

Министерство образования и науки Российской Федерации  
САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ  
ПЕТРА ВЕЛИКОГО

Институт металлургии, машиностроения и транспорта  
Кафедра «Мехатроника и роботостроение» при ЦНИИ РТК

ОТЧЕТ

по научно-исследовательской работе

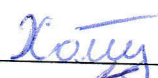

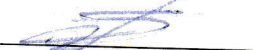
Исследование методов определения положения трехмерных слабо  
текстурированных объектов с использованием одной телевизионной камеры

Направление подготовки: Мехатроника и роботостроение

Выполнил студент гр. 43328/1

Научный руководитель

Научный консультант

|  |                |
|--|----------------|
|  | Е. Э. Хомутов  |
|  | А. В. Бахшиев  |
|  | Д. Н. Степанов |

Санкт-Петербург 2018

Министерство образования и науки Российской Федерации  
САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ  
ПЕТРА ВЕЛИКОГО

Институт металлургии, машиностроения и транспорта  
Кафедра «Мехатроника и роботостроение при ЦНИИ РТК»

ЗАДАНИЕ

на выполнение научно-исследовательской работы  
Хомутову Евгению Эдуардовичу

студенту гр. 43328/1

1 Тема работы

Исследование методов определения положения трехмерных  
слабо текстурированных объектов с использованием одной телевизионной  
камеры

2 Срок сдачи студентом законченной работы

3 Исходные данные к работе

4 Содержание расчетно-пояснительной записки

Введение, обзор существующих технических решений, выбранные технические  
решения, заключение

5 Перечень графического материала (с точным указанием обязательных  
чертежей)

6 Консультанты по работе

Степанов Д. Н. начальник лаборатории СТЗ, ЦНИИ РТК

7 Дата выдачи задания

10.03.18

Руководитель Бахшиев А. В. – доцент кафедры «Мехатроника и  
роботостроение»



(ФИО, должность, подпись руководителя)

Задание принял к исполнению

10.09.18

(дата)

Хотун

(подпись студента)

## РЕФЕРАТ

Отчет 49 с., 10 рис., 3 табл., 1 приложений, 55 источников

Ключевые слова: определение положения объекта, положение объекта в пространстве, слабо текстурированный объект, не текстурированный объект.

Объект исследования – процесс определения положения слабо текстурированного объекта.

Цель работы - исследование и выбор методов определения положения не текстурированных объектов по изображению.

Метод проведения работы – изучение литературы и других источников информации.

Область применения: 3D – реконструкция объектов, построение 3D – моделей объектов, осуществление локализации и навигации роботов, захват объектов манипулятором, дополненная реальность, виртуальная реальность.

## СОДЕРЖАНИЕ

|   |    |
|---|----|
| ВВЕДЕНИЕ.....   | 7  |
| 1 Глава 1: Обзор существующих технических решений.....      | 8  |
| 1.1 Детектирование объекта на изображении.....              | 8  |
| 1.1.1 Цветовая фильтрация .....                             | 9  |
| 1.1.2 Контурный анализ .....                                | 9  |
| 1.1.3 Сопоставление с шаблоном.....                         | 10 |
| 1.1.4 Особые точки .....                                    | 11 |
| 1.1.5 Алгоритмы сегментации.....                            | 13 |
| 1.1.5.1 MeanShift .....                                     | 13 |
| 1.1.5.2 Метод Оцу .....                                     | 13 |
| 1.1.6 Машинное обучение .....                               | 15 |
| 1.1.7 Идентификация заданного объекта среди найденных ..... | 16 |
| 1.2 Определение положения объекта .....                     | 18 |
| 1.2.1 PnP, PnL.....   | 18 |
| 1.2.2 Определение положения объекта по его модели .....     | 21 |
| 1.2.3 Машинное обучение (иерархический поиск).....          | 23 |
| 1.2.4 Определение положения на основе RGB-D.....            | 23 |
| 1.2.5 Intel RealSense.....                                  | 24 |
| 1.2.5.1 Времяпролетная камера .....                         | 24 |
| 1.2.5.2 Стереокамера .....                                  | 26 |
| 1.2.5.3 Смещение камеры .....                               | 27 |
| 1.2.6 Нейронные сети .....                                  | 27 |
| 2 Глава 2: Выбранные технические решения .....              | 29 |

|                                     |  |    |
|-------------------------------------|--|----|
| 2.1                                 | Цветовая фильтрация .....                            | 29 |
| 2.2                                 | Машинное обучение.....                               | 32 |
| 2.2.1                               | Генерация эталонных изображений модели объекта ..... | 35 |
| 2.2.2                               | Калибровка камеры .....                              | 35 |
| 2.2.3                               | Масштабирование изображений.....                     | 36 |
| 2.2.4                               | Положение объекта в пространстве .....               | 36 |
| 2.3                                 | Определение положения на основе нейронных сетей..... | 37 |
| ЗАКЛЮЧЕНИЕ .....                    |  | 44 |
| СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ..... |  | 45 |

## ВВЕДЕНИЕ

В настоящее время задача распознавания объектов и определения их положения в пространстве по изображению имеет множество практических применений в задачах робототехники. В общем смысле, задача определения положения – это задача определения относительного положения и ориентации камеры и объекта друг относительно друга в трехмерном пространстве. Она применима к таким задачам как: 3D – реконструкция объектов, построение 3D – моделей объектов, осуществление локализации и навигации роботов, захват объектов манипулятором, а также дополненная реальность – лишь некоторые из них. Задача может быть сформулирована либо как определение положения объекта (относительно камеры) или положения камеры (относительно объекта или в мировой системе координат). В данной работе используется первая формулировка.

К настоящему времени, в литературе рассмотрены алгоритмы для определения положения объектов в пространстве по изображению на основе контурного анализа [1, 2], цветовой фильтрации [3], а также нейронных сетей [4, 5]. Проблема заключается в том, что не существует универсальных методов, которые могли бы быть применимы ко всем объектам. Тем не менее, проблема является сложной из-за разнообразия объектов в реальном мире. Объекты имеют разную трехмерную форму, геометрию, и их появление на изображениях зависит от условий освещения, помех и окклюзий сцены. В данной работе рассматривается проблема определения положения слабо текстурированных объектов, т.е. трехмерных объектов, не имеющие ярко выраженных особенностей, не закрытых другими объектами сцены. К таким объектам можно отнести: объекты, однородные по цвету, например, детали на конвейере; объекты, обладающие осью симметрии, так как при наличии оси симметрии нескольким различным пространственным положениям объекта может соответствовать одно визуальное отображение; объекты с отражающей или прозрачной поверхностью.

## 1 Глава 1: Обзор существующих технических решений

Для определения положения объекта в пространстве, в общем случае, необходимо решить две задачи:

- Детектирование объекта на изображении;
- Определение положения детектированных объектов.

В данной главе рассмотрены существующие методы решения каждой задачи в отдельности.

### 1.1 Детектирование объекта на изображении

Для детектирования объектов на изображении существуют различные методы. Некоторые из них рассматривают внешний вид, в частности, цвет объекта [6-8], форму [9] или контур [7] объекта. Другие основаны на сопоставлении локальных особенностей, описываемых дескрипторами, например, такими как HOG (Histogram of Oriented Gradients) [10, 11], SIFT (Scale-Invariant Feature Transform) [12]. Также существуют подходы основе машинного обучения [13-15].

В общем случае, для обнаружения объекта на изображении необходима информация о внешнем виде объекта.

Задача обнаружения объектов заключается в установлении наличия на изображении объекта, обладающего некоторыми определенными характеристиками или свойствами.

Возможные подходы решения данной задачи:

- Цветовая фильтрация;
- Контурный анализ;
- Сопоставление с шаблоном;
- Особые точки;
- Алгоритмы сегментации;
- Методы машинного обучения.



### 1.1.1 Цветовая фильтрация

Такой характеристикой может быть, например, тон цвета. Одним из наиболее простых и естественных способов обнаружения объекта (или объектов) является выбор порога по цвету, или пороговая фильтрация (thresholding).

Для цветовой фильтрации достаточно часто строятся цветовые гистограммы, по которым определяется и подбирается пороговое значение [3].

Цветовая фильтрация отлично подойдет для фильтрации объектов на изображении, которые существенным образом отличаются от остальной части изображения (фона), и однородных по цвету.

Метод хорошо подойдет для решения данной задачи, так как цвет выбранного объекта известен, объект однороден по цвету и не закрыт другими объектами сцены.

### 1.1.2 Контурный анализ

Для объектов, существенным образом не выделяющихся на изображении или имеющих сложную окраску, метод цветовой фильтрации не даст хороших результатов. Для таких объектов можно попробовать применить контурный анализ.

Контурный анализ - это метод распознавания и поиска объектов по их контурам на изображении. Контур – это непрерывная кривая, соединяющая все точки вдоль границы, имеющих одинаковый цвет или интенсивность, и описывающая границы объекта на изображении. Граница объекта - это места резкого изменения градиента яркости изображения. Для поиска контуров наиболее распространенными являются алгоритмы Собеля и Кэнни [7]. Рассмотрение и анализ только контуров на изображении значительно снижает сложность алгоритмов и вычислений.

Главным достоинством контурного анализа является инвариантность к вращениям, масштабированию и смещению контура на анализируемом

изображении. Он отлично подходит для поиска объектов несложной заданной формы.

Однако на применение данного метода накладываются существенные ограничения, которые в основном связаны с проблемами выделения контуров на изображении:

- Сложность выделения четких границ объекта при подобности цветов объекта и фона;
- Окклюзия и группировка объектов приводит к выделению контура, не соответствующему границе объекта.

Таким образом, контурный анализ слабо устойчив к окклюзиям сцены, так как даже незначительное нарушение целостности контура объекта не позволяет детектировать четкий контур, а также приводит к ложным срабатываниям. Тем не менее, контурный анализ прост и быстр, что позволяет применять его к объектам, существенно отличающимся от фона [8, 9].

Для решения рассматриваемой проблемы контурный анализ подойдет, но его применение усложняется тем, что, если объект может иметь сложную геометрическую форму, которая может проецироваться на плоскость изображения камеры неоднозначно, вследствие чего получить набор шаблонов контуров объекта для получения хороших результатов является достаточно трудоемкой задачей.

### 1.1.3 Сопоставление с шаблоном

Если сцена содержит множество мелких деталей или объект имеет сложную окраску, то цветовая фильтрация и контурный анализ неприменимы. Тогда может быть применен метод, основанный на сопоставлении с шаблоном.

Данный метод применяется для поиска участков изображений, которые наиболее схожи с некоторым заданным шаблоном.

Сопоставление с шаблоном выполняется путем сравнения областей изображения с шаблоном и оценкой подобия области с шаблоном. По результатам сравнения выбирается наиболее схожая область изображения шаблону. То есть метод ищет точные совпадения пикселей шаблона с пикселями на изображении.

Сопоставление с шаблоном является хорошим выбором, когда необходимо быстро проверить наличие некоторого объекта на изображении. Такой подход значительно проще реализовать, нежели при помощи обучаемых алгоритмов.

Точность детектирования заданного объекта с помощью контурного анализа зависит от многих факторов, таких как масштаб изображения, угол съемки, поворот изображения и степень окклюзии. Также возможно ошибочное обнаружение объекта по контуру при наличии схожих частей на шаблоне и изображении. Одним из методов решения данной проблемы является введение некоторого коэффициента схожести [8, 9].

Также метод имеет существенное ограничение: он не устойчив к поворотам и масштабированию изображения относительно параметров шаблона.

Для данной задачи метод сопоставления с шаблоном скорее всего неприменим, так как заранее неизвестно расстояние между объектом и камерой для возможности масштабирования шаблона.

#### 1.1.4 Особые точки

Для преодоления ограничений, связанных с поворотом и масштабированием изображения, применяют методы, основанные на особых точках. Такие методы нацелены на выделение ключевых особенностей на изображении. Данные особенности затем используются для сравнения двух изображений с целью выявления на них общих составляющих.

Особые точки – это точки, которые существенным образом выделяются на изображении, т.е. такие, где резко меняется цвет, яркость изображения. Такими точками могут быть изолированные точки, кривые, углы, грани объектов или некоторые связанные области. Существует множество методов обнаружения особых точек, некоторые из них хорошо работают только для определенного вида особых точек. Так для поиска углов пойдет детектор Харриса [16], для поиска небольших связных областей подойдет метод Blob detection [17].

Каждый алгоритм определяет особые точки по-своему. В целом, алгоритмы на основе особых точек включают три составляющие:

- Детектор — обнаружение особых точек на изображении;
- Детектор — описание обнаруженных особых точек (область вокруг них);
- Матчер — сопоставление двух наборов особых точек изображений (шаблонного и реального).

Методы на основе особых точек более устойчивы к масштабированию, поворотам и трансформациям изображения, а также к окклюзиям сцены. Точность методов зависит от числа особых точек объекта, выделяемых детектором и сопоставляемых матчером. Скорость работы методов определяется скоростью работы всех трех составляющих методов: детектора, дескриптора и матчера [9, 18-20]. Они выбираются исходя из проводимых экспериментов с измерением времени и скорости работы, оценкой устойчивости к трансформациям изображений и изучением литературы, в которой данные методы уже реализованы.

Главным достоинством методов на основе обнаружения особых точек является их инвариантность к вращениям, масштабированию и окклюзии изображения. Однако, в основном, эти методы требуют наличия ярких текстур на объектах для обнаружения и сопоставления характерных точек. В результате

они не всегда применимы к объектам со слабо выраженной текстурой а также к однородным по цвету объектам.

### 1.1.5 Алгоритмы сегментации

#### 1.1.5.1 MeanShift

MeanShift относится к группе алгоритмов сегментации изображений. Он группирует объекты с близкими признаками. Пиксели со схожими признаками объединяются в один сегмент, на выходе получаем изображение с однородными областями.

К достоинствам можно отнести, что алгоритм показывает хорошие результаты с однородными по цвету объектами на пестром фоне. К недостаткам то, что если объект, который хотим выделить, состоит из областей, сильно различающихся по цвету, то MeanShift не сможет объединить эти регионы в один, и объект будет состоять из нескольких сегментов. [21, 22]

Для сцен с множеством мелких объектов и неоднородным фоном для правильной сегментации требуется вмешательство человека-оператора, который поставит множество точек или других меток на объекты сцены для улучшения показателей работы алгоритма, что уменьшает возможную сферу применимости метода.

#### 1.1.5.2 Метод Оцу

Метод Оцу также относится к группе алгоритмов сегментации изображений. Он относится к методам цветовой классификации. Для цветовой классификации необходимо подобрать значение порога таким образом, чтобы разделить изображение на два класса: заданный объект и фон. Пиксели, цвет которых превышает пороговое значение или находится в заданном интервале, относятся к классу объект, все остальные пиксели относятся к классу фон.

Для расчета порогового значения применяется гистограмма изображения. Рассмотрим простейший случай, когда на изображении один объект. По гистограмме видно два отличных друг от друга класса (рисунок 1).

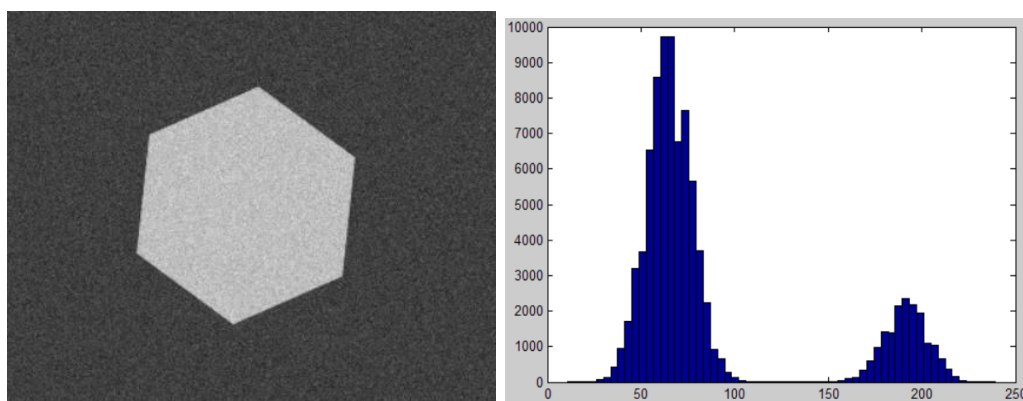


Рисунок 1 – Пример изображения с объектом и гистограмма изображения.

Суть метода заключается в подборе порога между классами или их диапазонов таким образом, чтобы пиксели из одного класса могли попасть в другой класс с наименьшей вероятностью (рисунок 2).

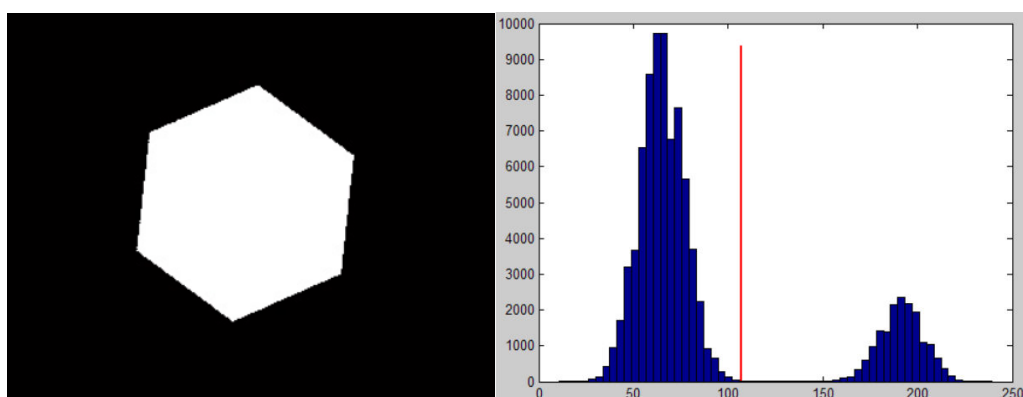


Рисунок 2 – Результат применения метода Оцу.

Достоинствами этого метода являются:

- Простота реализации;
- Метод хорошо адаптируется к различного рода изображения, выбирая наиболее оптимальный порог;
- Быстрое время выполнения ( $O(N)$  операций).

К недостаткам можно отнести, что сама по себе пороговая бинаризация чувствительна к неравномерной яркости изображения. Решением проблемы может быть введение нескольких пороговых значений для каждого класса,

вместо одного глобального [23]. Также метод неприменим к сценам с окклюзией и к объектам, имеющим схожий цветовой тон с фоном.

Методы сегментации не могут быть напрямую применены для детектирования объекта в рамках рассматриваемой задачи, однако, они могут использоваться для уточнения формы объекта на изображении.

#### 1.1.6 Машинное обучение

Для обнаружения однородных и слабо текстурированных объектов на изображении, как указывалось выше, метод на основе особых точек неприменим.

Для решения данной проблемы могут применяться методы машинного обучения с применением уже не одной картинки с объектом, а учебных наборов из сотен (а в некоторых случаях - сотен тысяч) разных картинок с изображением объекта в разных условиях [8].

Методы машинного обучения позволяют находить объекты, совпадающие с заданными эталонами. Основная особенность методов – обучение процессу подбора положения объекта, а не процедуре сравнения [13-15, 24].

К достоинствам метода можно отнести, что он не требует замкнутости границы, что позволяет применять его к пересекающимся и частично закрытым объектам, и обнаружение объекта с учетом преобразований вращения, масштабирования и параллельного переноса., к недостаткам – требуется большое количество изображений для обучения, вследствие чего для данной задачи метод скорее всего не применим [15].

#### 1.1.7 Идентификация заданного объекта среди найденных

Для сцен, содержащих несколько объектов с близкими цветами, алгоритмы цветовой фильтрации и сегментации изображений не позволяют выделить именно заданный объект. Для детектирования заданного можно воспользоваться процедурой сравнения моментов Хью.

Моменты Хью (моменты изображения) – это инвариантные моменты к масштабированию и повороту объектов [25].

Если есть модели (изображение) эталона искомого объекта – выполняем процедуру сравнения моментов эталона с моментами найденных объектов на изображении, и тот объект, моменты которого будут наиболее близки к моментам эталона – принимаем за искомый объект.

В таблице 1 приведено сравнение рассмотренных методов детектирования объектов.



| Метод/<br>критерий                                      | Цветовая<br>фильтрация                    | Контурный<br>анализ | Сопоставление<br>с шаблоном | Особые<br>точки            | Алгоритмы<br>сегментации         | Машинное<br>обучение |
|---|---|---------------------|-----------------------------|----------------------------|----------------------------------|----------------------|
| Сложность<br>реализации                                 | <b>Легко</b>                              | Легко               | Средняя                     | Средняя                    | Средняя                          | Сложно               |
| Применимость к<br>слабо<br>текстурированным<br>объектам | <b>Да</b>                                 | Да                  | Да                          | Нет                        | Да                               | Да                   |
| Устойчивость к<br>окклюзии                              | <b>Средняя</b>                            | Да                  | Высокая                     | Средняя                    | Слабая                           | Высокая              |
| Устойчивость к<br>масштабированию                       | <b>Да</b>                                 | Нет                 | Высокая                     | Да                         | Да                               | Да                   |
| Начальная<br>информация об<br>объекте                   | <b>Внешний<br/>вид объекта<br/>(цвет)</b> | Контур<br>объекта   | Кадры объекта               | Особые<br>точки<br>объекта | Внешний вид<br>объекта<br>(цвет) | Учебный<br>набор     |

Таблица 1 – Сравнение методов детектирования объектов

## 1.2 Определение положения объекта

Для определения положения 3D – объекта на монокулярном изображении, существуют различные методы. Некоторые методы основаны на глобальном сопоставлении шаблонов с использованием реальных шаблонов объекта. Некоторые из них рассматривают внешний вид [26, 27] или форму [28-32] для представления объекта. Другие основаны на сопоставлении локальных особенностей, описываемых дескрипторами, например, такими как HOG (Histogram of Oriented Gradients) [10, 11], SIFT (Scale-Invariant Feature Transform) [12]. Также существуют подходы основе глубоких нейронных сетей для определения положения объектов. В [4] предложен метод на основе глубокого обучения для итеративного определения положения объекта. В [5] для определения положения объекта на сцене с сильной окклюзией применяются свёрточные нейронные сети.

В общем случае, для определения положения объекта относительно камеры необходимы соответствия между известными в реальном мире 3D – объектами и их 2D аналогами в плоскости изображения.

Возможные подходы решения данной задачи:

- PnP, PnL;
- Определение положения объекта по его модели;
- Машинное обучение (иерархический поиск).
- RGB-D изображения;
- Нейронные сети.

### 1.2.1 PnP, PnL

Традиционно, проблема определения положения трехмерного объекта решается путем сопоставления характерных особенностей между 3D-моделью и реальным изображением объекта [10-12, 33]. PnP (Perspective n Points) – это задача определения положения объекта по n особым точкам модели объекта и его аналогом на изображении.

PnP метода заключается в нахождении на модели объекта и на тестируемом изображении особых точек с последующим сопоставлением особых точек с изображением камеры с точками модели. И по известным зависимостям между особыми точками на модели объекта можно определить положение объекта в пространстве относительно камеры.

Для решения задачи необходимо задать  $n$ -соответствующих точек между точками изображения и точками модели объекта, а также известные заранее взаимные расположения точек модели объекта.

Задачи для одной и двух особых точек имеют бесконечное число решений. Наибольший интерес представляет P3P проблема, так как для решения этой задачи достаточно трех соответствующих точек на изображении и модели. К сожалению, при точном решении P3P проблемы необходимо решать уравнения четвертой степени [34]. Также возможны такие конфигурации точек, что задача дает от 2 до 4-х различных решения, а выбор единственного верного решения является сложной задачей, требующей большого объема вычислений.

Существует оптимизированный метод определения положения на основе PnP с применением кватернионов [35].

Задачи для точек больше шести имеют единственное решение, но при этом требуют большое процессорное время [34].

PnP метод имеет достаточно хорошие результаты для объектов, обладающих большим числом особенностей, которые существенным образом отличаются от фона. В идеальном случае на объекте должно быть столько особенностей, чтоб в любом положении камеры относительно объекта можно было сопоставить 6 особых точек объекта, так как, чем больше удастся сопоставить особенностей, тем меньше требуется ресурсов для получения одного верного решения. PnP метод при малом числе  $n$ -соответствий обладает трудоемкостью получения решения и невозможностью применения для

объектов со слабо выраженными особенностями, или с малым числом особенностей.

Для данной задачи метод скорее всего не применим, так как в работе рассматриваются однородные по цвету объекты, которые могут являться телами вращения, т.е. имеют слабо выраженные особенности.

Также в качестве черт могут выступать линии (границы) объекта.

PnL – это задача, аналогичная задаче PnP, только в качестве особенностей объекта используются линии пересечения граней объекта.

Условием для решения задачи является задание  $n$ -соответствий между линиями изображения и линиями (гранями) модели объекта, а также известные заранее взаимные расположения точек модели объекта.

Если у объекта можно выделить ярко выраженные прямые линии пересечения граней, то метод на основе установления соответствий линий реального объекта на изображении и его модели предпочтительнее, так как на изображении легче выделить линии, нежели особые точки. Это связано с тем, что для детектирования особых точек требуется определение особенностей объекта, совпадающих с предварительно известными шаблонами, а для детектирования линий достаточно вычислить лишь градиент яркости пикселей изображения.

В [36] предложен метод DLT – Combined Lines на основе алгоритма DLT (Direct Linear Transformation – прямого линейного преобразования), который использует новую комбинацию на основе существующих методов DLT для определения положения объекта.

Подход PnP был изучен в первую очередь, так как точки легче обрабатывать математически, чем линии. Однако, PnP ограничивается только для объектов с достаточным количеством особых точек, т.е. в основном только для сильно текстурированных сцен, тогда как PnL подход, подходит для слабо текстурированных сцен.

Определение положения объекта по его модели

Суть подхода заключается в том, что заранее известна геометрия объекта. Таким образом, мы можем смоделировать объект и производить над ним любые действия по перемещению, вращению и масштабированию, сохраняя параметры изображения при этих действиях. Тем самым можно собрать базу данных изображений всех возможных положений объекта (шаблонов) в пространстве и, далее, при помощи этой базы производить распознавание.

Подход на основе шаблонного сравнения основан на сравнении исходного изображения со всеми шаблонами, имеющимися в базе и выборе шаблона с наименьшим количеством пикселей (было - точек), отличных от входного изображения.

Для шаблонного сравнения можно применять методы машинного обучения [38]. В [39] для сравнения с шаблоном применяется метод быстрой корреляции с использованием тернарных шаблонов.

Достоинством подхода на основе шаблонного сравнения является довольно высокая устойчивость к дефектам изображения, высокая скорость обработки входных данных. Но надежно распознаются только те объекты, шаблоны которых «известны».

Недостатком подхода распознавания объекта по его модели является необходимость получения большого объема первоначальных данных и чем выше требуется точность, тем этот объем должен быть больше. Для получения базы данных изображений положений объекта необходимо построить модель объекта и смоделировать его вращение, а также обработать все полученные изображения при моделировании, что может занять много времени и ресурсов [40].

Подход на основе шаблонных сравнений отлично подходит для решения данной задачи для выбранного объекта, так как геометрия объекта известна и построить его модель не составит труда.

Во многих случаях геометрия искоемых объектов известна заранее. На основании этого рассматривается модель объекта, представляющая собой набор черт и их взаимное расположение.

Модельное изображение с реальным можно сравнивать несколькими способами:

- По количеству совпадающих пикселей (абсолютная разность пикселей одного изображения от другого);
- С помощью моментов контуров;
- Средняя ошибка;
- Среднеквадратичное отклонение пикселей одного изображения от другого;
- Ковариация;
- Сравнение гистограмм изображений;
- Сравнение по ключевым точкам изображения (сегментированных объектов);
- Использование каскада Хаара;
- С помощью перцептуальных хешей (это свертка каких-либо признаков, которые описывают картинку).

### 1.2.2 Машинное обучение (иерархический поиск)

Поскольку процесс сопоставления всего набора представлений (положений) объекта с извлеченным представлением объекта из входного изображения может быть сложным с вычислительной точки зрения, предлагается группировать изображения объекта в иерархический граф представлений (дерево).

Метод имеет следующие достоинства:

- Робастный;
- Быстрый (определение положения за несколько сотых секунд);
- Устойчив к окклюзии;

- Используется только информация о геометрии объекта, следовательно, метод применим к слабо текстурированным (однородным) объектам.

Так как выбранный объект однородный и его геометрия известна, представленный выше метод может быть применим для решения данной задачи.

### 1.2.3 Определение положения на основе RGB-D

Определение положения на основе RGB-D изображений – достаточно распространенный подход. Когда доступна карта глубины пикселей изображения, ее можно комбинировать с RGB-изображениями для улучшения определения положения. Общая стратегия использования карты глубины заключается в преобразовании карты глубины в трехмерное облако точек и сопоставлением трехмерной модели объекта с полученным трехмерным облаком точек. Например, в [42] трехмерная модель объекта отображается в шаблон нормалей поверхности объекта, а затем этот шаблон сопоставляется с нормальями, вычисленными из облака точек.

Получить RGB-D изображение можно различными способами:

- Intel RealSense;
- Времяпролетная камера;
- Стереокамера;
- Смещение камеры.

### 1.2.4 Intel RealSense

Для преодоления проблем PnP метода к камере можно добавить проектор и получить устройство регистрации трехмерных образов объектов, основанное на активном параллаксном принципе, т.е. наложить на объект визуальную текстуру. В такой системе на объект проецируется некоторая картина (структурированная подсветка), и ее искажения, вызванные формой

объекта, регистрируются камерой [41, 43]. Такой камерой со структурированной подсветкой является Intel RealSense, изображенной на рисунке 3.



Рисунок 3 – Intel RealSense.

В настоящее время разработано множество различных вариантов картин для использования в системах структурированной подсветки, представляющих собой как серии изменяющихся картин, так и неизменные картины с использованием различных вариантов цветовой кодировки [41, 43]. Поскольку шаблон (картина) закодирован, то можно легко найти соответствия между точками изображения и точками проецируемого шаблона.

Данный метод хорошо подходит как для объектов со слабо выраженной текстурой, так и для объектов с ярко выраженной текстурой, поэтому для решения данной задачи метод может быть применим.

#### 1.2.4.1 Времяпролетная камера

Времяпролетная камера (ToF – Time of Flight) - это метод измерения расстояния между датчиком, расположенным на камере, и объектом, основанный на разнице во времени между испусканием сигнала и его возвратом в датчик, после отражения от объекта. Основной принцип работы времяпролетной камеры изображен на рисунке 4.





Рисунок 4 – Принцип работы времяпролетной камеры.

С помощью ToF можно получить подробную 3D-информацию об объекте. Для освещения сцены обычно используется инфракрасный или ближний инфракрасный свет, камера ToF может измерять расстояние между собой и объектами внутри сцены, в которой она расположена. По сравнению с другими методами получения 3D-информации - например, с использованием структурной подсветки или стереоскопического зрения - камеры ToF способны выдавать большую точность, будучи чрезвычайно быстрыми и доступными [44].

Системы камер ToF состоят из сенсора, процессора обработки изображения и модулирующего источника света [45]. ToF системы освещают сцену источником света, а затем измеряют фазовый сдвиг волны, которая отражается назад. Поскольку свет имеет постоянную скорость, камеры ToF могут вычислять расстояние до каждой точки сцены в зависимости от времени, которое требуется для того, чтобы этот свет возвращался к камере. Вместо того, чтобы сканировать изображение по линиям, система камер ToF освещает всю сцену сразу, а затем измеряет фазовый сдвиг света, отраженного обратно на сенсор. Камеры ToF позволяют быстро и эффективно определять расстояния до объектов за счет чрезвычайно высокой частоты получения кадров (даже сверх той, которую может видеть зрение человека) и встроенных процессоров. Это означает, что в отличие от многих других трехмерных представлений, ToF позволяет извлекать информацию о глубине 3D точек сцены в реальном времени [44].

Достоинства:

- Компактность [46];
- Простота алгоритмов определения расстояния. ToF вычисляет расстояние практически напрямую с сенсора. В результате используется лишь небольшая вычислительная мощность [46].
- Скорость. Поскольку камеры работают со скоростью до 160 кадров в секунду, они идеально подходят для использования в приложениях реального времени [46].

Недостатки:

- Окружающий свет. Так как дневной свет содержит весь световой диапазон, и он в разы мощнее испускаемого света времяпролетной камерой, окружающий свет может вносить значительные помехи в определении расстояния до объекта [45].
- Интерференция света. Если рядом одновременно работает несколько времяпролетных камер, то они могут искажать измерения друг друга. Решением проблемы может быть введение мультиплексирования времени, т.е. включения камер по очереди [46].
- Многократное отражение. В отличие от лазерных сканирующих систем, в которых подсвечивается одна точка, времяпролетная камера подсвечивает сразу всю сцену. Наличие углов и изогнутых поверхностей приводит к возникновению ошибок вследствие многократного отражения света [45 46].

Для данной задачи метод не применим, так как на сцене могут присутствовать объекты с изогнутой поверхностью и углами.

#### 1.2.4.2 Стереокамера

Также карту глубины можно получить с помощью стереокамеры.

Система стерео зрения используется для построения карты глубины по двумерным изображениям, полученным одновременно с двух идентичных

видеокамер, расположенных под разными углами к объекту наблюдения. Эта задача выполняется путем нахождения идентичных точек на обоих изображениях. После нахождения таковых можно выполнить триангуляцию, определить координаты прообраза в трехмерном пространстве и расстояние до плоскости камеры. Пример стереокамеры приведен на рисунке 5.



Рисунок 5 – Стереокамера Raspberry Pi

#### 1.2.4.3 Смещение камеры

Для построения карты глубины, как правило, требуется несколько изображений одной сцены с разного ракурса. Получить изображения можно путем смещения камеры на заданное расстояние. Однако такой способ существенно усложняет конструкцию платформы, на которую устанавливается камера, что снижает спектр применения данного способа.

#### 1.2.5 Нейронные сети

В последнее время подходы, основанные на нейронных сетях, обеспечивают приемлемые результаты для большинства задач компьютерного зрения, включая определение положения объектов по изображению.

Основная идея подходов на нейронных сетях заключается в обучении нейронной сети искусственно создаваемой выборкой изображений заданного объекта. В результате, не требуется большая реальная выборка изображений объекта.

В таблице 2 приведено сравнение методов определения положения объекта.

| Метод/критерий                          | Дороговизна оборудования | Сложность реализации | Применимость к слаботекстурированным объектам |
|---|--------------------------|----------------------|---|
| PnP                                     | Нет                      | Средне               | Нет   |
| PnL                                     | Нет                      | Средне               | Да  |
| Сопоставление с моделью объекта         | Нет                      | Средне               | Да  |
| Машинное обучение (иерархический поиск) | Нет                      | Сложно               | Да  |
| Intel RealSense                         | Да                       | Средне               | Необходимо проверить                          |
| Времяпролетная камера                   | Да                       | Легко                | Да  |
| Стереокамера                            | Да                       | Средне               | Нет   |
| Смещение камеры                         | Нет                      | Сложно               | Нет   |
| Нейронные сети                          | Нет                      | Сложно               | Да  |

Таблица 2 – Сравнение метод определения положения объектов

В результате сравнения для дальнейшей реализации выбраны методы на основе машинного обучения, сопоставления с моделью объекта, нейронных сетей, а также по возможности с камерой Intel RealSense.

## 2 Глава 2: Выбранные технические решения

### 2.1 Цветовая фильтрация

Как правило, изначально изображение представляется в RGB – цветовой модели. В модели RGB (от англ. red – красный, green – зелёный, blue – голубой) каждый цвет получается смешением трех базовых (красного, зеленого и синего) цветов в различных пропорциях, но у данной модели есть существенный недостаток. Человеку трудно оперировать цветами, заданными в этой модели, т.к. человек, описывая цвета, пользуется не содержанием в описываемом цвете базовых составляющих, а другими понятиями.

Человеку удобнее оперировать с такими понятиями как: цветовой тон, насыщенность и яркость, которые присущи цветовой модели HSV (Hue – цветовой тон, Saturation – насыщенность, Value – значение цвета (яркость)). Достоинством данной модели является то, что оперировать всем спектром цветов можно при помощи одной переменной.

Поэтому на первом этапе следует изменить цветовую модель изображения на HSV.

На втором этапе необходимо повысить контрастность изображения (ввиду не идеальности камеры и колебаний освещения) – это можно сделать методом эквализации (выравнивания) гистограммы изображения.

Под гистограммой изображения будем понимать графическое представление распределения интенсивности пикселей изображения. Она определяет количество пикселей каждой интенсивности изображения.

Эквализация гистограммы подразумевает отображение одного распределения (начальной гистограммы) на другое распределение (более широкое и равномерное распределение интенсивности) [6].

На третьем этапе – сглаживание (или размытие), так как после выравнивания появляется много небольших по размеру областей резкого изменения яркости/ шума). Размытие изображения достигается применением матрицы свертки. Матрицы свертки – это матрица коэффициентов, которая

умножается на значения пикселей изображения для получения требуемого результата. Фильтров, использующих матрицу свертки, достаточно много. Обычно матрица свертки заполняется по нормальному (гауссовскому закону) [47-49]. В таблице 3 приведена матрица размытия 5x5, заполненная по закону Гаусса.

|          |          |          |          |          |
|----------|----------|----------|----------|----------|
| 0,000789 | 0,006581 | 0,013347 | 0,006581 | 0,000789 |
| 0,006581 | 0,054901 | 0,111345 | 0,054901 | 0,006581 |
| 0,013347 | 0,111345 | 0,225821 | 0,111345 | 0,013347 |
| 0,006581 | 0,054901 | 0,111345 | 0,054901 | 0,006581 |
| 0,000789 | 0,006581 | 0,013347 | 0,006581 | 0,000789 |

Таблица 3 – Матрица распределения Гаусса

Также для уменьшения шума на изображении применяется медианный фильтр. Фильтр работает с матрицами различного размера, размер матрицы влияет только на количество рассматриваемых пикселей вокруг центрального. Медианный фильтр заключается в следующем: для текущего пикселя, пиксели, которые «попадают» в матрицу, сортируются, и выбирается среднее значение из отсортированного массива. Это значение и является выходным для текущего пикселя.

На четвертом этапе – пороговая фильтрация (бинаризация). Целью операции бинаризации является радикальное уменьшение количества информации, содержащейся на изображении. Так как фильтрация осуществляется в HSV – пространстве, в процессе бинаризации в исходном изображении в соответствующих каналах выбираются диапазоны значений, которым соответствует искомый объект, и изображение преобразуется в черно-белое изображение, пиксели которого имеют только два значения – 0 и 1, в зависимости от того, к какому уровню яркости относится пиксель.

Пятый этап – морфологическое преобразование.

Основными операциями математической морфологии являются наращивание, эрозия, замыкание и размыкание. Наращивание увеличивает область

изображения, то есть происходит увеличение ярких (белых) объектов, а эрозия приводит к увеличению темных объектов, операция замыкания позволяет замкнуть внутренние отверстия области и устранить заливы вдоль границы области, операция размыкания помогает избавиться от маленьких фрагментов, выступающих наружу области вблизи её границы [50].

Метод на основе цветовой фильтрации достаточно прост в реализации, может быть применен для выделения объектов на изображении, но имеет ряд ограничений: необходимо заранее знать цвет объекта; для разноцветных объектов и объектов, цвет которых схож с фоном, для изображений с большой окклюзией метод не применим.

## 2.2 Машинное обучение

В [38] представлен метод шаблонного сравнения для определения положения объектов относительно камеры с применением CAD-систем.

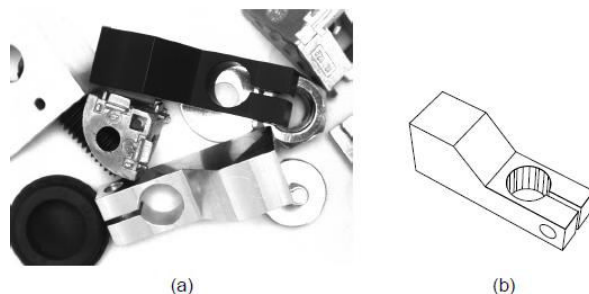


Рисунок 5 – Изображение двух металлических зажимов разного цвета. Модель зажима в САПР (b), показанного на (a), которая служит в качестве входных данных. Для визуализации скрытые края, а также края между копланарными гранями не отображаются.

Строится дерево определения положения: есть 3д модель объекта (рисунок 5), в cad-системе виртуальные камеры размещаются в предзаданных позициях вокруг объекта и объект проецируется на плоскость изображения каждой камеры. Предполагается, что объект находится в центре сферы, задаваемой сферической системой координат. Виртуальные камеры, которые используются для создания таких представлений (изображений) объекта, располагаются таким образом, что они все направлены в центр объекта. Диапазон позиций тогда будет ограничен определенным сферическим четырехугольником, задаваемым сферическими параметрами: широтой, долготой и радиусом.

Выборка представлений в пределах набора позиций автоматически определяется во время генерации 3д модели, чтобы максимизировать надежность и скорость распознавания объекта. Для дальнейшего увеличения скорости распознавания, модель объекта создается на нескольких уровнях пирамиды изображения. Поскольку более высокие уровни пирамиды позволяют более грубую выборку представлений, вычисление представлений выполняется для каждого уровня пирамиды отдельно.



Выборка представления начинается с самого низкого уровня пирамиды изображений, путем применения передискретизации выборки представлений. Затем сходство между всеми представлениями соседних позиций камеры вычисляется путем применения меры подобия, которая используется в онлайн фазе. Пара представлений с наивысшим сходством выбирается и объединяется в одно представление, и вычисляется сходство между новым представлением и его соседними. Этот процесс повторяется до тех пор, пока наибольшее сходство не будет ниже определенного порогового значения.

Если пары представлений объекта, чье сходство превышает пороговое значение, не остается, то остальные представления копируются в 3д модель. Вычисленные представления выше, хранятся на самом нижнем уровне пирамиды.

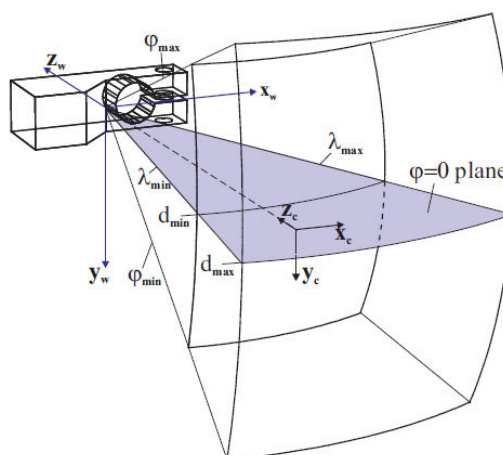


Рисунок 6 – Сферическая система координат сферы взгляда на модель объекта.

Положение мировой системы координат ( $x_w, y_w, z_w$ ) относительно системы координат камеры ( $x_c, y_c, z_c$ ) может быть описано долготой ( $\lambda$ ), широтой ( $\phi$ ) и расстоянием ( $d$ ). Диапазон положений описывается минимальными и максимальными значениями  $\lambda$ ,  $\phi$  и  $d$ .

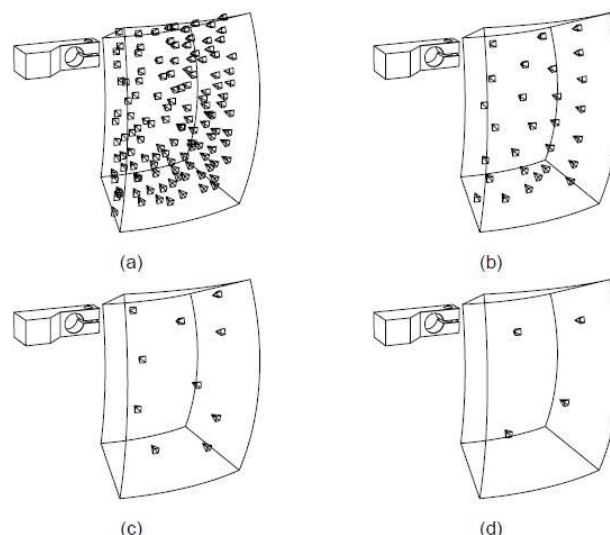


Рисунок 7 - Результирующие представления модели на уровне 1 (a) пирамиды, уровне 2 (b), уровне 3 (c) и уровне 4 (d). Представления визуализируются маленькими квадратными пирамидами, которые представляют камеры.

На рисунке 7(a) для всех представлений на нижнем уровне пирамиды визуализируются соответствующие камеры, которые получаются при применении описанного метода к диапазону позиций, показанного на рисунке 6. Чтобы получить представления на следующем уровне пирамиды, объединение продолжается при уменьшении порогового значения подобия. Для этого показатель подобия вычисляется на субдискретизированном (уменьшенном) изображении соответствующего уровня пирамиды изображения. Порог подобия автоматически уменьшается, т.к. более мелкие различия устраняются за счет уменьшения разрешения изображения. Если никакая пара представлений объекта, чье сходство превышает пороговое значение, не остается, то остальные представления копируются в соответствующий уровень 3д модели. На рисунке 7(b-d) показаны представления, полученные на более высоких уровнях пирамиды. В этом примере достаточно выделить только четыре разных представления на четвертом уровне пирамиды.

Кроме того, при каждом представлении сохраняется ссылка на все дочерние представления. Дочерние представления – это те представления на

следующем нижнем уровне, которые были объединены, чтобы получить представление на текущем уровне пирамиды или представление, которое невозможно было объединить. Ссылки хранятся в древовидной структуре. Эта информация используется в онлайн-фазе для запроса определенного представления на более высоком уровне пирамиды представлений на следующем нижнем уровне пирамиды, которые используются для уточнения совпадений.

При шаблонном сравнении изображения модели объекта, полученные виртуальной камерой, не имеют искажений, так как модель виртуальной камеры принимается за идеальную. На деле, реальная камера не идеальна, следовательно, изображения объектов на реальной камере – также не идеальны, с искажениями. Для устранения этой проблемы требуется определить калибровочные коэффициенты камеры.

Для получения данных коэффициентов существуют готовые программные решения с применением библиотеки OpenCV. Для этого необходимо подготовить калибровочный образец – изображение шахматной доски с известной стороной каждого квадрата, сделать 5-6 снимков образца на калибруемую камеру.

### 2.2.1 Генерация эталонных изображений модели объекта

Для повышения точности определения положения необходимо получить большой набор эталонных снимков объекта. Поэтому для получения набора снимков следует применять автоматизированные средства рендеринга модели объекта. В процессе анализа литературы найдены два способа для генерации изображений объекта:

- Использование скриптов для AutoDesk 3ds MAX [51];
- Использование сценариев для AutoCAD [52, 53].

### 2.2.2 Калибровка камеры

Для повышения точности калибровки следует делать изображения калибровочного образца с разным приближением, наклоном и поворотом. И на основании полученных изображений можно получить калибровочные коэффициенты камеры для удаления искажений с изображений.

В результате откалиброванные изображения можно использовать в методах шаблонного сравнения определения положения объектов.

### 2.2.3 Масштабирование изображений

Для шаблонного сравнения изображений также требуется, чтобы изображения были в одном масштабе, в противном случае, метод шаблонного сравнения неприменим. Так как при генерации эталонных изображений используется виртуальная камера, положение которой относительно модельного объекта известно, можно растягивать изображения до одинакового размера и принять масштаб эталонных изображений за 1. Тогда следует масштабировать реальное изображение объекта до эталонного. Для этого следует изменять размер реального изображения объекта до заданного эталонного.

### 2.2.4 Положение объекта в пространстве

Положение объекта в пространстве определяется шестью координатами: 3 координаты задают смещение объекта относительно начала отсчета, и 3 координаты (угла) задают поворот объекта относительно начала отсчета. Расстояние до объекта определяется по соотношению размера реального изображения объекта и эталонного; вертикальное и горизонтальное смещение - по положению центра масс объекта на изображении; три угла поворота – по шаблонному сравнению.

### 2.3 Определение положения на основе нейронных сетей

В [5] предлагается новая свёрточная нейронная сеть для определения положения объекта, названная Pose CNN. Основная идея Pose CNN состоит в том, чтобы разделить задачу определения положения на различные этапы, которые позволят сети точно моделировать зависимости и независимости между ними. В частности, нейронная сеть Pose CNN решает три связанные задачи, показанные на рисунке 8.

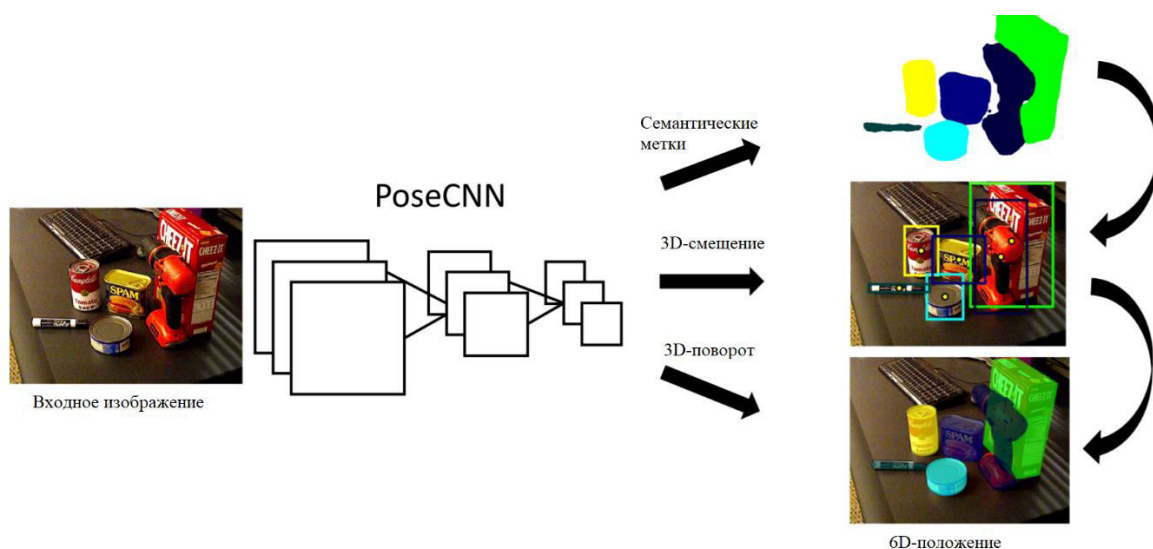


Рисунок 8 – Предложенная новая сеть Pose CNN для определения положения объекта, сеть обучается выполнять три задачи: семантическую маркировку, определение трехмерного смещения и регрессию трехмерного вращения.

Во-первых, она предсказывает метку объекта для каждого пикселя во входном изображении (семантическая маркировка объектов). Во-вторых, определяет двумерные пиксельные координаты центра объекта путем прогнозирования единичного вектора от каждого пикселя в направлении центра. Используя семантические метки, пиксели изображения, связанные с объектами, голосуют за местоположение центра объекта на изображении. Кроме того, сеть также оценивает расстояние до центра объекта. Предполагая известные матрицы коэффициентов камеры, определение 2D-центра и расстояния до него позволяют восстановить его смещение в пространстве (T).

Наконец, 3D – поворот оценивается путем регрессии свёрточных элементов, извлеченных внутри ограничительной рамки объекта в представление в кватернионах. В статье показывается, что Pose CNN может применяться к текстурированным, однородным объектам, она устойчива к окклюзиям поскольку обучается определять положение по центру объекта, даже когда они закрыты.

При заданном входном изображении задача определения 6D-координат объекта состоит в том, чтобы определить жесткое преобразование из системы координат объекта  $O$  в систему координат камеры  $C$ . Авторы предполагают, что трехмерная модель объекта известна, и система координат объекта определена в трехмерном пространстве модели.

Жесткое преобразование состоит из преобразования  $SE(3)$ , содержащего трехмерное вращение  $R$  и трехмерное смещение  $T$ , где  $R$  задает углы поворота вокруг осей  $X$ ,  $Y$  и  $Z$  системы координат объекта  $O$ , а  $T$  представляет собой координаты начала координат  $O$  в системе координат камеры  $C$ . В процессе формирования изображения  $T$  определяет местоположение объекта и масштаб на изображении, в то время как  $R$  влияет на внешний вид изображения объекта (поворот) в соответствии с 3D-моделью и текстурой объекта.

Поскольку эти два параметра имеют различные визуальные свойства, предлагается свёрточная архитектура нейронной сети, которая внутренне разделяет определение  $R$  и  $T$ . Рисунок 7 иллюстрирует архитектуру сети Pose CNN для определения 6D координат объекта. Сеть состоит из двух ступеней. Первая ступень состоит из 13 свёрточных слоев и 4 слоев подвыборки, которые извлекают карты признаков с разными разрешениями входного изображения. Эта ступень является основой сети, поскольку извлеченные признаки являются общими для всех задач, решаемых сетью. Вторая ступень состоит из этапа внедрения, который встраивает карты признаков, полученные на первой ступени, в низкоразмерные признаки для конкретной задачи. Затем, сеть выполняет три разные задачи: семантическая маркировка, определение

смещения в пространстве и определение трехмерного поворота, в результате выполнения которых сеть определяет положение объекта.

На этапе внедрения ветви семантической маркировки, как показано на рисунке 9, в качестве входных данных принимаются две карты признаков с размером канала 512, сгенерированных первой ступенью. Карты признаков имеют разрешение  $1/8$  и  $1/16$  от исходного размера изображения соответственно. Сеть сначала уменьшает размер канала двух карт до 64, используя два свёрточных слоя. Затем, она удваивает разрешение карты объекта, размером  $1/16$  с помощью транспонированного свёрточного слоя (ТСС). После этого карты признаков суммируются и далее с помощью еще одного транспонированного свёрточного слоя разрешение суммированной карты увеличивается в 8 раз до размера исходного изображения. Далее, свёрточный слой действует на карте признаков и формирует оценки семантической маркировки для пикселей. На выходе этого слоя  $n$  каналов с  $n$  количеством семантических классов. В процессе обучения применяется softmax ошибка перекрестной энтропии, а softmax cross entropy loss для обучения ветви семантической маркировки. Во время тестирования, softmax функция используется для вычисления вероятности принадлежности пикселя к классам. Конструкция ветви семантической маркировки взята из [54].

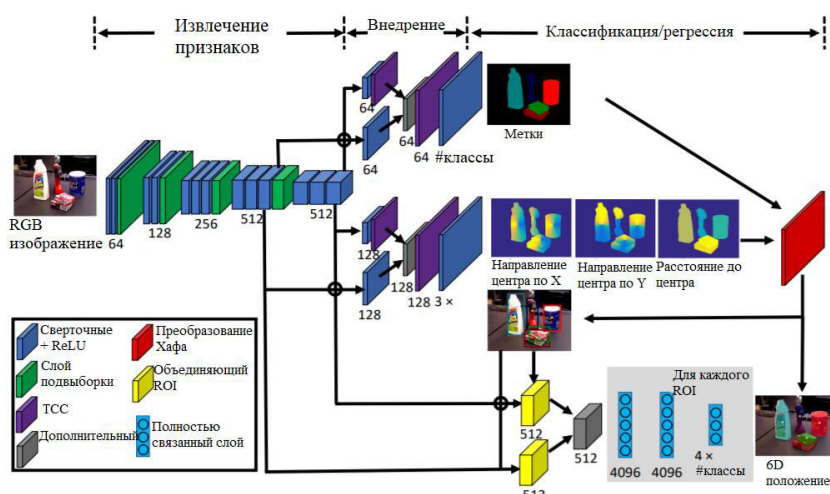


Рисунок 9 – Архитектура Pose CNN для определения положения объекта  
Определение трехмерного смещения

Как показано на рисунке 10,  $T = (T_x, T_y, T_z)^T$  – есть координаты начала системы координат объекта в системе координат камеры. Наивным способом оценки  $T$  является прямая регрессия элементов изображения в  $T$ . Однако этот способ не может быть обобщен, поскольку объекты могут находиться в любом месте изображения. Кроме того, он не может обрабатывать несколько объектов одного класса. Поэтому предлагается определить  $T$  путем локализации центра 2D-объекта на изображении и определения расстояния до объекта от камеры. Предположим, что проекция  $T$  на изображении  $c = (c_x, c_y)^T$ .

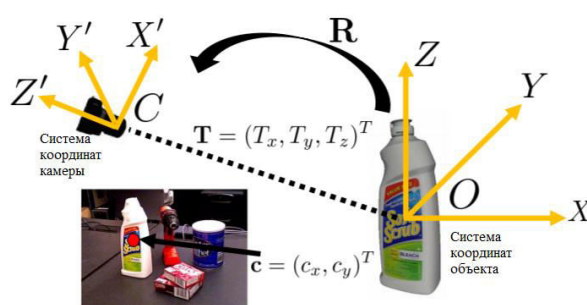


Рисунок 10 - Иллюстрация системы координат объекта и системы координат камеры.

Трехмерное смещение можно определить путем локализации 2D-центра объекта и оценки расстояния до 3D-центра от камеры. Если сеть может локализовать  $c$  на изображении и определить глубину  $T_z$ , можно восстановить  $T_x$  и  $T_y$  согласно следующему уравнению для проекции, предполагая, что у нас камера Обскура:

$$\begin{bmatrix} c_x \\ c_y \end{bmatrix} = \begin{bmatrix} f_x \frac{T_x}{T_z} + p_x \\ f_y \frac{T_y}{T_z} + p_y \end{bmatrix}, \quad (1)$$

где  $f_x$  и  $f_y$  – фокусные расстояния камеры, а  $(p_x, p_y)$  – центр изображения. Если начало системы отсчета объекта  $O$  является центром тяжести,  $c$  – называем 2D-центром объекта.

Простой способ локализации центра 2D-объекта состоит в непосредственном обнаружении центральной точки, как в существующих методах обнаружения ключевых точек [55]. Однако эти методы не будут



работать, если центр будет закрыт. Для обхода этой проблемы авторы предлагают спроектировать нейронную сеть так, чтобы она регрессировала в направлении центра для каждого пикселя изображения на основе традиционной модели неявной формы, где участки изображения голосуют за центр объекта. В частности, для пикселя  $p = (x, y)^T$  изображения она регрессирует до трех переменных:

$$(x, y) \rightarrow \left( n_x = \frac{c_x - x}{\|c - p\|}, n_y = \frac{c_y - y}{\|c - p\|}, T_z \right). \quad (2)$$

Заметим, что вместо прямой регрессии к вектору  $c-p$ , спроектируем сеть так, чтобы она регрессировала к вектору единичной длины  $\mathbf{n} = (n_x, n_y)^T = \frac{c-p}{\|c-p\|}$ , т.е. в направлении 2D-центра, которое не зависит от масштаба и поэтому сеть легче обучать. Центральная ветвь регрессии предложенной сети (рисунок 7) использует ту же архитектуру, что и ветвь семантической маркировки, за исключением, что размеры каналов сверточных слоев и транспонированных сверточных слоев различны. Авторы встраивают многомерные признаки в 128-мерное пространство вместо 64-мерного, поскольку эта ветвь должна регрессировать 3 переменных для каждого класса. Последний сверточный слой в этой ветви имеет размерность  $3 \times n$ , с  $n$ -числом классов. При обучении сглаженная функция потерь L1 применяется для регрессии как в [60].

Для того чтобы найти 2D-центр  $c$  в сеть был создан и интегрирован слой преобразования Хафа в нейронную сеть. Слой Хафа принимает результаты семантической маркировки по пикселям и результаты регрессии центра в качестве входных данных. Для каждого класса объектов сначала вычисляется вес локальной области на изображении. Оценка веса показывает, насколько вероятно, что соответствующая локальная область является центром объекта в классе. В частности, каждый пиксель в классе объектов добавляет вес за локальную область изображения вдоль луча, предсказанного сетью (рисунок 4). После обработки всех пикселей в классе объекта мы получаем результаты

весов для всех локальных областей изображения. Затем за центр объекта выбирается область с максимальным весом. В случаях, когда на изображении могут появляться несколько объектов одного класса, применяется не максимальное подавление к весам, а затем выбирается локальная область с весом, превышающим определенный порог.

После нахождения центров объектов рассматриваются пиксели, которые голосуют за центр объекта, как внутренние точки центра. Затем предсказание глубины центра,  $T_z$ , просто вычисляется как среднее значение глубин предсказанных инлайерами («не выбросами», хорошие точки). В конце, используя формулу (1) можно оценить трехмерное смещение  $T$ . Кроме того, сеть генерирует ограничивающий прямоугольник объекта в виде 2D-прямоугольника, который ограничивает все входящие инлайеры, и ограничивающий прямоугольник используется для регрессии трехмерного вращения.

#### Трехмерная регрессия вращения

В нижней части рисунка 7 показана ветвь регрессии трехмерного вращения. Используя ограничивающие прямоугольники объектов, предсказанные из слоя преобразования Хафа, используются два слоя объединения ROI для «объединения и обрезки» [60] визуальных признаков, созданных на первой ступени сети для регрессии трехмерного вращения. Объединенные карты признаков суммируются вместе и подаются в три полностью связанных слоя (FC – Fully Connected). Первые два слоя имеют размерность 4096, а последний слой FC имеет размерность  $4 \cdot n$ , где  $n$  – число классов объекта. Для каждого слоя последний слой FC выводит трехмерное вращение, представленное кватернионом. Для обучения кватернионной регрессии предлагается две функции потерь, одна из которых специально предназначена для обработки симметричных объектов. Первая функция потерь, называемая PoseLoss(PLOSS), работает в пространстве трехмерной модели и измеряет среднеквадратичное расстояние между точками в правильном

положении модели и их соответствующими точками на модели в предполагаемом положении.

PLOSS определяется как:

$$\text{PLOSS}(\tilde{\mathbf{q}}, \mathbf{q}) = \frac{1}{2m} \sum_{\mathbf{x} \in \mathcal{M}} \|R(\tilde{\mathbf{q}})\mathbf{x} - R(\mathbf{q})\mathbf{x}\|^2, \quad (3)$$

где  $\mathcal{M}$  обозначает набор точек 3D-модели и  $m$  – количество этих точек.  $R(\tilde{\mathbf{q}})$  и  $R(\mathbf{q})$  – матрицы вращения, вычисленные из оценочного кватерниона и основного истинного кватерниона соответственно. Эта функция потерь имеет свой уникальный минимум, когда предполагаемая ориентация идентична истинной ориентации. К сожалению, PLOSS не обрабатывает симметричные объекты надлежащим образом, поскольку симметричный объект может иметь несколько подходящих трехмерных положений. Использование такой функции потерь на симметричных объектах негативно отражается на сети для регрессии к одному из альтернативных трехмерных вращений, что дает, возможно, противоречивые обучающие выборки для тренировки.

В то время как PLOSS потенциально может быть изменен для обработки симметричных объектов путем ручного указания симметрии объектов и последующего рассмотрения всех правильных ориентаций в качестве опций истинности, здесь вводится вторая функция потерь ShapeMatchLoss (SLOSS), которая не требует уточнения симметрии. SLOSS определяется как:

$$\text{SLOSS}(\tilde{\mathbf{q}}, \mathbf{q}) = \frac{1}{2m} \sum_{\mathbf{x}_1 \in \mathcal{M}} \min_{\mathbf{x}_2 \in \mathcal{M}} \|R(\tilde{\mathbf{q}})\mathbf{x}_1 - R(\mathbf{q})\mathbf{x}_2\|^2. \quad (4)$$

Видно, что эта функция потерь измеряет смещение между каждой точкой на предполагаемой ориентации модели и ближайшей точкой на истинной модели. SLOSS сводится к минимуму, когда две 3D-модели соответствуют друг другу. Таким образом, SLOSS не будет негативно отражать повороты, которые эквиваленты по отношению к симметрии 3D-модели объекта.



## ЗАКЛЮЧЕНИЕ

В результате проведенной работы были проанализированы различные существующие методы для определения положения объекта в пространстве. Также выбраны наиболее подходящие для заданной категории объектов. Для детектирования выбранного объекта применимы методы цветовой фильтрации и сегментации изображения, для определения положения методы с применением машинного обучения для сопоставления с шаблонами, а также методы на основе нейронных сетей.

## СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

- 1 D. Leng and W. Sun, "Contour-based iterative pose estimation of 3d rigid object," IET computer vision, vol. 5, no. 5, pp. 291–300, 2011.
- 2 Payet N, Todorovic S. From contours to 3D object detection and pose estimation. ICCV; 2011. p. 983–990.
- 3 Ekvall, Staffan, Danica Kragic and Frank Hoffmann. "Object recognition and pose estimation using color cooccurrence histograms and geometric modeling." Image Vision Comput. 23 (2005): 943-955.
- 4 Y. Li, G. Wang, X. Ji, Y. Xiang, and D. Fox, "DeepIM: Deep iterative matching for 6D pose estimation," in Proc. Eur. Conf. Comput. Vis. (ECCV), 2018, pp. 695–711.
- 5 Y. Xiang, T. Schmidt, V. Narayanan, D. Fox: Pose CNN: A Convolutional Neural Network for 6D Object Pose Estimation in Cluttered Scenes 2017
- 6 Гистограмма и гистограммная обработка изображений [http://wiki.technicalvision.ru/index.php/Гистограмма\\_и\\_гистограммная\\_обработка\\_изображений](http://wiki.technicalvision.ru/index.php/Гистограмма_и_гистограммная_обработка_изображений) 2018
- 7 Живрин Я. Э., Алкзир Н. Б. Методы определения объектов на изображении // Молодой ученый. — 2018. — №7. — С. 8-19. — URL <https://moluch.ru/archive/193/48447/> (дата обращения: 02.12.2018). 2018
- 8 О задаче поиска объекта на изображении. // <http://mechanoid.kiev.ua/> URL: <http://mechanoid.kiev.ua/cv-image-detector.html> (дата обращения: 10.11.2018).
- 9 Анализ алгоритмов компьютерного зрения <https://arealidea.ru/articles/statii-publikatsii/analiz-algoritmov-kompyuternogo-zreniya-poiska-obektov-i-sravneniya-izobrazheniy/> 2018
- 10 N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In IEEE CVPR, pages 886–893, 2005.
- 11 Histogram of Oriented Gradients // <https://www.learnopencv.com/> URL: <https://www.learnopencv.com/histogram-of-oriented-gradients/> (дата обращения: 10.11.2018).

- 12 D. Lowe. Distinctive image features from scale-invariant keypoints. IJCV, 60(2):91–110, 2004.
- 13 Large Scale Visual Recognition Challenge 2014 // <http://image-net.org/> URL: <http://image-net.org/challenges/LSVRC/2014/> (дата обращения: 10.11.2018).
- 14 Review of Deep Learning Algorithms for Object Detection // <https://medium.com/> URL: <https://medium.com/zylapp/review-of-deep-learning-algorithms-for-object-detection-c1f3d437b852> (дата обращения: 10.11.2018).
- 15 О задаче поиска объекта на изображении. // <http://mechanoid.kiev.ua/> URL: <http://mechanoid.kiev.ua/cv-image-detector2.html> (дата обращения: 10.11.2018).
- 16 OpenCV documentation: Harris Corner Detection [http://docs.opencv.org/3.0-beta/doc/py\\_tutorials/py\\_feature2d/py\\_features\\_harris/py\\_features\\_harris.html](http://docs.opencv.org/3.0-beta/doc/py_tutorials/py_feature2d/py_features_harris/py_features_harris.html) 2014
- 17 Blob detection // <https://en.wikipedia.org/> URL: [https://en.wikipedia.org/wiki/Blob\\_detection](https://en.wikipedia.org/wiki/Blob_detection) (дата обращения: 15.11.2018).
- 18 E. Karami, S. Prasad, M. Shehata, Image matching using SIFT, SURF, BRIEF and ORB: 466 performance comparison for distorted images, ArXiv Prepr. ArXiv1710.02726.467 abs/1710.0 (2017).
- 19 D.G. Lowe, Object recognition from local scale-invariant features, in: Proc. Seventh IEEE 443 Int. Conf. Comput. Vis., 1999: pp. 1150-1157 vol.2. doi:10.1109/ICCV.1999.790410.
- 20 Эффективные дескрипторы для локального описания изображений // <http://en.cs.msu.ru/> URL: <http://en.cs.msu.ru/sites/cmc/files/docs/lukianitsa.pdf> (дата обращения: 20.10.2018).
- 21 D. Comaniciu, P. Meer Mean shift analysis and applications, IEEE International Conference on Computer Vision, 1999, vol. 2, pp. 1197.
- 22 D. Comaniciu, P. Meer Mean Shift: A Robust Approach Toward Feature Space Analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2002, pp. 603–619.

- 23 Otsu, N., «A Threshold Selection Method from Gray-Level Histograms,» IEEE Transactions on Systems, Man, and Cybernetics, Vol. 9, No. 1, 1979, pp. 62-66.
- 24 Deep Learning in Object Recognition, Detection and Segmentation, Xiaogang Wand, The Chinese University of Hong Kong 2014
- 25 Моменты в OpenCV // <http://recog.ru/> URL: <http://recog.ru/library/opencv/opencvmoments.pdf> (дата обращения: 20.10.2018).
- 26 C. Olson and D. Huttenlocher. Automatic target recognition by matching oriented edge pixels. IEEE T. on IP, 6(1):103–113, 1997.
- 27 Reinbacher, C., Ruether, M., Bischof, H.: Pose estimation of know objects by efficient silhouette matching. In: ICPR (2010)
- 28 D. Gavrila and V. Philomin. Real-time object detection for smart vehicles. In IEEE ICCV, Vancouver, 1999.
- 29 S. Hinterstoisser, V. Lepetit, S. Ilic, P. Fua, and N. Navab. Dominant orientation templates for real-time detection of texture-less objects. In IEEE CVPR, San Francisco, 2010.
- 30 S. Holzer, S. Hinterstoisser, S. Ilic, and N. Navab. Distance transform templates for object detection and pose estimation. In IEEE CVPR, Miami, 2009.
- 31 Imperoli M., Pretto A.: D2CO: Fast and Robust Registration of 3D Textureless Objects using the Directional Chamfer Distance. In: ICVS, pp. 316-328 (2015)
- 32 Imperoli M., Pretto A.: D2CO: Active Detection and Localization of Textureless Objects in Cluttered Environments. In: arXiv preprint arXiv:1603.07022. 2016
- 33 Ross Girshick. Fast R-CNN. In IEEE International Conference on Computer Vision (ICCV), pages 1440–1448, 2015
- 34 Общая формулировка задачи внешней калибровки камеры // <http://masters.donntu.org/> URL: <http://masters.donntu.org/2012/iii/chigarev/library/article5.html> (дата обращения: 16.11.2018).



- 35 Yinqiang Zheng, Yubin Kuang, Shigeki Sugimoto Revisiting the PnP Problem: A Fast, General and Optimal Solution // ICCV2013. Sydney, NSW, Australia: 2014.
- 36 Pribyl, B.; Zemčík, P.; Cadík, M. Absolute Pose Estimation from Line Correspondences using Direct Linear Transformation. Comput. Vis. Image Underst. 2017, 161, 130–144.
- 37 C. Xu, L. Zhang, L. Cheng, and R. Koch, “Pose Estimation from Line Correspondences: A Complete Analysis and a Series of Solutions,” IEEE TPAMI, vol. 39, no. 6, pp. 1209–1222, 2017.
- 38 Ulrich M., Wiedemann C., Steger C.: CAD-based recognition of 3D objects in monocular images. In: ICRA, pp. 1191-1198 (2009)
- 39 Метод быстрой корреляции с использованием тернарных шаблонов при распознавании объектов на изображениях / Глумов Н.И., Мясников Е.В., Копенков В.Н., Чичева М.А. // Компьютерная оптика. – 2008. – Т. 32, № 3. – С. 277-282.
- 40 Детектирование объекта на изображении и определение его смещения на двух различных изображениях / Д.С. Азаренко // Искусственный интеллект. — 2013. — № 3. — С. 90–97.
- 41 Salvi J., Pages J., Batlle J. Pattern codification strategies in structured light systems // Pattern Recognition. 2004. Vol. 37(4). P. 827–849.
- 42 Hodan, T., Haluza, P., Obdr̥z̥alek, S., Matas, J., Lourakis, M., Zabulis, X.: T-less: An rgb-d dataset for 6d pose estimation of texture-less objects. In: IEEE Winter Conference on Applications of Computer Vision (WACV). pp. 880–888. IEEE (2017)
- 43 Geng J. Structured-light 3d surface imaging: a tutorial // Advances in Optics and Photonics. 2011. Vol. 3. P. 128–160.
- 44 Capturing 3D Images with Time-of-Flight Camera Technology // <https://www.allaboutcircuits.com/articles/capturing-3d-images-with-tof-camera-technology/> (дата обращения: 20.11.2018).

- 45     Времяпролетные камеры: 2D- и 3D-изображения за один кадр // <http://secuteck.ru/> URL: <http://secuteck.ru/articles2/videonabl/vremyaproletnye-kamery-2d-i-3d-izobrazheniya-za-odin-kadr> (дата обращения: 20.11.2018).
- 46     Time-of-flight camera // <https://en.wikipedia.org/> URL: [https://en.wikipedia.org/wiki/Time-of-flight\\_camera](https://en.wikipedia.org/wiki/Time-of-flight_camera) (дата обращения: 20.11.2018).
- 47     Программная реализация размытия по Гауссу // <http://rsync.altlinux.ru/> URL: <http://rsync.altlinux.ru/pub/people/at/gblur.pdf> (дата обращения: 20.11.2018).
- 48     I.T. Young and L.J. van Vliet, Recursive implementation of the Gaussian filter, Signal Processing, vol. 44, no. 2, 1995, 139-151.
- 49     L.J. van Vliet, I.T. Young, and P.W. Verbeek, Recursive Gaussian derivative filters (reprint of: A.K. Jain, S. Venkatesh, B.C. Lovell (eds.), ICPR'98, Proc. 14th Int. Conference on Pattern Recognition (Brisbane, Aug. 16-20), IEEE Computer Society Press, Los Alamitos, 1998, 509-514), in: A.M. Vossepoel, F.M. Vos (eds.), Fourth Quinquennial Review 1996-2001 Dutch Society for Pattern Recognition and Image Processing, NVPHBV, Delft, 2001, 317-322.
- 50     OpenCV шаг за шагом. Обработка изображения - морфологические преобразования // <http://robocraft.ru/> URL: <http://robocraft.ru/blog/computervision/319.html> (дата обращения: 10.11.2018).
- 51     Автоматизация работы в AutoCAD при помощи скриптов // <http://imolodec.com/> URL: <http://imolodec.com/ace-articles/avtomatizatsiya-raboty-v-autocad-pri-pomoshchi-skriptov> (дата обращения: 16.12.2018).
- 52     Generating Large Scale Image Datasets from 3D CAD Models // <https://github.com/> URL: <https://github.com/UMassLowell-Vision-Group/datasets> (дата обращения: 16.12.2018).
- 53     Baochen Sun, Xingchao Peng, Kate Saenko Generating Large Scale Image Datasets from 3D CAD Models. 2015
- 54     Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 3431–3440, 2015.

55 Georgios Pavlakos, Xiaowei Zhou, Aaron Chan, Konstantinos G Derpanis, and Kostas Daniilidis. 6-DOF object pose from semantic keypoints. IEEE International Conference on Robotics and Automation (ICRA), 2017