij}	TN_{ij}

В данной матрице ячейка TP_{ij} – число верно детектированных взаимодействий пользователей с объектами класса j на i -ой видеопоследовательности; FP_{ij} – число ложно детектированных взаимодействий пользователя с объектом класса j на i -ой видеопоследовательности; FN_{ij} – число случаев, когда алгоритм ложно показал отсутствие взаимодействия пользователя с объектом класса j на i -ой видеопоследовательности; TN_{ij} – число верно определенных случаев отсутствия взаимодействия пользователей с объектами класса j на i -ой видеопоследовательности.

2. Далее для каждой матрицы M_{ij} были определены показатели ассюрасу A_{ij} , recall R_{ij} и precision Pr_{ij} в соответствии со следующими формулами:

$$A_{ij} = \frac{TP_{ij} + TN_{ij}}{TP_{ij} + FP_{ij} + FN_{ij} + TN_{ij}};$$

$$R_{ij} = \frac{TP_{ij}}{TP_{ij} + FN_{ij}};$$

$$Pr_{ij} = \frac{TP_{ij}}{TP_{ij} + FP_{ij}}.$$

3. На заключительном шаге осуществлялось усреднение данных показателей в разрезе отдельных классов объектов $j = \{1 \dots 15\}$ и уровней освещенности сцены $L = \{30\%, 50\%, 100\%\}$. С этой целью из общего набора видеопоследовательностей формировались специфические наборы видеопоследовательностей $V_{set_{jL}}$, включающие в себя лишь видеопоследовательности V_k с заданным уровнем освещенности L и содержащие на сцене объекты j -го класса. Таким образом, усредненные показатели точности A_{jL} , R_{jL} , Pr_{jL} были определены в соответствии со следующими выражениями:

$$A_{jL} = \frac{1}{n_{jL}} \sum_{k=1}^{n_{jL}} A_{jL}(V_k), \quad V_k \in V_{set_{jL}};$$

$$R_{jL} = \frac{1}{n_{jL}} \sum_{k=1}^{n_{jL}} R_{jL}(V_k), \quad V_k \in V_{set_{jL}};$$

$$Pr_{jL} = \frac{1}{n_{jL}} \sum_{k=1}^{n_{jL}} Pr_{jL}(V_k), \quad V_k \in V_{set_{jL}};$$

Ниже на рис. 50 представлены диаграммы полученных значений показателей A_{jL} , R_{jL} , Pr_{jL} для различных классов объектов j при разных уровнях освещенности сцены L .

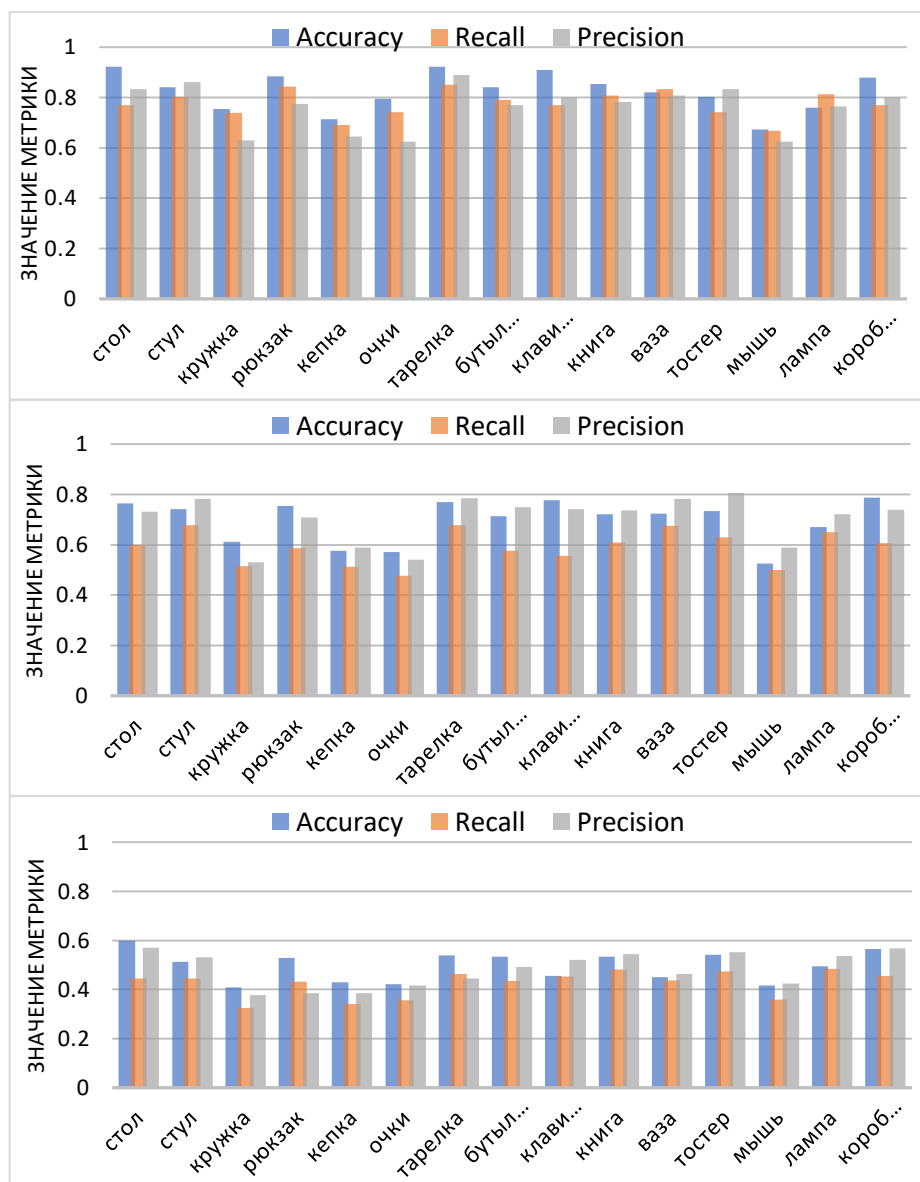


Рисунок 50 – Диаграммы полученных значений показателей A_{jL} , R_{jL} , Pr_{jL} для различных классов объектов j при уровнях освещенности сцены L : 100% (а), 50% (б), 30% (в)

Из представленных выше рисунков можно заключить, что точность работы метода существенным образом зависит от уровня освещенности и значительно снижается при понижении уровня освещенности. При этом стоит отметить, что при переходе от уровня освещенности в 100% к 50% наибольшее снижение испытывает показатель R_{jL} , при следующем переходе к 30% уровню освещенности все показатели испытывают примерно равную степень снижения. Полученные результаты могут быть объяснены тем, что понижение

уровня освещенности сказывается в первую очередь на качестве детектирования объектов. Таким образом, пропускается часть потенциальных пар пользователь-объект, при этом качество идентификации взаимодействий ухудшается в меньшей степени, что обуславливает меньшую долю ошибок вида FP. Однако при еще больших снижениях уровня освещенности помимо ошибок детектирования, возникают также значительные погрешности при построении карт глубины, что существенным образом сказывается на качестве идентификации взаимодействий. Данная ситуация наблюдается при применении разработанного метода на видеопоследовательностях с уровнем освещенности в 30%. Важно отметить, что класс объекта также влияет на качество детектирования взаимодействий, в частности рассчитанные показатели имеют более низкие значения для объектов следующих классов: кружка, кепка, очки, мышь. Предполагается, что подобные результаты связаны с меньшим размером данных объектов, а также с их геометрическими особенностями.

Усредненные по классам значения показателей A_{jL} , R_{jL} , Pr_{jL} для видеопоследовательностей с различными уровнями освещенности равны $\{0.82, 0.78, 0.76\}$, $\{0.70, 0.59, 0.70\}$, $\{0.49, 0.43, 0.48\}$ для 100%, 50% и 30% уровней освещенности соответственно. В целом на основе полученных результатов можно заключить, что разработанный метод демонстрирует приемлемое качество детектирования взаимодействий для видеопоследовательностей с уровнями освещенности в 100% и 50%.

Для осуществления оценки непосредственно качества отслеживания процессов взаимодействия пользователей с объектами было принято решение в качестве основной метрики воспользоваться долей корректно отслеженных взаимодействий. Взаимодействие рассматривалось как корректно отслеженное в случае, если:

- взаимодействие было задетектировано не позднее чем через время $\Delta t = 0.05T$ от фактического начала взаимодействия, где T соответствует интервалу времени, когда взаимодействие фактически имело место;
- временной интервал фиксации взаимодействия посредством разработанного метода соответствует реальному не менее, чем на 70% с нулевым смещением;
- момент времени завершения взаимодействия, полученный с использованием разработанного метода, соответствует фактическому с отклонением не более чем $\pm 0.05T$.

Ниже на рисунке 51 представлена диаграмма доли корректно отслеженных взаимодействий для каждой видеопоследовательности из тестового набора.

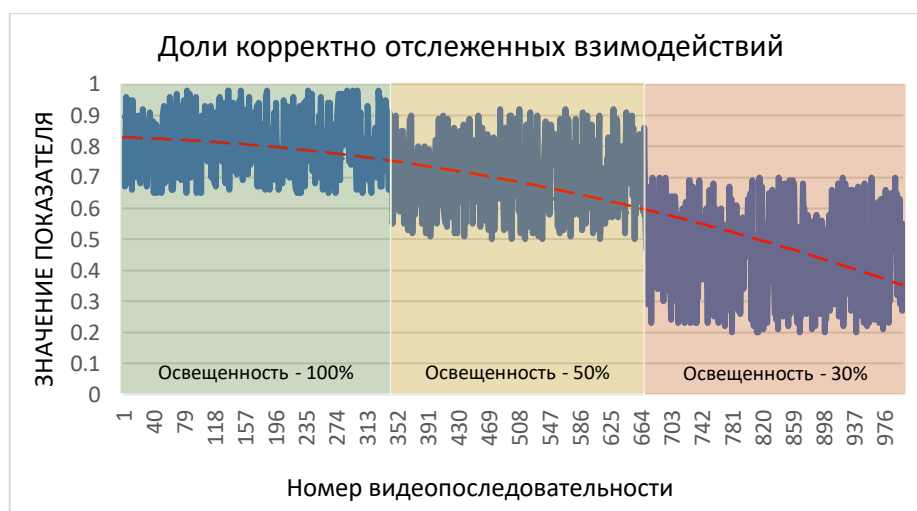


Рисунок 51 – Диаграмма доли корректно отслеженных взаимодействий для тестового набора видеопоследовательностей

Разработанный метод демонстрирует довольно высокую точность отслеживания взаимодействий для видеопоследовательностей со 100% уровнем освещенности. При этом по аналогии с качеством детектирования взаимодействий при использовании предложенного метода на видеопоследовательностях с более низким уровнем освещенности точность отслеживания взаимодействий становится существенно ниже, что проиллюстрировано на рисунке выше соответствующей линией тренда. Усредненная доля корректно отслеженных взаимодействий для наборов видеопоследовательностей с уровнем освещенности 100%, 50% и 30% составила соответственно 0.81, 0.71 и 0.45, а оценка среднеквадратического отклонения данной величины 0.09, 0.12, 0.15. Полученные результаты свидетельствуют о том, что со снижением уровня освещенности имеет место не только снижение точности работы предложенного решения, но и повышение вариабельности результатов. Тем не менее, качество работы метода остается на приемлемом уровне для видеопоследовательностей даже с 50% уровнем освещенности. Однако при дальнейшем снижении освещенности наблюдается критическое снижение качества отслеживания взаимодействий и уже при 30% уровне освещенности метод становится непригоден к практическому применению. Таким образом, предложенное решение позволяет успешно детектировать и отслеживать взаимодействия пользователей с различными классами объектов по видеопоследовательностям и в определенной мере является пригодным к использованию в условиях неполной освещенности сцены.

По результатам апробации предложенного метода отслеживания взаимодействий пользователей с объектами на тестовом наборе из 1000 видеопоследовательностей, предложенное решение показало довольно высокое качество детектирования и отслеживания взаимодействий для видеопоследовательностей с уровнями освещенности 100% и 50%. Усредненные показатели точности (accuracy, recall, precision) детектирования взаимодействий для соответствующих наборов видеопоследовательностей

имеют значение $\{0.82, 0.78, 0.76\}$ и $\{0.70, 0.59, 0.70\}$, а усредненные доли корректно отслеженных взаимодействий для данных наборов видеопоследовательностей составили 81% и 71%. Однако при более низких уровнях освещенности точность работы разработанного решения продемонстрировала значительное снижение – на наборе видеопоследовательностей с уровнем освещенности 30% усредненные показатели точности составили $\{0.49, 0.43, 0.48\}$, а доля корректно отслеженных взаимодействий снизилась до 45%. Полученные результаты могут быть объяснены тем, что при снижении уровня освещенности также существенно снижается качество функционирования заложенных в основу предложенного метода моделей. Таким образом, разработанный метод в определенной мере является устойчивым к изменению уровня освещенности сцены и обеспечивает успешное решение задачи детектирования и отслеживания взаимодействия пользователей с различными классами объектов по видеопоследовательности, не требуя при этом применения специализированного оборудования.

В рамках проекта контроль, управление, а также распределение задач между компонентами КФП и сервисными РС обеспечивается разработанной в рамках проекта алгоритмической моделью мониторинга и управления процессами манипуляций над объектами на территории КФП. В основе функционирования данной алгоритмической модели находятся, разработанные в рамках проекта и рассмотренные выше методы, подходы и алгоритмы, основанные на широком применении средств технического зрения и технологий машинного обучения, в том числе установленных непосредственно на сервисных РС, функционирующих на территории КФП.

В контексте настоящей модели каждый зарегистрированный объект Ob_i в КФП может быть охарактеризован конечным набором значений ряда параметров, к которым относятся: класс объекта, линейный размер объекта, форма объекта, пространственное положение объекта, уникальный идентификатор объекта, а также маркер верификации параметров посредством РС. Класс некоторого объекта Cl_i представляет собой некоторое определенное значение из конечного множества классов Cl , отслеживаемых системой:

$$Cl_i \in Cl, \\ Cl = \{Cl_1, Cl_2, \dots, Cl_n | j = \overline{1, n}\}.$$

В данном случае индекс n характеризует предельное число классов, которые могут быть определены рассматриваемой системой, таким образом, данный параметр может быть использован в качестве косвенной характеристики универсальности системы.

Линейный размер объекта $R(Ob_i)$ отражает оценку длин проекций объекта на оси координат в рамках трехмерного евклидова пространства и представляет собой кортеж из следующих значений:

$$R_i = R(Ob_i) = \langle L(Ob_i), W(Ob_i), H(Ob_i) \rangle,$$

где $L(Ob_i)$ – длина объекта, определяемая как длина проекции объекта Ob_i на ось x , $W(Ob_i)$ – ширина объекта, определяемая как длина проекции объекта Ob_i

на ось y , $H(Ob_i)$ – высота объекта, определяемая как длина проекции объекта Ob_i на ось z .

Форма объекта F_i может быть охарактеризована облаком точек соответствующего объекта и определена как некоторый конечный неупорядоченный набор точек в трехмерном пространстве, связанных с исследуемым объектом:

$$F_i = \{P_1, P_2, \dots, P_m | P_j \in Ob_i, j = \overline{1, m}\},$$

где P_j – точки в трехмерном евклидовом пространстве, в отношении которых установлена их принадлежность некоторому объекту Ob_i .

Пространственное положение объекта Pos_i в общем случае характеризует структуру распределения точек объекта в пространстве, в рамках настоящей работы в качестве обобщающей оценки распределения точек объекта в пространстве было принято решение использовать центрост. Таким образом, оценка пространственного положения объекта Ob_i задается следующим выражением:

$$Pos_i = \frac{\sum_{j=1}^m P_j | P_j \in Ob_i}{m}.$$

Уникальный идентификатор объекта ID_i присваивается каждому вновь зарегистрированному в системе объекту, при этом гарантируется, что в каждый момент времени, такой идентификатор однозначно ассоциирован с некоторым объектом Ob_i . Данный идентификатор присваивается по результатам первичной идентификации объекта и остается неизменным на протяжении всего времени нахождения объекта в КФП.

Для повышения точности определения пространственных характеристик объектов в дополнение к средствам КФП используются сервисные РС, применение которых позволяет повысить точность оценок, указанных выше параметров, характеризующих некоторый объект Ob_i . Маркер верификации параметров посредством РС M_i представляет собой бинарный идентификатор, связанный с объектом Ob_i и характеризующий факт проведения оценки параметров данного объекта с использованием сервисных РС.

Для контроля состояния объекта с точки зрения факта нахождения его на территории КФП используется параметр $ObSt_i$. Данный параметр представляет собой счетчик, значение которого соответствует числу последовательных циклов исполнения модели, в рамках которых данный объект не был задетектирован системой. При превышении данным параметром некоторого критического значения, соответствующий объект считается утерянным, а его отслеживание прекращается.

Таким образом, в общем случае каждый объект КФП может быть представлен кортежем следующего вида:

$$Ob_i = \langle Cl_i, R_i, F_i, Pos_i, ID_i, M_i, ObSt_i \rangle.$$

В рамках настоящей модели каждый пользователь КФП U_a описывается конечным набором значений следующих параметров: уникальный идентификатор пользователя и биометрические данные лица пользователя.

Уникальный идентификатор пользователя U_ID_a присваивается каждому пользователю в момент получения допуска на территорию КФП и является однозначно ассоциированным с данным пользователем U_a .

Биометрические данные лица пользователя U_face_a представляют собой набор векторов визуальных признаков данного пользователя, используемых нейросетевой моделью распознавания лиц в целях идентификации пользователей [43]. Соответствующие данные извлекаются в рамках процедуры получения допуска на территорию КФП и являются уникальными для каждого пользователя.

Таким образом, каждый пользователь U_a , допущенный на территорию КФП, может быть охарактеризован кортежем следующего вида:

$$U_a = \langle U_ID_a, U_face_a \rangle.$$

Согласно настоящей модели, представление сервисного РС RS_b реализуется набором значений следующих параметров: уникальный идентификатор РС и положение сервисного РС в пространстве. Уникальный идентификатор РС RS_ID_b присваивается каждому РС в момент его интеграции в КФП, такой идентификатор является однозначно ассоциированным с данным РС RS_b .

Пространственное положение сервисного РС RS_Pos_b представляется вектором, описывающим положение данного устройства относительно некоторой системы отсчета в КФП. Данное положение определяется с использованием алгоритмов инерциальной навигации [44], а также SLAM-алгоритмов [16] непосредственно самими РС.

Таким образом, любая сервисная РС, интегрированная в КФП, может быть представлена следующим кортежем:

$$RS_b = \langle RS_ID_b, RS_Pos_b \rangle.$$

Процесс взаимодействия (манипуляции) объектами в КФП также может быть охарактеризован конечным набором значений ряда параметров, ключевыми из которых являются: статус взаимодействия, продолжительность взаимодействия, тип взаимодействия, уникальный идентификатор объекта взаимодействия, уникальный идентификатор взаимодействующего субъекта, исходное пространственное положение объекта, конечное пространственное положение объекта.

Статус взаимодействия St_j является параметром, характеризующим актуальность данного процесса, и представляет собой бинарный идентификатор, единичное значение которого означает наличие взаимодействия в текущий момент времени, а ноль – факт завершения такого взаимодействия.

Продолжительность взаимодействия T_j отражает фактическую длительность взаимодействия, значение данного параметра при $St_j=1$ равно разнице между текущим моментом времени t и моментом начала данного взаимодействия t_0 , а при $St_j=0$ – разнице между моментами начала и завершения процесса взаимодействия соответственно.

Тип взаимодействия $Type_j$ представляет собой бинарный идентификатор, характеризующий природу взаимодействия, с точки зрения типа субъекта взаимодействия, $Type_j = 1$ свидетельствует о том, что субъектом взаимодействия является человек, $Type_j$ равен 0 в случае, если взаимодействие осуществляется со стороны сервисного РС.

Уникальный идентификатор взаимодействующего субъекта ME_j – в зависимости от типа управляющего субъекта однозначно ассоциирован с уникальным идентификатором пользователя в системе $ID(U_j)$ или же идентификатором сервисного РС $ID(RS_j)$.

Исходное пространственное положение объекта Pos_start_j . В качестве оценки положения объекта взаимодействия Ob_j в пространстве используется центроид. Таким образом, оценка исходного пространственного положения объекта может быть определена как:

$$Pos_start_j = Pos(Ob_j, t = t_0),$$

где $Pos(Ob, t)$ – функция, возвращаемое значение которой представляет собой вектор в трехмерном пространстве, возвращаемое значение соответствует пространственному положению некоторого объекта Ob в момент времени t ; Ob_j – объект взаимодействия; t_0 – время начала данного взаимодействия.

Конечное пространственное положение объекта Pos_fin_j . Оценка конечного пространственного положения объекта Ob_j может быть определена следующим образом:

$$Pos_fin_j = Pos(Ob_j, t = t_f),$$

где Ob_j – объект взаимодействия; t_f – время завершения данного взаимодействия.

Таким образом, в общем виде процесс взаимодействия (манипуляции) некоторым объектом Ob_i в КФП может быть представлен кортежем следующего вида:

$$Int_j = \langle St_j, T_j, Type_j, ID(Ob_i), ME_j, Pos_start_j, Pos_fin_j \rangle.$$

Каждому текущему взаимодействию между пользователем и объектом Int_j соответствует некоторый набор процессов трекинга такого взаимодействия $TrackSet_j$:

$$\{Int_j \rightarrow TrackSet_j | j = \overline{1, W}\},$$

где W – число текущих взаимодействий между пользователями и объектами КФП, зарегистрированных в рамках предыдущего цикла исполнения модели.

В свою очередь каждый процесс трекинга взаимодействия $Track_{jc}$, входящий в набор $TrackSet_j$, однозначно ассоциирован с некоторым средством КФП $Mean_c$ и представляет собой кортеж следующего вида:

$$Track_{jc} =$$

$$\langle ID(Ob_j), U_ID_j, Box(Ob_j)^{t-\Delta t}, Box(Ob_j)^t, Box(U_j)^{t-\Delta t}, Box(U_j)^t \rangle,$$

где $ID(Ob_j)$ – уникальный идентификатор объекта с которым осуществляется взаимодействие; U_ID_j – уникальный идентификатор пользователя U_j , осуществляющего данное взаимодействие; $Box(Ob_j)^{t-\Delta t}$ – положение ограничивающего прямоугольника, связанного с объектом взаимодействия, на

изображении, полученном средством КФП $Mean_c$ в момент времени $t-\Delta t$; $Box(Ob_j)^t$ – положение ограничивающего прямоугольника, связанного с объектом взаимодействия, на изображении, полученном средством КФП $Mean_c$ в момент времени t ; $Box(U_j)^{t-\Delta t}$ – положение ограничивающего прямоугольника, связанного с пользователем U_j , на изображении, полученном средством КФП $Mean_c$ в момент времени $t-\Delta t$; $Box(U_j)^t$ – положение ограничивающего прямоугольника, связанного с пользователем U_j , на изображении, полученном средством КФП $Mean_c$ в момент времени t ; t – текущий момент времени; Δt – временной интервал между предыдущим и текущим циклами исполнения модели.

Далее рассмотрим алгоритмическую модель мониторинга и управления процессом манипуляций над объектами в киберфизическом интеллектуальном пространстве. Предложенная алгоритмическая модель мониторинга и управления процессами манипуляций над объектами имеет циклическую природу и функционирует следующим образом:

1. Запрашивается актуальный набор зарегистрированных в КФП объектов – Ob , каждый элемент которого соответствует некоторому объекту Ob_i и включает в себя ассоциированный с данным объектом кортеж значений параметров $\langle Cl_i, R_i, F_i, Pos_i, ID_i, M_i, ObSt_i \rangle$.
2. Запрашивается набор текущих взаимодействий на территории КФП – Int , каждый элемент данного набора соответствует конкретному экземпляру взаимодействия Int_j и включает в себя ассоциированный с данным объектом кортеж значений параметров:

$$\langle St_j, T_j, Type_j, ID(Ob_i), ME_j, Pos_start_j, Pos_fin_j \rangle$$
3. Запрашивается набор допущенных на территорию КФП пользователей – U , каждый элемент которого соответствует некоторому пользователю КФП U_a и включает в себя ассоциированный с данным пользователем кортеж значений параметров: $\langle U_ID_a, U_face_a \rangle$.
4. Запрашивается набор интегрированных в КФП сервисных РС – RS , каждый элемент которого представляет конкретное РС RS_b и включает в себя ассоциированный с данным РС кортеж значений параметров: $\langle RS_ID_b, RS_Pos_b \rangle$.
5. С каждого активного средства КФП – $Mean_k$, осуществляющего сбор видеоданных, запрашиваются актуальные кадры Img_k с зарегистрированными на них участками территории КФП. Таким образом, формируется множество пар вида:

$$\{ \langle Mean_k, Img_k \rangle | k = \overline{1, K} \},$$
где K – число киберфизических средств видео-мониторинга на территории КФП.
6. Запрашиваются данные о текущих процессах трекинга взаимодействий пользователей с объектами в виде ассоциированных с взаимодействиями Int_j наборов $TrackSet_j$, каждый элемент которых представляет собой кортеж вида:

$$Track_{jc} = \langle ID(Ob_j), U_ID_j, Box(Ob_j)^{t-\Delta t}, Box(Ob_j)^t, Box(U_j)^{t-\Delta t}, Box(U_j)^t \rangle.$$

7. За счет применения подхода к определению пространственного положения и размеров объектов сцены по изображению, представленного выше, в отношении каждого актуального изображения Img_k , полученного средствами КФС, осуществляется детектирование и оценка пространственного положения объектов на территории КФП. В результате применения данного подхода в отношении каждого изображения Img_k формируется соответствующий набор Ob_set_k , каждый элемент которого является представлением некоторого объекта и содержит информацию о классе, положении ограничивающего прямоугольника и пространственном положении некоторого задетектированного на данном изображении объекта Ob_m :

$$Ob_set_k = \{ \langle Cl_m, Pos_m, Box_m \rangle | m = \overline{1, v_k} \}, \quad (39)$$

где v_k число распознанных на изображении Img_k объектов.

8. В результате объединения данных о распознанных на актуальных изображениях Img_k объектах, формируется единый набор распознанных представлений объектов Ob_rec :

$$Ob_rec = \{ Ob_set_1 \cup Ob_set_2 \cup \dots \cup Ob_set_K | k = \overline{1, K} \},$$

где K – число сформированных наборов представлений объектов, полученных по результатам распознавания объектов на актуальных изображениях. В рамках соответствующего процесса два представления вида (1) Ob_m и Ob_n признаются тождественными в случае, если выполняется следующий набор условий:

$$\begin{cases} Cl_m == Cl_n \\ |Pos_m - Pos_n| < \varepsilon(Cl_m) \end{cases}, \quad (40)$$

где $\varepsilon(Cl_m)$ – предельно допустимая величина отклонения пространственного положения объекта, индивидуальная для каждого класса Cl_i ($i = \overline{1, n}$).

Результирующее представление Ob_r ($Ob_r = \langle Cl_r, Pos_r, Box_r \rangle$) для множества представлений Ob_s ($s = \overline{1, S}$) некоторого объекта Ob_i , признанных тождественными согласно (2), формируется следующим образом:

$$\begin{cases} Cl_r = Cl_1 = \dots = Cl_S \\ Pos_r = \frac{1}{S} \sum_{s=1}^S Pos_s \\ Box_r = \{ \langle Box_s, Mean_s \rangle | s = \overline{1, S} \} \end{cases},$$

где Pos_r – результирующее пространственное положение объекта Ob_i , полученное в результате усреднения значений соответствующего параметра по всем тождественным представлениям такого объекта; Box_s – положение ограничивающего прямоугольника объекта Ob_i на некотором изображении Img_s ; $Mean_s$ – средство КФП, с которого такое изображение Img_s было получено; Box_r – набор пар транзитивно ассоциированных между собой ограничивающих прямоугольников Box_s

и средств КФС $Mean_s$, которые также ассоциированы с данным объектом Ob_i .

Таким образом, набор Ob_rec представляет собой общий набор уникальных результирующих представлений распознанных объектов типа Ob_r , конечный вид набора Ob_rec может быть представлен следующим образом:

$$Ob_rec = \{Ob_r | r = \overline{1, R}\},$$

где R – число уникальных результирующих представлений распознанных объектов.

9. Далее с использованием системы идентификации пользователей на основе нейросетевой модели распознавания лиц FaceNet и нейросетевой модели Mask-R-CNN, для каждого из актуальных изображений Img_k , полученного средствами КФС, реализуется идентификация пользователей КФП. В результате применения данной модели в отношении каждого изображения Img_k формируется соответствующий набор $USet_k$, каждый элемент которого содержит уникальный идентификатор U_ID_a идентифицированного пользователя U_a , а также параметры ограничивающего прямоугольника в отношении данного пользователя U_Box_a , характеризующего положение данного пользователя на соответствующем изображении Img_k :

$$USet_k = \{\{U_ID_a, UBox_a\} | a = \overline{1, \eta_k}\},$$

где η_k – число распознанных на изображении Img_k пользователей.

10. В результате объединения данных о распознанных на актуальных изображениях Img_k ($Img_k \rightarrow Mean_k$) пользователях, формируется единый набор задетектированных на территории КФП пользователей $UDet$, каждый элемент которого $UDet_d = \langle U_ID_d, UBox_d \rangle$ ассоциирован с некоторым пользователем U_d и может быть представлен в следующем виде:

$$UDet_d = \left\{ \begin{array}{l} U_ID_d \\ UBox_d = \{ \langle UBox_{dx}, Mean_x \rangle | x = \overline{1, X}, UBox_{dx} \rightarrow Mean_x \} \end{array} \right\}'$$

где $d = \overline{1, D}$, D – число задетектированных на территории КФП пользователей на текущем цикле исполнения модели.

11. Далее с использованием представленного выше метода отслеживания процессов взаимодействия пользователей с объектами на видеопоследовательностях в отношении каждой актуальной последовательности изображений $SImg_k$, полученной некоторым киберфизическим средством $Mean_k$, осуществляется детектирование взаимодействий пользователей с объектами на территории КФП. По результатам применения данного метода в отношении каждой актуальной видеопоследовательности $SImg_k$, полученной соответствующим средством КФП $Mean_k$, формируется набор $UIntSet_k$, каждый элемент которого является представлением некоторого задетектированного взаимодействия и содержит информацию об ограничивающем прямоугольнике объекта взаимодействия $Box(Ob_{ki})$ и

ограничивающем прямоугольнике пользователя $Box(U_{ki})$, взаимодействующего с данным объектом в момент времени t :

$$UIntSet_k = \{(Box(Ob_{k1}), Box(U_{k1})), \dots, (Box(Ob_{kl}), Box(U_{kl})) \mid i = \overline{1, l}\}.$$

12. В результате объединения полученных наборов $UIntSet_k$, характеризующих задетектированные взаимодействия, формируется единый набор, отражающий все задетектированные взаимодействия пользователей с объектами $UIntSet$:

$$UIntSet = \{UIntSet_1 \cup UIntSet_2 \cup \dots \cup UIntSet_K \mid k = \overline{1, K}\},$$

где K – число сформированных наборов взаимодействий, полученных по результатам их детектирования на актуальных видеопоследовательностях $SImg_k$ ($k = \overline{1, K}$).

Таким образом, набор $UIntSet$ представляет собой общий набор уникальных представлений взаимодействий пользователей с объектами, конечный вид набора $UIntSet$ может быть представлен в следующем виде:

$$UIntSet = \{(Box(Ob), Box(U))_q \mid q = \overline{1, Q}\},$$

где Q – число уникальных представлений задетектированных взаимодействий в момент времени t .

13. Далее в отношении каждого элемента $UIntSet_q$ набора $UIntSet$ осуществляется проверка на соответствие данного элемента какому-либо процессу трекинга взаимодействия, ассоциированному с взаимодействием Int_j , которое было определено как активное в рамках предыдущего цикла исполнения модели, $Int_j \in Int \mid j = \overline{1, W}$. Для проведения соответствующей проверки запрашивается ассоциированный таким взаимодействием Int_j набор активных процессов трекинга взаимодействия $TrackSet_j$. Каждый элемент $Track_{jc}$ набора $TrackSet_j$ проходит проверку на связанность с представлением задетектированного взаимодействия $UIntSet_q$. Представление взаимодействия $UIntSet_q$ признается связанным с процессом трекинга взаимодействия $Track_{jc}$ в случае, если выполняется следующий набор условий:

$$\begin{cases} IoU(Box(Ob)_q, Box(Ob)_{jc}^t) \geq \alpha \\ IoU(Box(U)_q, Box(U)_{jc}^t) \geq \beta \end{cases} \quad (41)$$

где IoU – метрика Intersection over Union; $Box(Ob)_q$ и $Box(Ob)_{jc}^t$ – ограничивающие прямоугольники объекта взаимодействия, $Box(Ob)_q \in UIntSet_q$, $Box(Ob)_{jc}^t \in Track_{jc}$ соответственно; $Box(U)_q$ и $Box(U)_{jc}^t$ – ограничивающие прямоугольники взаимодействующего пользователя, $Box(U)_q \in UIntSet_q$, $Box(U)_{jc}^t \in Track_{jc}$ соответственно; $\alpha, \beta \in [0.5, 1]$ – пороговые значения метрики IoU, конкретные значения которых определяются в рамках первичной настройки модели.

В случае выполнения условий (3) для некоторого представления $UIntSet_q$ и процесса трекинга взаимодействия $Track_{jc}$ процесс проверки

связанности в отношении данного представления взаимодействия завершается.

14. По результатам проведения представленной выше процедуры проверки в отношении каждого представления $U\text{IntSet}_q$ ($q=\overline{1, Q}$) из набора $U\text{IntSet}$, каждый процесс трекинга взаимодействия Track_{jc} становится либо ассоциированным с некоторым конкретным представлением взаимодействия $U\text{IntSet}_q$, либо остается неассоциированным. Таким образом, если для некоторого Track_{jc} справедливо выражение:

$$(\text{Track}_{jc} \rightarrow U\text{IntSet}) = \emptyset, \quad (42)$$

то данный процесс трекинга перестает быть актуальным и считается неассоциированным с текущими взаимодействиями. Такой процесс трекинга Track_{jc} подлежит прерыванию и последующему удалению из набора процессов трекинга TrackSet_j , ассоциированных с взаимодействием Int_j . Таким образом, после завершения данного шага, с каждым взаимодействием Int_j ($j=\overline{1, W}$) остаются ассоциированными исключительно актуальные процессы трекинга взаимодействия.

Набор задетектированных представлений $U\text{IntSet}_q$ ($U\text{IntSet}_q \in U\text{IntSet}$), для которых справедливо выражение:

$$(U\text{IntSet}_q \rightarrow \text{TrackSet}_j) \neq \emptyset \mid j \in (1, \dots, W), \quad (43)$$

обозначим как ActUIntSet . Элементами данного набора являются такие представления взаимодействий $U\text{IntSet}_q$, каждое из которых является однозначно ассоциированным с некоторым актуальным процессом трекинга Track_{jc} и соответственно ассоциированным с некоторым текущим взаимодействием Int_j . Представления взаимодействий $U\text{IntSet}_q$, в отношении которых выражение (5) является несправедливым, признаются новыми представлениями взаимодействий, набор соответствующих представлений обозначим как NewUIntSet .

15. На следующем шаге в рамках настоящей модели в отношении каждого представления взаимодействия из набора ActUIntSet осуществляется сопоставление такого представления ActUIntSet_p с представлениями распознанных объектов из набора Ob_rec . Представление взаимодействия ActUIntSet_p признается ассоциированным с представлением объекта Ob_rec_p в случае, если выполняется следующее условие:

$$\exists \text{Box}_{ps} \in \text{Box}_p: \text{IoU}(\text{Box}(\text{Ob})_p, \text{Box}_{ps}) \geq \alpha \mid s \in (1, \dots, S_p), \\ \text{Box}_p \in \text{Ob_rec}_p,$$

где Box_{ps} ($s=\overline{1, S_p}$) – ограничивающие прямоугольники, ассоциированные с представлением распознанного объекта Ob_rec_p .

По результатам выполнения данного шага, каждое представление взаимодействия ActUIntSet_p из набора ActUIntSet становится однозначно ассоциированным с некоторым представлением объекта Ob_rec_p , а также некоторым взаимодействием Int_j . Поскольку связанные с Int_j процессы трекинга взаимодействий Track_{jc} содержат информацию об

уникальном идентификаторе объекта взаимодействия $ID(Ob_j)$, на основе данной информации может быть определено однозначное соответствие между представлением объекта Ob_rec_p и самим объектом Ob_j . Далее, обновленные данные об объекте взаимодействия, содержащие в представлении Ob_rec_p , используются для актуализации значений параметров данного объекта Ob_j и параметров взаимодействия Int_j :

$$\begin{aligned} Pos(Ob_j) &= Pos(Ob_rec_p); \\ ObSt(Ob_j) &= 0; \\ Pos_fin(Int_j) &= Pos(Ob_rec_p); \\ T(Int_j) &= T(Int_j) + \Delta t. \end{aligned} \quad (44)$$

16. На следующем шаге в рамках настоящей модели в отношении каждого представления взаимодействия $NewUIntSet_z$ из набора $NewUIntSet$ осуществляется сопоставление такого представления $NewUIntSet_z$ с задетектированными на территории КФП пользователями $UDet_d$ из набора $UDet$. Представление взаимодействия $NewUIntSet_z$ признается ассоциированным с пользователем $UDet_d$ в случае, если выполняется следующее условие:

$$\begin{aligned} \exists UBox_{dx} \in UBox_d: IoU(Box(Ob)_z, UBox_{dx}) \geq \alpha \mid x \in (1, \dots, X_d), \\ UBox_d \in UDet_d, \end{aligned}$$

где $UBox_{dx} \ (x=\overline{1, X_d})$ – ограничивающие прямоугольники, ассоциированные с задетектированным на территории КФП пользователем $UDet_d$.

По результатам выполнения данного шага, каждое представление взаимодействия $NewUIntSet_z$ из набора $NewUIntSet$ становится однозначно ассоциированным с некоторым задетектированным на территории КФП пользователем $UDet_d$.

17. Далее в отношении каждого представления взаимодействия из набора $NewUIntSet$ осуществляется сопоставление такого представления $NewUIntSet_z$ с представлениями распознанных объектов из набора Ob_rec . Представление взаимодействия $NewUIntSet_z$ признается ассоциированным с представлением объекта Ob_rec_z в случае, если выполняется следующее условие:

$$\begin{cases} \exists Box_{zs} \in Box_z: IoU(Box(Ob)_z, Box_{zs}) \geq \alpha \mid s \in (1, \dots, S_p), \\ Box_z \in Ob_rec_z \end{cases},$$

где $Box_{zs} \ (s=\overline{1, S_z})$ – ограничивающие прямоугольники, ассоциированные с представлением распознанного объекта Ob_rec_z .

По результатам выполнения данного шага, каждое представление взаимодействия $NewUIntSet_z$ из набора $NewUIntSet$ становится однозначно ассоциированным с некоторым представлением объекта Ob_rec_z .

18. Следующим шагом является сопоставление представлений объектов Ob_rec_z и самих объектов Ob_z из набора Ob . Представление Ob_rec_z

признается ассоциированным с некоторым объектом Ob_z в случае, если выполняется следующий набор условий:

$$Ob_rec_z \rightarrow Ob_z, \text{ если: } \begin{cases} Cl(Ob_rec_z) == Cl(Ob_z) \\ |Pos(Ob_rec_z) - Pos(Ob_z)| < \varepsilon(Cl(Ob_z)) \end{cases} \quad (45)$$

Таким образом, каждое представление объекта Ob_rec_z , ассоциированное с некоторым представлением взаимодействия $NewUIntSet_z$ в случае, если выполняется условие (45), становится также однозначно ассоциированным с некоторым конкретным объектом Ob_z из набора Ob . Обозначим набор таких представлений взаимодействий $NewUIntSet_z$ как $rUIntSet$ ($rUIntSet \subseteq NewUIntSet$).

19. Далее в отношении каждого представления взаимодействия $rUIntSet_l$ из набора $rUIntSet$ осуществляется проверка, направленная на определение, является ли данное представление взаимодействия связанным с какими-либо уже существующими взаимодействиями Int_j ($j = \overline{1, W}$). Поскольку каждое представление $rUIntSet_l$ является однозначно ассоциированным с некоторым пользователем $UDet_l$, задетектированным на территории КФП, а также с некоторым конкретным объектом Ob_l из набора зарегистрированных на территории КФП объектов Ob , то соответствующая проверка может быть осуществлена на основе следующего выражения:

$$\exists Int_j: \begin{cases} ID_l == ID(Ob_j) | ID_l \in Ob_l, ID(Ob_j) \in Int_j \\ U_ID(UDet_l) == ME_j | U_ID(UDet_l) \in UDet_l, ME_j \in Int_j \end{cases} | j = \overline{(1, W)} \quad (46)$$

Каждое представление взаимодействия $rUIntSet_l$ из набора $rUIntSet$, в отношении которого справедливо выражение (46) является однозначно ассоциированным с некоторым текущим взаимодействием Int_j и связано с фактом детектирования такого взаимодействия со стороны некоторого средства КФП $Mean_k$, которым такое взаимодействие ранее не наблюдалось. В таком случае осуществляется инициализация нового процесса трекинга взаимодействия Int_j со следующими параметрами:

$$NewTrack_j = \langle ID_j, U_ID_j, Box(Ob)^t = Box(Ob)_l, Box(Ob)^{t+\Delta t}, \\ Box(U)^t = Box(U)_l, Box(U)^{t+\Delta t} \rangle,$$

где $ID_j, U_ID_j \in Int_j$; $Box(Ob)_l, Box(U)_l \in rUIntSet_l$; $Box(Ob)_l^{t+\Delta t}, Box(U)_l^{t+\Delta t}$ – соответствующие значения будут определены в рамках исполнения соответствующего процесса трекинга взаимодействия. После того, как процесс трекинга $NewTrack_j$ был инициирован, необходимо обновить набор актуальных процессов трекинга $TrackSet_j$, связанный с данным взаимодействием Int_j , а также актуализировать данные в отношении параметров ассоциированного объекта Ob_l и параметров взаимодействия Int_j :

$$\begin{aligned} TrackSet_j &= TrackSet_j \cup \{NewTrack_j\}; \\ Pos(Ob_l) &= Pos(Ob_rec_z), Ob_rec_z \rightarrow Ob_l; \\ ObSt(Ob_l) &= 0; \end{aligned}$$

$$\begin{aligned} Pos_{fin}(Int_j) &= Pos(Ob_{rec_z}); \\ T(Int_j) &= T(Int_j) + \Delta t. \end{aligned}$$

В отношении представлений взаимодействий $rUIntSet_l$ из набора $rUIntSet$, в отношении которых выражение (46) не является справедливым, можно заключить, что данные представления характеризуют новые взаимодействия, не наблюдавшиеся на территории КФП в рамках предыдущего цикла исполнения модели. В таком случае осуществляется инициализация нового взаимодействия Int_y со следующими параметрами:

$$\begin{aligned} Int_y &= \langle St_y=1, T_y=0, Type_y=1, ID_y = ID(Ob_l), ME_y=ID(UDet_l), \\ &Pos_{start_y}=Pos(Ob_{rec_z}), Pos_{fin_y}=Pos(Ob_{rec_z}) \rangle, \end{aligned}$$

где $Ob_{rec_z} \rightarrow Ob_l$.

Далее необходимо инициализировать соответствующий процесс трекинга $NewTrack_y$ взаимодействия Int_y со следующими параметрами:

$$\begin{aligned} NewTrack_y &= \langle ID_y, U_ID_y, Box(Ob)^t=Box(Ob)_l, Box(Ob)^{t+\Delta t}, \\ &Box(U)^t=Box(U)_l, Box(U)^{t+\Delta t} \rangle, \end{aligned}$$

где $ID_y, U_ID_y \in Int_y$; $Box(Ob)_l, Box(U)_l \in rUIntSet_l$; $Box(Ob)^{t+\Delta t}, Box(U)^{t+\Delta t}$ – соответствующие значения будут определены в рамках исполнения соответствующего процесса трекинга взаимодействия. После того, как процесс трекинга $NewTrack_y$ был инициирован, необходимо инициализировать набор актуальных процессов трекинга $TrackSet_y$, связанный с данным взаимодействием Int_y , а также актуализировать данные в отношении параметров ассоциированного объекта Ob_l :

$$\begin{aligned} TrackSet_y &= \{NewTrack_y\}, Int_y \rightarrow TrackSet_y; \\ Int &= Int \cup Int_y; \\ Pos(Ob_l) &= Pos(Ob_{rec_z}), Ob_{rec_z} \rightarrow Ob_l; \\ ObSt(Ob_l) &= 0. \end{aligned}$$

Поскольку факт инициализации нового взаимодействия Int_y , может повлиять на выполнение условий (46) в отношении некоторой группы представлений взаимодействий из набора $rUIntSet$, то после каждой инициализации некоторого взаимодействия Int_y , справедливость выражения (46) проверяется повторно для всех еще необработанных представлений взаимодействий из набора $rUIntSet$.

По результатам выполнения данного шага, в отношении каждого задетектированного на территории КФП взаимодействия, связанного с представлениями $rUIntSet_l$ из набора $rUIntSet$, обеспечивается актуализация соответствующих данных о таком взаимодействии и параметрах объектов соответствующих взаимодействий.

20. Обозначим все представления взаимодействий из набора $NewUIntSet$, в отношении которых согласно условиям (45) не были найдены соответствующие им объекты Ob_z из набора зарегистрированных объектов Ob , как $nUIntSet$ ($nUIntSet \subseteq NewUIntSet$). Каждое представление взаимодействия $nUIntSet_z$ из набора $nUIntSet$ признается

связанным с некоторым новым объектом $NewOb_z$, внесенным на территорию КФП. В тоже время, каждое представление $nUIntSet_z$ однозначно ассоциировано с некоторым представлением объекта Ob_rec_z из набора Ob_rec . Важно отметить, что сразу несколько представлений взаимодействий $nUIntSet_z$ могут быть связаны с одним и тем же представлением объекта Ob_rec_z . Таким образом, число новых объектов $NewOb_z$, внесенных на территорию КФП, соответствует числу уникальных представлений объектов Ob_rec_z , с которыми ассоциированы элементы набора $nUIntSet$, причем каждый такой объект $NewOb_z$ является однозначно ассоциированным с некоторым конкретным представлением объекта Ob_rec_z .

Далее, в отношении каждого Ob_rec_z такого, что $Ob_rec_z \rightarrow nUIntSet_z, nUIntSet_z \in nUIntSet$ выполняется инициализация экземпляра объекта $NewOb_z$ на основе представления Ob_rec_z такого объекта:

$$NewOb_z = \langle Cl_z = Cl(Ob_rec_z), R_z, F_z = \emptyset, Pos_z = Pos(Ob_rec_z), ID_z, M_z = 0, ObSt_z = 0 \rangle,$$

где R_z – оценка линейного размера данного объекта, полученная в результате применения подхода к определению пространственного положения и размеров объектов сцены по изображению. Определение формы объекта F_z осуществляется с использованием сервисных РС, потому на данном этапе форма объекта инициализируется пустым множеством. ID_z – сгенерированный уникальный идентификатор для данного объекта, присваиваемый в момент инициализации данного объекта.

Все представления взаимодействий $nUIntSet_z$, ассоциированные с некоторым Ob_rec_z ($Ob_rec_z \rightarrow NewOb_z$), также являются однозначно ассоциированным с некоторым конкретным пользователем $UDet_z$ из набора $UDet$, задетектированным на территории КФП, где $UDet_z \rightarrow U_z, U_z \in U$. Таким образом, обладая информацией о взаимодействующем пользователе U_z и об объекте взаимодействия $NewOb_z$, становится возможным инициализировать соответствующий экземпляр взаимодействия $NewInt_z$ со следующими параметрами:

$$NewInt_z = \langle St_z=1, T_z=0, Type_z=1, ID_z = ID(NewOb_z), ME_z=ID(UDet_z), Pos_start_z=Pos(Ob_rec_z), Pos_fin_z=Pos(Ob_rec_z) \rangle,$$

где $Ob_rec_z \rightarrow NewOb_z$. Далее в отношении данного взаимодействия необходимо инициализировать соответствующий набор процессов трекинга $TrackSet_z$:

$$TrackSet_z = \emptyset, TrackSet_z \rightarrow NewInt_z.$$

Набор новых объектов, сформированных на данном шаге, обозначим как $NewOb$, а набор инициализированных взаимодействий с такими объектами – как $NewInt$.

21. На следующем шаге в отношении каждого представления взаимодействия $nUIntSet_z$ из набора $nUIntSet$, ассоциированного с

некоторым объектом $NewOb_z$ и взаимодействием $NewInt_z$, необходимо инициализировать соответствующий процесс трекинга $NewTrack_r$ такого взаимодействия $NewInt_z$ со следующими параметрами:

$$NewTrack_r = \langle ID_z, U_ID_z, Box(Ob)^t = Box(Ob)_z, Box(Ob)^{t+\Delta t}, \\ Box(U)^t = Box(U)_z, Box(U)^{t+\Delta t} \rangle,$$

где $ID_z, U_ID_z \in NewInt_z$; $Box(Ob)_l, Box(U)_l \in nUIntSet_z$; $Box(Ob)^{t+\Delta t}, Box(U)^{t+\Delta t}$ – соответствующие значения будут определены в рамках исполнения соответствующего процесса трекинга взаимодействия.

После того, как процесс трекинга $NewTrack_r$ был инициирован, необходимо обновить набор актуальных процессов трекинга $TrackSet_z$, связанный с данным взаимодействием $NewInt_z$:

$$TrackSet_z = TrackSet_z \cup \{NewTrack_r\}, NewInt_z \rightarrow TrackSet_z.$$

По завершению обработки всех представлений взаимодействий из набора $nUIntSet$ выполняется добавление новых инициированных объектов и взаимодействий из наборов $NewOb$ и $NewInt$ соответственно к набору зарегистрированных объектов Ob и набору текущих взаимодействий Int :

$$Ob = Ob \cup NewOb; \\ Int = Int \cup NewInt.$$

По результатам выполнения данного шага, завершается обработка всех новых задетектированных представлений взаимодействий пользователей с объектами из набора $NewUIntSet$.

22. Далее в отношении каждого взаимодействия между пользователем и объектом Int_j из обновленного набора Int осуществляется проверка на факт завершения такого взаимодействия. В случае, если для некоторого взаимодействия Int_j и, в частности, ассоциированных с ним процессов трекинга $Track_{jc}$ справедливо:

$$\left\{ \begin{array}{l} \nexists Track_{jc}: (Track_{jc} \rightarrow UIntSet) \neq \emptyset \mid \forall Track_{jc} \in TrackSet_j, Int_j \rightarrow TrackSet_j, \\ Type(Int_j) == 0 \end{array} \right., \quad (47)$$

то такое взаимодействие Int_j считается завершенным. Набор взаимодействий Int_j ($Int_j \in Int$) для которых справедливо выражение (47) обозначим как $FUInt$.

23. Далее в отношении каждого взаимодействия Int_f из набора $FUInt$ осуществляется сопоставление ассоциированных с данным взаимодействием процессов трекинга взаимодействий $Track_{fc}$ ($c = \overline{1, C_f}$) с представлениями распознанных объектов из набора Ob_rec . Поскольку каждый процесс трекинга однозначно ассоциирован с некоторым $Mean_c$, также как и ограничивающие прямоугольники Box_{hs} представления объекта Ob_rec_h , то сопоставление взаимодействия Int_f с представлением объекта Ob_rec_h может быть выполнено на основе следующего выражения:

$$\exists Ob_rec_h: \begin{cases} IoU(Box(Ob)_{fc}^t, Box_{hs}) \geq \alpha \mid s \in (1, \dots, S_h), c \in (1, \dots, C_f) \\ Box_{hs} \in Ob_rec_h, Box(Ob)_{fc}^t \in Track_{fc} \\ Mean_s == Mean_c \mid Box_{hs} \rightarrow Mean_s, Track_{fc} \rightarrow Mean_c \end{cases}. \quad (48)$$

Если представленное выше выражение справедливо для какого-либо представления объекта $Ob_rec_h \in Ob_rec$, то данное представление считается связанным с завершившимся взаимодействием Int_f . Далее, обновленные данные об объекте взаимодействия, содержащиеся в представлении Ob_rec_h , используются для актуализации значений параметров взаимодействия Int_f и параметров соответствующего объекта взаимодействия Ob_f из набора Ob :

$$\begin{aligned} Pos(Ob_f) &= Pos(Ob_rec_h); \\ Pos_fin(Int_f) &= Pos(Ob_rec_h); \\ T(Int_f) &= T(Int_f) + \Delta t; \\ St(Int_f) &= 0; \\ ObSt(Ob_f) &= 0. \end{aligned}$$

Если условия выражения (10) для случая взаимодействия Int_f не выполняются ни для какого-либо представления объекта из набора Ob_rec . То в отношении такого взаимодействия инициализируется проверка наличия пользователя, ассоциированного с данным взаимодействием, на территории КФП. Проверка осуществляется на основе следующего выражения:

$$\exists UDet_d: U_ID_d == ME(Int_f) \mid U_ID_d \in UDet_d, d \in (1, \dots, D). \quad (49)$$

Если представленное выше выражение несправедливо для Int_f , то делается вывод, что в рамках взаимодействия Int_f , соответствующий объект Ob_f был вынесен пользователем U_f за пределы территории КФП. Данные о взаимодействии обновляются, а объект Ob_f добавляется к набору $ObOut$, который включает в себя все объекты, которые в рамках текущего цикла исполнения модели покинули территорию КФП:

$$\begin{aligned} ObOut &= ObOut \cup \{Ob_f\}; \\ T(Int_f) &= T(Int_f) + \Delta t; \\ St(Int_f) &= 0. \end{aligned}$$

Если же для Int_f выражение (11) оказывается справедливым, то взаимодействие Int_f также считается завершенным, а объект взаимодействия Ob_f признается нераспознанным в рамках текущего цикла исполнения модели, после чего производится актуализация соответствующей информации о взаимодействии и об объекте:

$$\begin{aligned} ObSt(Ob_f) &= ObSt(Ob_f) + 1; \\ T(Int_f) &= T(Int_f) + \Delta t; \\ St(Int_f) &= 0. \end{aligned}$$

По результатам выполнения представленных выше шагов, обеспечивается анализ всех задетектированных средствами КФП

представлений взаимодействий пользователей с объектами из набора $UIntSet$.

24. Далее в рамках текущего цикла исполнения модели запрашиваются данные о статусе взаимодействия сервисных РС с объектами, зарегистрированными на территории КФП. Каждое РС RS_b , функционирующее на территории КФП, передает актуальные данные в отношении осуществляемых взаимодействий в следующем виде:

$$RSInt_b = \langle RS_ID_b, ID(Ob)_b, RS_Pos_b, Ob_Pos_b, State_b \rangle,$$

где RS_ID_b – уникальный идентификатор сервисного РС RS_b ; $ID(Ob)_b$ – уникальный идентификатор объекта взаимодействия Ob_b ($Ob_b \in Ob$); RS_Pos_b – пространственное положение сервисного РС; Ob_Pos_b – пространственное положение объекта взаимодействия Ob_b ; $State_b$ – состояние взаимодействия, $State_b$ представляет собой идентификатор, характеризующий состояние взаимодействия РС с объектом, $State_b = 0$ в случае, если взаимодействие инициировано, $State_b = 1$, если взаимодействие уже осуществлялось на предыдущем цикле исполнения модели, $State_b = 2$, если взаимодействие завершено. Набор представлений взаимодействий вида $RSInt_b$ обозначим как $RSInt$. Важно отметить, что сервисные РС функционируют исключительно на территории КФП и не могут участвовать во взаимодействиях, направленных на внесение или вынос объектов с территории КФП.

25. На следующем шаге в рамках настоящей модели в отношении каждого представления взаимодействия РС с объектом $RSInt_b$ осуществляется сопоставление такого представления с текущими взаимодействиями из набора Int . Поскольку каждое взаимодействие может быть однозначно идентифицировано на основе идентификатора взаимодействующего субъекта и идентификатора объекта взаимодействия, то сопоставления представлений $RSInt_b$ с взаимодействиями из набора Int осуществляется в соответствии со следующим выражением:

$$RSInt_b \rightarrow Int_e, \text{ если: } \begin{cases} RSID_b == ME(Int_e) \\ ID(Ob)_b == ID(Ob)_e \end{cases} \quad (50)$$

Объект взаимодействия Ob_e , ассоциированный с представлением $RSInt_b$, в свою очередь может быть однозначно определен по $ID(Ob)_b$ ($ID(Ob)_b \in RSInt_b$).

Далее осуществляется актуализация значений параметров каждого взаимодействия Int_e , взаимодействующего РС и параметров каждого соответствующего объекта взаимодействия Ob_e , ассоциированных с некоторым представлением $RSInt_b$ из набора $RSInt$.

В случае, если $State(RSInt_b) = 1$:

$$\begin{aligned} Pos(Ob_e) &= Ob_Pos_b; \\ Pos_fin(Int_e) &= Pos(Ob_Pos_b); \\ T(Int_e) &= T(Int_e) + \Delta t; \\ ObSt(Ob_e) &= 0; \\ Pos(RS_b) &= RS_Pos_b. \end{aligned}$$

В случае, если $State(RSInt_b) = 2$:

$$\begin{aligned} Pos(Ob_e) &= Ob_Pos_b; \\ Pos_fin(Int_e) &= Pos(Ob_Pos_b); \\ T(Int_e) &= T(Int_e) + \Delta t; \\ St(Int_e) &= 0; \\ ObSt(Ob_e) &= 0; \\ Pos(RS_b) &= RS_Pos_b. \end{aligned}$$

В случае, если $State(RSInt_b) = 0$, то условие (50) для такого представления не выполняется, данное представление взаимодействия $RSInt_b$ характеризует новое инициированное взаимодействие $NewRSInt_e$. В таком случае необходимо инициализировать соответствующий экземпляр взаимодействия $NewRSInt_e$ со следующими значениями параметров:

$$\begin{aligned} NewRSInt_e &= \langle St_e=1, T_e=0, Type_e=0, ID_e = ID(Ob_e), ME_e=RS_ID_b, \\ &Pos_start_e=Ob_Pos_b, Pos_fin_e= Ob_Pos_b \rangle, \end{aligned}$$

где $RSInt_b \rightarrow NewRSInt_e$.

После того, как соответствующее взаимодействие было инициализировано, требуется также актуализировать параметры РС RS_b и объекта взаимодействия Ob_e :

$$\begin{aligned} Pos(Ob_e) &= Ob_Pos_b; \\ ObSt(Ob_e) &= 0; \\ Pos(RS_b) &= RS_Pos_b. \end{aligned}$$

Таким образом, обеспечивается актуализация информации, связанной с взаимодействиями сервисных РС с объектами на территории КФП. Набор вновь инициированных взаимодействий, включающий в себя все взаимодействия вида $NewRSInt_e$, обозначим как $NewRSInt$. Данный набор подлежит объединению с набором текущих взаимодействий Int :

$$Int = Int \cup NewRSInt.$$

По результатам выполнения данного шага, завершается обработка всей полученной от сервисных РС информации в отношении взаимодействий РС с объектами, отраженной в наборе представлений взаимодействий $RSInt$.

26. В рамках настоящей модели, на основе алгоритма оценивания пространственных характеристик объектов с использованием сервисных РС обеспечивается уточнение линейных размеров объектов, а также реализуется оценка форм объектов, зарегистрированных на территории КФП. Таким образом, на следующем шаге данной модели запрашиваются данные о результатах оценки пространственных характеристик объектов. Каждое РС RS_g , завершившее выполнение такой оценки с момента предыдущего цикла исполнения модели, передает соответствующие данные следующего виде:

$$RSEst_g = \langle RS_ID_g, ID(Ob)_g, RS_Pos_b, Ob_Pos_g, R(Ob)_g, F(Ob)_g \rangle,$$

где RS_ID_g уникальный идентификатор сервисного РС RS_g ; $ID(Ob)_g$ – уникальный идентификатор оцениваемого объекта Ob_g ($Ob_g \in Ob$); RS_Pos_g – пространственное положение сервисного РС; Ob_Pos_g – пространственное положение объекта Ob_g ; $R(Ob)_g$ – оценка линейного размера объекта Ob_g ; $F(Ob)_g$ – оценка формы объекта Ob_g , представленная набором точек в трехмерном пространстве, ассоциированных с поверхностью объекта.

Далее на основе полученных данных осуществляется актуализация значений параметров каждого оцениваемого объекта Ob_g , а также РС RS_g , проводившей соответствующую оценку:

$$\begin{aligned} R(Ob_g) &= R(Ob)_g; \\ F(Ob_g) &= F(Ob)_g; \\ Pos(Ob_g) &= Ob_Pos_g; \\ M(Ob_g) &= 1; \\ ObSt(Ob_g) &= 0; \\ Pos(RS_g) &= RS_Pos_g. \end{aligned}$$

По завершению данного шага, достигается актуальность данных в отношении всех распознанных объектов из набора Ob на территории КФП в рамках текущего цикла исполнения модели.

27. Далее осуществляется выделение подмножества нераспознанных в рамках текущего цикла исполнения модели объектов из числа зарегистрированных на территории КФП. К данному подмножеству отнесем лишь такие объекты Ob_o , которые на текущий момент времени, с одной стороны, не ассоциированы ни с одним существующим взаимодействием Int_o из набора Int независимо от статуса такого взаимодействия, а с другой, не связаны ни с одним представлением распознанного на территории КФП объекта Ob_rec_k ($Ob_rec_k \in Ob_rec$). Таким образом, объект Ob_o относится к соответствующему набору $MissOb$ в случае, если выполняется следующий набор условий:

$$\begin{cases} Ob_o \rightarrow Int = \emptyset \\ \nexists Ob_rec_k \in Ob_rec: \left\{ \begin{aligned} Cl(Ob_rec_k) &= Cl(Ob_o) \\ |Pos(Ob_rec_k) - Pos(Ob_o)| &< \varepsilon(Cl(Ob_o)) \end{aligned} \right\} | k = (1, \dots, K) \end{cases}$$

В отношении всех объектов Ob_o , для которых справедливо представленное выше условие, осуществляется следующее изменение значений параметров:

$$ObSt(Ob_o) = ObSt(Ob_o) + 1.$$

28. Следующий шаг настоящей модели посвящен актуализации набора зарегистрированных на территории КФП объектов Ob , а также набора текущих взаимодействий Int . На данном этапе, все значения параметров отдельных объектов Ob_i и взаимодействий Int_j являются обновленными и отражают текущее состояние КФП. Однако, в соответствующих наборах по-прежнему присутствуют как уже завершенные взаимодействия, так и объекты, вынесенные с территории КФП.

Согласно настоящей модели, актуализированный набор зарегистрированных объектов $FinOb$ определяется следующим образом:

$$FinOb = \{Ob_i | Ob_i \notin ObOut, ObSt(Ob_i) \leq \lambda, Ob_i \in Ob\},$$

где $ObOut$ – набор объектов, к настоящему моменту вынесенных с территории КФП; λ – критическое число последовательных циклов исполнения модели, в рамках которых объект Ob_i не был задетектирован системой, значение данного параметра задается в рамках первичной настройки модели.

После того, как актуализированный набор зарегистрированных объектов $FinOb$ был определен, осуществляется переопределение набора Ob :

$$Ob = FinOb.$$

Актуализированный набор взаимодействий $FinInt$ в свою очередь определяется следующим образом:

$$FinInt = \{Int_j | St(Int_j) \neq 0, Int_j \in Int\}.$$

После того, как актуализированный набор текущих взаимодействий на территории КФП $FinInt$ был определен, осуществляется переопределение набора Int :

$$Int = FinInt.$$

29. На заключительном шаге настоящей модели, в отношении всех объектов Ob_i из актуального набора Ob , значение параметров которых удовлетворяет следующему условию:

$$M(Ob_i) == 0,$$

формируются задачи по оценке пространственных характеристик таких объектов с использованием сервисных РС. Управление и контроль над набором таких задач реализуется соответствующей системой распределения задач между сервисными РС.

Таким образом, предложенная алгоритмическая модель мониторинга и управления процессами манипуляций над объектами на территории КФП, используя полный спектр разработанных в рамках данного проекта решений, обеспечивает полный автоматизированный мониторинг положения и состояния различного рода объектов на территории КФП, позволяет отслеживать факты появления внесения новых объектов, а также изъятия уже зарегистрированных объектов с территории КФП. Предложенная модель также контролирует все процесса взаимодействия с объектами, осуществляемые как со стороны пользователей КФП, так и компонентов КФП, представленных сервисными РС.

Литература

1. Krestovnikov K., Saveliev A., Cherskikh E. Development of a circuit design for a capacitive pressure sensor, applied in walking robot foot // 2020 IEEE 20th Mediterranean Electrotechnical Conference (MELECON). IEEE, 2020. С. 243-247. DOI: 10.1109/MELECON48756.2020.9140509
2. Ерашов А.А., Крестовников К.С., Быков А.Н. Разработка схемотехнического решения и конструкции емкостной матрицы датчиков давления для применения в робототехнике // Робототехника и техническая кибернетика. №4. 2020 (Принято к печати)
3. Krestovnikov K., Cherskikh E., Zimuldinov E. Combined Capacitive Pressure and Proximity Sensor for Using in Robotic Systems // Proceedings of 15th International Conference on Electromechanics and Robotics “Zavalishin’s Readings” (ER(ZR) 2020). – Springer, Singapore, 2021. – P. 513-522. https://doi.org/10.1007/978-981-15-5580-0_42
4. Павлюк Н. А., Бизин М. М. Конструктивные решения для антропоморфного робота АНТАРЕС // Математические методы в технике и технологиях-ММТТ. – 2016. – №. 9. – С. 138-141.
5. Kodyakov A. S. et al. Stability study of anthropomorphic robot antares under external load action // Journal of Physics: Conference Series. – IOP Publishing, 2017. – Т. 803. – №. 1. – С. 012074.
6. Rubtsova J., Iakovlev R. Comparative Analysis of Approaches to Depth Map Generation for Robot Navigation // International Conference on Interactive Collaborative Robotics. – Springer, Cham, 2020. – С. 265-272. https://doi.org/10.1007/978-3-030-60337-3_26
7. Яковлев Р.Н., Рубцова Ю.И., Ерашов А.А. Сравнительная оценка подходов к определению точек захвата объектов робототехническим средством // Мехатроника. Автоматизация. Управление. №2. 2021 (Принято к печати)
8. Kuffner J. J., LaValle S. M. RRT-connect: An efficient approach to single-query path planning // Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065). – IEEE, 2000. – Т. 2. – С. 995-1001.
9. Pavluk N., Ivin A., Budkov V., Kodyakov A., Ronzhin A. Mechanical leg design of the anthropomorphic robot Antares // International Conference on Interactive Collaborative Robotics. – Springer, Cham, 2016. – С. 113-123.
10. N. Pavluk, A. Denisov, A. Kodyakov, and A. Ronzhin, “Mechanical Engineering of Leg Joints of Anthropomorphic Robot,” In MATEC Web of Conferences, vol. 77, p. 04006, EDP Sciences, 2016
11. Keselman L. et al. Intel realsense stereoscopic depth cameras // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. – 2017. – С. 1-10
12. Devernay F., Faugeras O. D. Automatic calibration and removal of distortion from scenes of structured environments // Investigative and Trial Image

Processing. – International Society for Optics and Photonics, 1995. – Т. 2567. – С. 62-72

13. He K., Gkioxari G., Dollár P., Girshick R. Mask r-cnn //Proceedings of the IEEE international conference on computer vision. – 2017. – С. 2961-2969.

14. Ammirato P., Berg A. C. A Mask-RCNN Baseline for Probabilistic Object Detection //arXiv preprint arXiv:1908.03621. – 2019.

15. ГОСТ Р 55710-2013 Освещение рабочих мест внутри зданий. Нормы и методы измерений: утвержден и введен в действие Приказом Федерального агентства по техническому регулированию и метрологии от 8 ноября 2013 г. N 1364-ст. – URL: <http://docs.cntd.ru/document/1200105707> (дата обращения: 27.04.2020). –Текст: электронный.

16. Krüsi P., Furgale P., Bosse M., Siegwart R. Driving on point clouds: Motion planning, trajectory optimization, and terrain assessment in generic nonplanar environments //Journal of Field Robotics. – 2017. – Т. 34. – №. 5. – С. 940-984.

17. Engel J., Schöps T., Cremers D. LSD-SLAM: Large-scale direct monocular SLAM //European conference on computer vision. – Springer, Cham, 2014. – С. 834-849.

18. Schubert E. et al. DBSCAN revisited, revisited: why and how you should (still) use DBSCAN //ACM Transactions on Database Systems (TODS). – 2017. – Т. 42. – №. 3. – С. 1-21.

19. Dai J., He K., Sun J. Instance-aware semantic segmentation via multi-task network cascades //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. – 2016. – С. 3150-3158.

20. Shrivastava A. et al. Beyond skip connections: Top-down modulation for object detection //arXiv preprint arXiv:1612.06851. – 2016.

21. Krinski B. A. et al. Masking salient object detection, a mask region-based convo-lu-tional neural network analysis for segmentation of salient objects //arXiv pre-print arXiv:1909.08038. – 2019.

22. Gkioxari G., Girshick R., Dollár P., He K. Detecting and recognizing human-object interactions //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. – 2018. – С. 8359-8367.

23. Shen L., Yeung S., Hoffman J., Mori G., Fei-Fei L. Scaling human-object interaction recognition through zero-shot learning //2018 IEEE Winter Conference on Applications of Computer Vision (WACV). – IEEE, 2018. – С. 1568-1576.

24. Cao Z., Hidalgo G., Simon T., Wei S.E., Sheikh Y. OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields //IEEE transactions on pattern analysis and machine intelligence. – 2019.

25. Gkioxari G., Girshick R., Dollár P., He K. Detecting and recognizing human-object interactions //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. – 2018. – С. 8359-8367.

26. Rogez G., Supancic J. S., Ramanan D. First-person pose recognition using egocentric workspaces //Proceedings of the IEEE conference on computer vision and pattern recognition. – 2015. – C. 4325-4333.
27. Rogez G., Khademi M., Supancic III, J., Montiel J.M.M., Ramanan D. 3D hand pose detection in egocentric RGB-D images //European Conference on Computer Vision. – Springer, Cham, 2014. – C. 356-371.
28. Keskin C., Kırac F., Kara Y.E., Akarun L. Hand pose estimation and hand shape classification using multi-layered randomized decision forests //European Conference on Computer Vision. – Springer, Berlin, Heidelberg, 2012. – C. 852-863.
29. Oberweger M., Wohlhart P., Lepetit V. Hands deep in deep learning for hand pose estimation //arXiv preprint arXiv:1502.06807. – 2015.
30. Garcia-Hernando G. et al. First-person hand action benchmark with rgb-d videos and 3d hand pose annotations //Proceedings of the IEEE conference on computer vision and pattern recognition. – 2018. – C. 409-419.
31. Moon G., Yong Chang J., Mu Lee K. V2v-posenet: Voxel-to-voxel prediction network for accurate 3d hand and human pose estimation from a single depth map //Proceedings of the IEEE conference on computer vision and pattern Recognition. – 2018. – C. 5079-5088.
32. Kölsch M., Turk M. Robust Hand Detection //FGR. – 2004. – C. 614-619.
33. Redmon J., Farhadi A. Yolov3: An incremental improvement //arXiv preprint arXiv:1804.02767. – 2018.
34. Murawski K., Murawska M., Pustelny T. Optimizing the light source design for a sensor to measure the stroke volume of the artificial heart //13th Conference on Integrated Optics: Sen-sors, Sensing Structures, and Methods. – International Society for Optics and Photonics, 2018. – T. 10830. – C. 1083006.
35. Karsch K., Liu C., Kang S. B. Depth extraction from video using non-parametric sampling //European Conference on Computer Vision. – Springer, Berlin, Heidelberg, 2012. – C. 775-788.
36. Eigen D., Fergus R. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture //Proceedings of the IEEE international conference on computer vision. – 2015. – C. 2650-2658.
37. Laina I., Rupprecht C., Belagiannis V., Tombari F., Navab N.: Deeper depth prediction with fully convolutional residual networks //2016 Fourth international conference on 3D vision (3DV). – IEEE, 2016. – C. 239-248.
38. He K. et al. Deep residual learning for image recognition //Proceedings of the IEEE confer-ence on computer vision and pattern recognition. – 2016. – C. 770-778.
39. Cheong Y. Z., Chew W. J. The Application of Image Processing to Solve Occlusion Issue in Object Tracking //MATEC Web of Conferences. – EDP Sciences, 2018. – T. 152. – C. 03001.

40. Ning G. et al. Spatially supervised recurrent convolutional neural networks for visual object tracking //2017 IEEE International Symposium on Circuits and Systems (IS-CAS). – IEEE, 2017. – C. 1-4.
41. Feng Q. et al. Tell Me What to Track //arXiv preprint arXiv:1907.11751. – 2019.
42. Zhang D. et al. Deep reinforcement learning for visual object tracking in videos //arXiv pre-print arXiv:1701.08936. – 2017.
43. Redmon J. et al. You only look once: Unified, real-time object detection //Proceedings of the IEEE conference on computer vision and pattern recognition. – 2016. – C. 779-788.
44. Schroff F., Kalenichenko D., Philbin J. Facenet: A unified embedding for face recognition and clustering //Proceedings of the IEEE conference on computer vision and pattern recognition. – 2015. – C. 815-823.
45. Salychev O. S. Verified approaches to inertial navigation //Moscow, BMSTU Publ. – 2017.