Гр. 43328/1 Хомутов Евгений

Обзор методов детектирования и определения положения заданного объекта на изображении

Задача определения положения – это задача определения относительного положения и ориентации камеры и объекта друг относительно друга в трехмерном пространстве. Она имеет множество практических применений в компьютерном зрении: 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 рассмотрено достаточно алгоритмов для определения положения объектов в пространстве по изображению. Проблема заключается в том, что не существует универсальных методов, которые могли бы быть применимы ко всем объектам, так как каждый объект обладает определенными особенностями. В данной работе рассматривается проблема определения положения слаботекстурированных объектов – это трехмерные объекты, не имеющие ярко выраженных особенностей. К таким объектам можно отнести: объекты, однородные по цвету; 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.

**Задача 1: Детектирование объектов на изображении**

Возможные подходы решения данной задачи:

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

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

* 1. **Цветовая фильтрация**

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

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 спектров 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 неидеальности 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)[1].

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) [2-4]. На рисунке 1 приведена матрица размытия 5х5, 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 | 054901 | 0,111345 | 054901 | 0,006581 |
| 0,000789 | 0,006581 | 0,013347 | 0,006581 | 0,000789 |

Рисунок 1 – 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 средние 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. В процессе бинаризации исходное полутоновое изображение, имеющее некое количество уровней яркости, преобразуется в черно-белое изображение, пиксели которого имеют только два значения – 0 и 1, в зависимости от того, к какому уровню яркости относится пиксель.

Четвертый 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 [5].

Метод на основе цветовой фильтрация достаточно прост в реализации, может быть применим 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.

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. **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, сравнения и поиска графических образов (объектов) по их контурам на изображении. Под контуром понимается кривая, которая описывает границу объекта на изображении, граница объекта - это места резкого изменения градиента яркости изображения. Для поиска контуров наиболее распространенными являются алгоритмы Собеля и Кэнни [[6]](https://moluch.ru/archive/193/48447/). Предполагается, что контур содержит достаточно информации о форме объекта, при этом точки внутри контура не учитываются. Рассмотрение и анализ только контуров на изображении существенно 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:

* При одинаково яркости с фоном объект может не иметь чёткой границы, или может быть зашумлен помехами, что приводит к невозможности выделения контура;
* Окклюзия и группировка объектов приводит к выделению контура, не соответствующему границе объекта.

Таким образом, контурный анализ имеет довольно слабую устойчивость к помехам, и любое нарушение целостности контура или плохая видимость объекта приводят либо к невозможности детектирования, либо к ложным срабатываниям. Однако простота и быстродействие контурного анализа, позволяют вполне успешно применять данный подход при условии наличия чётко выраженного объекта на контрастном фоне и отсутствии помех [7, 8].

Для поставленной задачи применение контурного анализа усложняется тем, что объект имеет сложную форму горловины банки и фаску у основания, вследствие чего получить набор шаблонов контуров объекта для получения хороших результатов является достаточно трудоемкой задачей. Также, метод может быть применим, но не всегда, так как форма объекта может иметь незначительные отклонения от идеальной (боковые повреждения банки, вогнутость внутрь вследствие механический ее повреждений).

* 1. **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.

Поиск шаблона производится путем последовательного перемещения его на один пиксель за раз по анализируемому изображению, и оценкой схожести каждой новой области с шаблоном. По результатам проверки выбирается та область, которая имеет наивысший коэффициент совпадения. То есть метод ищет точные совпадения пикселей шаблона с пикселями на изображения.

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.

Однако стоит понимать, что данный метод не позволяет с уверенностью сказать был ли найден исходный объект, поскольку это вероятностная характеристика, зависящая от масштаба, углов обзора, поворотов картинки и наличия физических помех. Также возможны ложные срабатывания алгоритма, когда искомого объекта на самом деле нет, но имеются какие-то общие детали у шаблона и области на тестируемом изображении. Конечно, подобной ситуации можно избежать путём проверки значения коэффициента совпадения (чтобы он не был меньше некоторого граничного предела), однако это не всегда будет работать должным образом ввиду описанных выше причин [7, 8]. 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. **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 [9], для поиска небольших связных областей подойдет метод Blob detection [10].

Подобные точки каждый алгоритм определяет по своему. Для нахождения ключевых точек на изображениях и последующего их сравнения используются три составляющие:

* Детектор (feature detector) — осуществляет поиск ключевых точек на изображении;
* Дескриптор (descriptor extractor) — производит описание найденных ключевых точек, оценивая их позиции через описание окружающих областей;
* Матчер (matcher) — осуществляет построение соответствий между двумя наборами точек изображений.

Библиотека OpenCV имеет достаточно широкий набор детекторов, дескрипторов и матчеров [[](https://arxiv.org/ftp/arxiv/papers/1710/1710.02726.pdf)11-13]. При этом имеются возможности различного сочетания их друг с другом. Все они отличаются по скорости работы, числу выделяемых точек, а также устойчивости к трансформациям изображения: вращениям, сменам углов обзора, изменениям масштаба [8].

В отличие от сопоставления с шаблоном и контурного анализа, алгоритмы поиска ключевых точек более устойчивы к помехам, трансформациям и позволяют находить объекты даже при наличии физических помех. Для достижения как можно более качественного быстрого обнаружения объекта, он должен обладать достаточно большим числом уникальных (стабильных) ключевых точек, которые детектор быстро может выделить на изображении и сопоставить с имеющимся шаблонным набором. Для этого необходимо использовать как можно более быстрый детектор, дескриптор и матчер. Выбор данных трёх компонентов осуществляется путём проведения экспериментов с замером скорости работы и оценкой инвариантности относительно различных трансформаций 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. К недостаткам можно отнести, что данные методы не дадут хорошего результата для объектов со слабо выраженными особенностями (тела вращения), а также к однородным по цвету объектам.

* 1. **Алгоритмы сегментации**

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. [14, 15]

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: белые объекты и черный фон. Пиксели, яркость которых превышает пороговое значение, относятся к классу объект, все остальные пиксели относятся к классу фон.

Метод Оцу использует гистограмму изображения для расчета порогового значения. Рассмотрим простейший случай, когда на изображении один объект. По гистограмме человек легко видит два разделяющихся класса (рисунок 1.6.1).

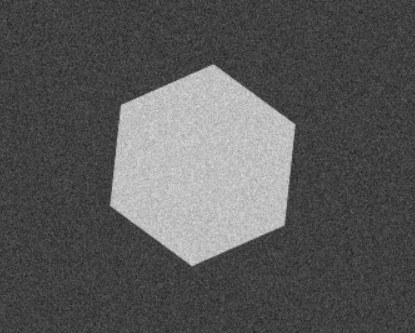
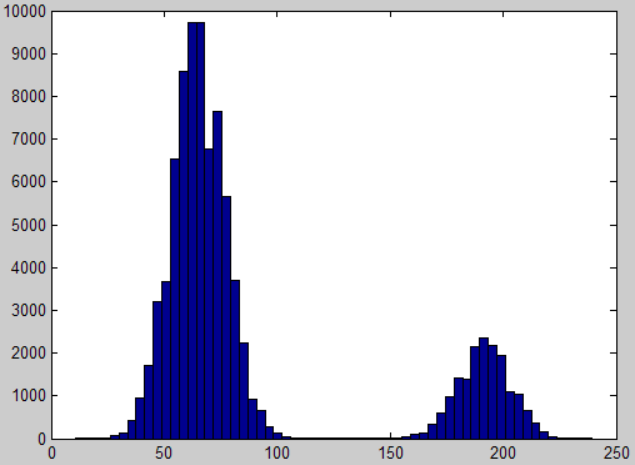
 

Рисунок 1.6.1 – Пример изображения с объектом и гистограмма изображения.

Суть метода заключается в том, чтобы выставить порог между классами таким образом, чтобы каждый из них был как можно более плотным (рисунок 1.6.2).

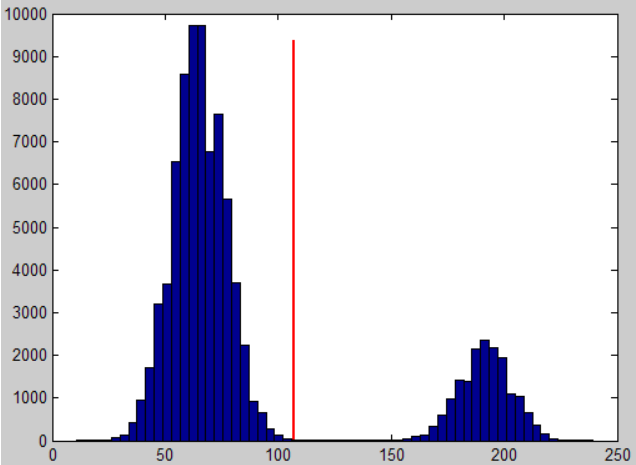
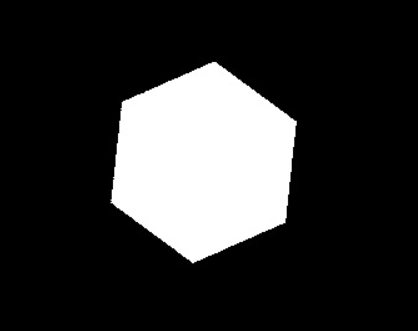


Рисунок 1.6.2 – Результат применения метода Оцу.

Достоинствами этого метода являются:

* Простота реализации;
* Метод хорошо адаптируется к различного рода изображения, выбирая наиболее оптимальный порог;
* Быстрое время выполнения. Требуется O(N) операций, где N — количество пикселей в изображении;

К недостаткам можно отнести, что сама по себе пороговая бинаризация чувствительна к неравномерной яркости изображения. Решением такой проблемы может быть введение локальных порогов, вместо одного глобального [16]. Также метод неприменим к сценам с окклюзией.

Для данной задачи метод Оцу имеет существенное ограничение: объект должен быть на пестром фоне, т.е. существенно отличаться от него, что в данной задачи не гарантируется.

* 1. **Машинное обучение**

Если у нас стоит задача выделение красной алюминиевой банки на изображении, то по особым точкам метод будет искать одну конкретную фотографию. Ту фотографию, на которой были выделены особые точки, банки другого цвета будут выделятся хуже, потому как им, скорее всего, соответствуют другие наборы особых точек. Результаты могут быть ещё хуже если изменить ракурс съемки.

Для решения этих проблем уже необходимы методы машинного обучения и не одна картинка с объектом, а учебные наборы из сотен (а в некоторых случаях - сотен тысяч) разных картинок с изображением объекта в разных условиях[7].

Методы машинного обучения позволяют находить объекты, совпадающие с заданными эталонами. Основная особенность методов – обучение процессу подбора положения объекта, а не процедуре сравнения [17-20].

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 [19].

* 1. **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 Хью-моментов.

**Хью-моменты** (моменты изображения) – это инвариантные моменты к масштабированию и повороту объектах[21].

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.

**Задача 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 [22, 23] или форму [24-28] для представления объекта. Другие основаны на сопоставлении локальных особенностей, описываемых дескрипторами, например такими как HOG (Histogram of Oriented Gradients) [29, 30], SIFT (Scale-Invariant Feature Transform) [31]. Также существуют подходы основе глубоких нейронных сетей для определения положения объектов. В [56] предложен метод на основе глубокого обучения для итеративного определения положения объекта. В [57] для определения положения объекта на сцене с сильной окклюзией применяются сверточные 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:

* Определение положения объекта по его модели;
* PnP;
* Структурированная подсветка;
* Машинное обучение (иерархический поиск).
* Времяпролетная камера;
  1. **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[32]. В [33] 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 получение 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 [34].

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 искомых объектов известна заранее. На основании этого рассматривается модель объекта, представляющая собой набор черт и их взаимное расположение.

* 1. **PnP, PnL**

Такими чертами могут выступать определенные точки (особые точки, вершины), грани и углы объекта.

PnP (Perspective n Points) – это задача определения положения объекта по наблюдаемым n-соответствиям особых точек модели объекта и его аналогом на изображении.

Определение положения по особым точкам заключается в следующем: на модели объекта и на тестируемом изображении определяются особые точки, далее, особые точки с изображения камеры сопоставляются с точками модели. И по известным зависимостям между особыми точками на модели объекта можно определить положение объекта в пространстве относительно камеры.

Условием для решения задачи является задание n-соответствий между точками изображения и точками модели объекта, а также известные заранее взаимные расположения точек модели объекта.

P1P и P2P проблемы имеют бесконечное множество решений и поэтому в литературе не рассматриваются. Наибольший интерес представляет P3P проблема, так как для решения этой задачи достаточно лишь тех соответствий между точками изображения и модели. Интерес к данной задаче связан с ее минимальными требованиями к числу найденных соответствий. Дело в том, что выделение черт на изображении и установление соответствия между ними и их проекциями отнимает значительное процессорное время. Следовательно, для приложений, ориентированных на работу в режиме реального времени, требуется использовать минимальное число черт. К сожалению, при точном решении Р3Р проблемы необходимо решать уравнения четвертой степени [35]. 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.

Для повышения скорости определения положения PnP может быть оптимизирован, путем применения кватернионов [36].

В связи с этим иногда рассматриваются варианты задачи для четырех, пяти и шести точек. Шести точек достаточно для задания системы линейных уравнений, имеющих единственное решение, поэтому PnP проблемы для n > 6 не рассматриваются [35].

PnP метод имеет достаточно хорошие результаты для объектов, обладающих большим числом особенностей, которые существенным образом отличаются от фона. В идеальном случае особенностей должно быть столько, чтоб в любом положении камеры относительно объекта можно было сопоставить 6 особых точек объекта. так как чем больше удается сопоставить особенностей, тем меньше требуется ресурсов для получения одного верного решения. К недостаткам метода можно отнести трудоемкость получения решения при малом числе n-соответствий и невозможности применения для объектов со слабо выраженными особенностями, или с малым числом особенностей.

Для данной задачи метод скорее всего не применим, так как выбранный объект является однородным по цвету и телом вращения, т.е. имеет слабо выраженные особенности.

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

PnL (Perspective n Lines) – это задача, аналогичная задаче PnP, только в качестве черт объекта используются линии (грани) 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 четкие прямые линии перехода (грани) или если для контура объекта характеры прямые линии, 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 детектированипя особых точек требуется поиск черт (особенностей объекта), коррелирующих с предварительно заданными шаблонами (соответствие дескрипторов особых точек), а для выделения линий достаточно вычислить лишь градиент интенсивности (яркости) пикселей изображения.

(В [44] 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. (В методах DLT – практически одна математика, сложно словами описать, что в них происходит, стоит их описывать, т.е. например привести часть статьи?). В [45] предложен новый подход для решения PnL задачи на основе линейной композиции PnL.)

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. Подход на на основе PnL наоборот, подходит для сцен со слабо выраженной текстурой. Кроме того, линейные особенности более устойчивы к частичным окклюзиям, чем точечные.

* 1. **Структурированная подсветка**

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[37, 38]. 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 [37, 38]. Поскольку шаблон (картина) закодирован, то можно легко найти соответствия между точками изображения и точками проецируемого шаблона.

Данный метод хорошо подходит для объектов с не ярко выраженной текстурой, но имеет существенный недостаток – метод не применим к объектам с ярко выраженной текстурой, а также к сценам, содержащим множество текстурированных объектов, поэтому для решения данной задачи метод не применим.

* 1. **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).

В [32] (было 40, 32 и 40 – дублировались) представлен метод шаблонного сравнения для определения положения объектов относительно камеры с применением CAD-систем.

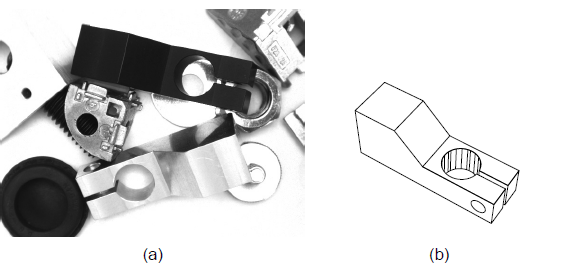


Рисунок 2.6.1 – Изображение двух металлических зажимов разного цвета. Модель зажима в САПР (b), показанного на (a), которая служит в качестве входных данных. Для визуализации скрытые края, а также края между копланарными гранями не отображаются.

Строится дерево определения положения: есть 3д модель объекта, 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.

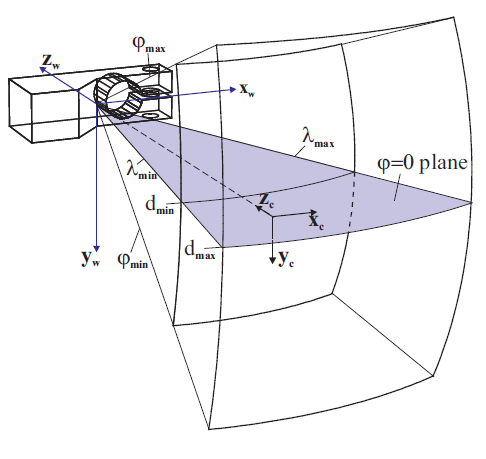


Рисунок 2.6.2 – Сферическая система координат сферы взгляда на модель объекта. Положение мировой системы координат (xw, yw, zw) относительно системы координат камеры (xc, yc, zc) может быть описано долготой (λ), широтой (φ) и расстоянием (d). Диапазон положений описывается минимальными и максимальными значениями λ, φ и d.

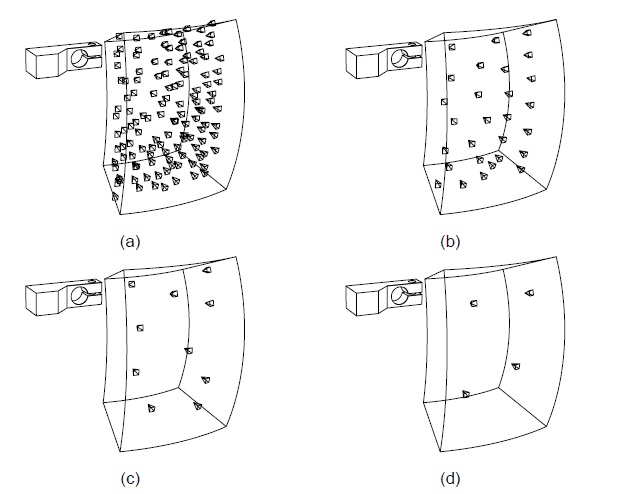


Рисунок 2.6.3 - Результирующие представления модели на уровне 1 (а) пирамиды, уровне 2 (b), уровне 3 (c) и уровне 4 (d). Представления визуализируются маленькими квадратными пирамидами, которые представляют камеры.

На рис.3(а) для всех представлений на нижнем уровне пирамиды визуализируются соответствующие камеры, которые получаются при применении описанного метода к диапазону позиций, показанного на рис.2.

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 рис 3.(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.

Метод имеет следующие достоинства:

* Робастно устойчив;
* Быстрый (определение положения за несколько сотен 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. **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. С помощью ToF можно получить подробную 3D-информацию об объекте. Для освящения 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 [41].

Системы камер ToF состоят из сенсора, процессора обработки изображения и модулирующего источника света [42]. 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 [41].

Достоинства:

* Компактность [43];
* Простота алгоритмов определения расстояния. ToF вычисляет расстояние практически напрямую с сенсора. В результате используется лишь небольшая вычислительная мощность [43].
* Скорость. Поскольку камеры работают со скоростью до 160 кадров в секунду, они идеально подходят для использования в приложениях реального времени [43].

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 [42].
* 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. включении камер по очереди [43].
* 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 [42, 43].

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

* + 1. Сравнение модельного изображения с реальным

Модельное изображение с реальным можно сравнивать несколькими способами:

* По количество 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-то признаков, которые описывают картинку).
  1. **Определение положения на основе RGB-D**

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 [9] трехмерная модель объекта отображается в шаблон нормалей поверхности объекта, а затем этот шаблон сопоставляется с нормалями, вычисленными из облака точек.

В [1, 2, 17] каждый пиксель на объекте во входном изображении регрессируют до 3D-координаты этого пикселя на 3D-модели. Когда доступна карта глубины, регрессия трехмерных координат устанавливает соответствия между точками трехмерной сцены и точками трехмерной модели, из которых положение объекта может быть вычислено путем решения задачи наименьших квадратов. Для уточнения положения широко используется алгоритм итеративной ближайшей точки (ICP) [9, 17, 30]. Однако ICP чувствителен к первоначальной оценке и может сходиться к локальным минимумам [56].

* 1. **Определение положения на основе нейронных сетей**

В [57] предлагается новая сверточная нейронная сеть для определения положения объекта, названная PoseCNN. Основная идея PoseCNN состоит в том, чтобы разделить задачу определения положения на различные компоненты, которые позволят сети точно моделировать зависимости и независимости между ними. В частности, нейронная сеть PoseCNN решает три связные задачи, показанные на рисунке 1.

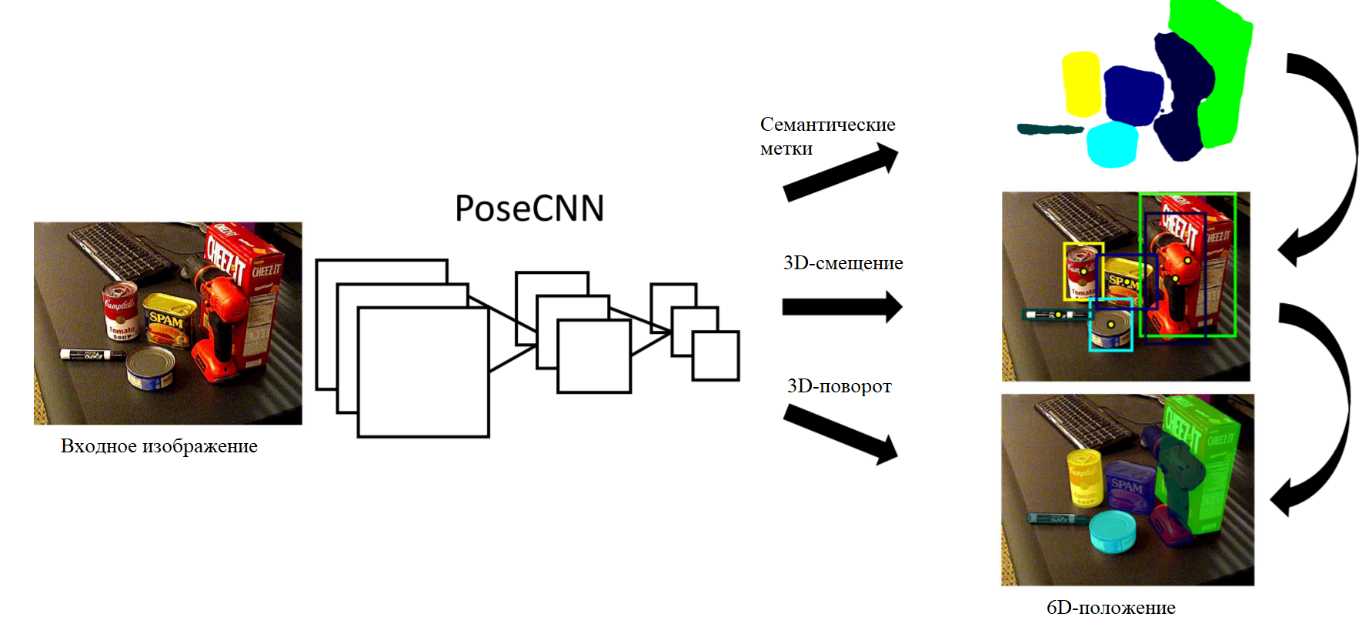


Рисунок 1 – Предложенная новая сеть PoseCNN для определения положения объекта, сеть обучается выполнять три задачи: семантическую маркировку, определение трехмерного смещения и регрессию трехмерного вращения.

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 объектов, 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 сверточных элементов, извлеченных внутри ограничительной рамки объекта в представление в кватернионах. В статье показывается, что PoseCNN 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 сверточная архитектура нейронной сети, которая внутренне разделяет определение R и T. Рисунок 2 иллюстрирует архитектуру сети PoseCNN для определения 6D координат объекта. Сеть состоит из двух ступеней. Первая ступень состоит из 13 сверточных 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 2, 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 сверточных слоя. Затем, она удваивает разрешение карты объекта, размером 1/16 с помощью транспонированного сверточного слоя (ТСС). После этого карты признаков суммируются и далее с помощью еще одного транспонированного сверточного слоя разрешение суммированной карты увеличивается в 8 раз до размера исходного изображения. Далее, сверточный 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 (a 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 [58].

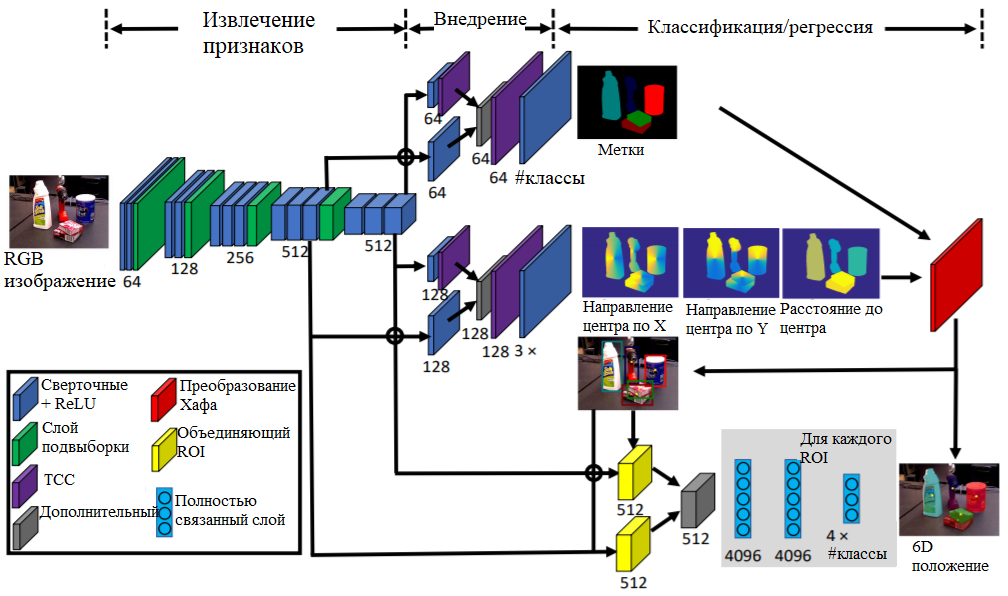


Рисунок 2 – Архитектура PoseCNN для определения положения объекта

Определение трехмерного смещения

Как показано на рисунке 3, T = (Tx, Ty, Tz)T – есть координаты начала системы координат объекта в системе координат камеры. Наивным способом оценки Т является прямая регрессия элементов изображения в Т. 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 Т путем локализации центра 2D-объекта на изображении и определения расстояния до объекта от камеры. Предположим, что проекция Т на изображении с = (сх, су)Т.

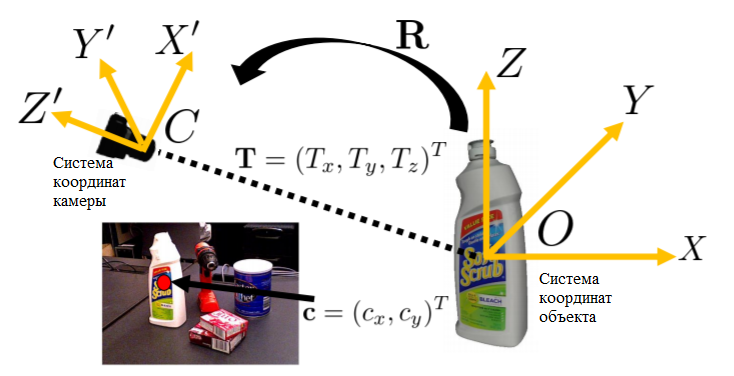
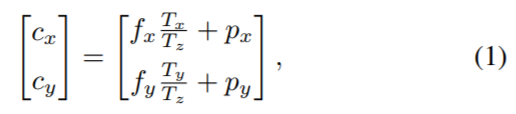


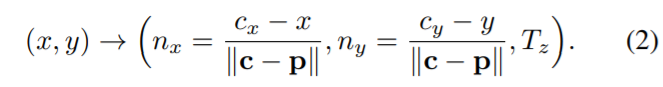
Рисунок 3 - Иллюстрация системы координат объекта и системы координат камеры. Трехмерное смещение можно определить путем локализации 2D-центра объекта и оценки расстояния до 3D-центра от камеры.

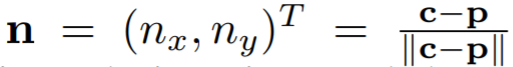
Если сеть может локализовать ***с*** на изображении и определить глубину ТZ, можно восстановить Тх и Ту согласно следующему уравнению для проекции, предполагая, что у нас камера Обскура:



где fx и fy – фокусные расстояния камеры, а (px, py) – центр изображения. Если начало системы отсчета объекта О является центром тяжести, с – называем 2D-центром объекта.

Простой способ локализации центра 2D-объекта состоит в непосредственном обнаружении центральной точки, как в существующих методах обнаружения ключевых точек [59]. 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:



Заметим, что вместо прямой регрессии к вектору c-p, спроектируем сеть так, чтобы она регрессировала к вектору единичной длины , т.е. в направлении 2D-центра, которое не зависит от масштаба и поэтому сеть легче обучать. Центральная ветвь регрессии предложенной сети (рисунок 2) 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\*n, с n-числом классов. При обучении сглаженная функция потерь L1 применяется для регрессии как в [60].

Для того чтобы найти 2D-центр ***с*** 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 сеть (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 немаксимальное поддавление 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, TZ, просто вычисляется как среднее значение глубин предсказанных инлайерами («не выбросами», хорошие точки). В конце, используя формулу (1) можно оценить трехмерное смещение ***Т.*** 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

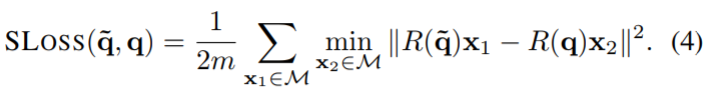
В нижней части рисунка 2 показана ветвь регрессии трехмерного вращения. Используя ограничивающие прямоугольники объектов, предсказанные из слоя преобразования Хафа, используются два слоя объединения ROI для «объединения и обрезки»[60] визуальных признаков, созданных на первой ступени сети для регрессии 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\*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:



где ***М*** обозначает набор точек 3D-модели и ***m –*** количество этих точек. и R(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:



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.

**Результат**

В результате проведенного обзора, для детектирования выбранного объекта (однородной алюминиевой банки) применимы методы цветовой фильтрации и сегментации изображения, для определения положения – определение положения объекта по его изображению с применением машинного обучения для сопоставления с шаблонами.

Библиография

[1] Гистограмма и гистограммная обработка изображений http://wiki.technicalvision.ru/index.php/Гистограмма\_и\_гистограммная\_обработка\_изображений

[2] Программная реализация размытия по Гауссу

http://rsync.altlinux.ru/pub/people/at/gblur.pdf

[3] I.T. Young and L.J. van Vliet, Recursive implementation of the Gaussian filter, Signal Processing, vol. 44, no. 2, 1995, 139-151.

[4] 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.

[5] <http://robocraft.ru/blog/computervision/319.html>

[6] 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).

[7] О задаче поиска объекта на изображении. Часть 1: Базовые методы.

http://mechanoid.kiev.ua/cv-image-detector.html

[8] Анализ алгоритмов компьютерного зрения

<https://arealidea.ru/articles/stati-i-publikatsii/analiz-algoritmov-kompyuternogo-zreniya-poiska-obektov-i-sravneniya-izobrazheniy/>

[9] OpenCV documentation: Harris Corner Detection

<http://docs.opencv.org/3.0-beta/doc/py_tutorials/py_feature2d/py_features_harris/py_features_harris.html>

[10] Wikipedia: Blob\_detection

<http://en.wikipedia.org/wiki/Blob_detection>

[11] Image Matching Using SIFT, SURF, BRIEF and ORB: Performance Comparison for Distorted Images

<https://arxiv.org/ftp/arxiv/papers/1710/1710.02726.pdf>

[12] Object Recognition from Local Scale-Invariant Features

<https://www.cs.ubc.ca/~lowe/papers/iccv99.pdf>

[13] Эффективные дескрипторы для локального описания изображений <http://en.cs.msu.ru/sites/cmc/files/docs/lukianitsa.pdf>

[14] D. Comaniciu, P. Meer Mean shift analysis and applications, IEEE International Conference on Computer Vision, 1999, vol. 2, pp. 1197.

[15]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.

[16] 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.

[17] Large Scale Visual Recognition Challenge 2014 (ILSVRC2014) // URL:<http://image-net.org/challenges/LSVRC/2014/>

[18] Review of Deep Learning Algorithms for Object Detection. // URL:<https://medium.com/comet-app/review-of-deep-learning-algorithms-for-object-detection-c1f3d437b852>

[19] О задаче поиска объекта на изображении. Часть 2: Применение методов машинного обучения.

<http://mechanoid.kiev.ua/cv-image-detector2.html>

[20] Deep Learning in Object Recognition, Detection and Segmentation, Xiaogang Wand, The Chinese University of Hong Kong 2014

[21] Моменты в OpenCV

http://recog.ru/library/opencv/opencvmoments.pdf

[22] C. Olson and D. Huttenlocher. Automatic target recognition by matching oriented edge pixels. IEEE T. on IP, 6(1):103–113, 1997.

[23] Reinbacher, C., Ruether, M., Bischof, H.: Pose estimation of know objects by efficient silhouette matching. In: ICPR (2010)

[24] D. Gavrila and V. Philomin. Real-time object detection for smart vehicles. In IEEE ICCV, Vancouver, 1999.

[25] 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.

[26] S. Holzer, S. Hinterstoisser, S. Ilic, and N. Navab. Distance transform templates for object detection and pose estimation. In IEEE CVPR, Miami, 2009.

[27] Imperoli M., Pretto A.: D2CO: Fast and Robust Registration of 3D Textureless Objects using the Directional Chamfer Distance. In: ICVS, pp. 316-328 (2015)

[28] Imperoli M., Pretto A.: D2CO: Active Detection and Localization of Textureless Objects in Cluttered Environments. In: arXiv preprint arXiv:1603.07022.

[29] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In IEEE CVPR, pages 886–893, 2005.

[30] Про гистограмму ориентированных градиентов // URL: <https://www.learnopencv.com/histogram-of-oriented-gradients/> (2016)

[31] D. Lowe. Distinctive image features from scale-invariant keypoints. IJCV, 60(2):91–110, 2004.

[32] Ulrich M., Wiedemann C., Steger C.: CAD-based recognition of 3D objects in monocular images. In: ICRA, pp. 1191-1198 (2009)

[33] 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.

[34] Про определение положения объектов <http://dspace.nbuv.gov.ua/bitstream/handle/123456789/85076/11-Azarenko.pdf?sequence=1>

[35] Общая формулировка задачи внешней калибровки камеры <http://ict.informika.ru/ft/002404/num1krav.pdf>

[36] Пересмотр проблемы PnP: быстрое, общее и оптимальное решение

<http://www.maths.lth.se/vision/publdb/reports/pdf/zheng-kuang-etal-iiccvi-13.pdf>

[37] Salvi J., Pages J., Batlle J. Pattern codification strategies in structured light systems // Pattern Recognition. 2004. Vol. 37(4). Р. 827–849.

[38] Geng J. Structured-light 3d surface imaging: a tutorial // Advances in Optics and Photonics. 2011. Vol. 3. Р. 128–160.

[39] Методы восстановления трехмерной структуры объектов для многоканальных систем регистрации с использованием структурированной подсветки

https://docplayer.ru/55188788-Metody-vosstanovleniya-trehmernoy-struktury-obektov-dlya-mnogokanalnyh-sistem-registracii-s-ispolzovaniem-strukturirovannoy-podsvetki.html

[41] Capturing 3D Images with Time-of-Flight Camera Technology <https://www.allaboutcircuits.com/industry-articles/capturing-3d-images-with-tof-camera-technology/> (2018)

[42] Времяпролетные камеры: 2D- и 3D-изображения за один кадр <http://secuteck.ru/articles2/videonabl/vremyaproletnye-kamery-2d-i-3d-izobrazheniya-za-odin-kadr> (2016)

[43] Time-of-flight camera

<https://en.wikipedia.org/wiki/Time-of-flight_camera#Disadvantages> (2018)

[44] Pribyl, B.; Zemcík, P.; Cadík, M. Absolute Pose Estimation from Line Correspondences using Direct LinearTransformation. Comput. Vis. Image Underst. 2017, 161, 130–144.

[45] C. Xu, L. Zhang, L. Cheng, and R. Koch, “Pose Estimation from Line Correspondences: A CompleteAnalysis and a Series of Solutions,”IEEE TPAMI, vol. 39, no. 6, pp. 1209–1222, 2017.

[46] Padeleris, P., Zabulis, X., Argyros, A.A.: Head pose estimation on depth data based on Particle Swarm Optimization. In: IEEE CVPRW, pp. 42–49 (June 2012) – ПОСМОТРЕТЬ

[47] Malassiotis, S., Strintzis, M.G.: Robust real-time 3D head pose estimation from range data. Pattern Recognition 38, 1153–1165 (2005) – ПОСМОТРЕТЬ

[48] Mykhaylo, A., Stefan, R., Bernt, S.: Monocular 3D pose estimation and tracking by detection. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 623–630 (2010) – ПОСМОТРЕТЬ

[49] Yang, Y., Ramanan, D.: Articulated pose estimation using flexible mixtures of parts. In: CVPR (2011)

[50] Jarrett, K., et al.: What is the best multi-stage architecture for object recognition? In: 2009 IEEE 12th International Conference on Computer Vision. IEEE (2009)

[51] Bengio, Y.: Learning deep architectures for AI. Foundations and Trends in Machine Learning 2(1), 1–127 (2009)

[52] Hinterstoisser, S., Cagniart, C., Ilic, S., Sturm, P., Navab, N., Fua, P., Lepetit, V.: Gradient response maps for real-time detection of texture-less objects. PAMI 34(5), 876–888 (2012)

[52] Hinterstoisser, S., Lepetit, V., Ilic, S., Holzer, S., Bradski, G., Konolige, K., Navab, N.: Model based training, detection and pose estimation of texture-less 3D objects in heavily cluttered scenes. In: Lee, K.M., Matsushita, Y., Rehg, J.M., Hu, Z. (eds.) ACCV 2012, Part I. LNCS, vol. 7724, pp. 548–562. Springer, Heidelberg (2013)

[53] Liebelt, J., Schmid, C., Schertler, K.: Viewpoint-independent object class detection using 3D feature maps. In: CVPR (2008)

[54] Liu, M., Tuzel, O., Veeraraghavan, A., Taguchi, Y., Marks, T.K., Chellappa, R.: Fast object localization and pose estimation in heavy clutter for robotic bin-picking. International Journal on Robotic Research 31(8) (2012)

[55] Hodan, T., Haluza, P., Obdrˇz´alek, 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)

[56] Y. Li, G. Wang, X. Ji, Y. Xiang, and D. Fox, ‘‘DeepIM: Deep iter-ative matching for 6D pose estimation,’’ inProc. Eur. Conf. Comput.Vis. (ECCV), 2018, pp. 695–711.

[57] Y. Xiang, T. Schmidt, V. Narayanan, D. Fox: Pose CNN: A Convolutional Neural Network for 6D Object Pose Estimation in Cluttered Scenes

[58] 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.

[59] 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

[60] Ross Girshick. Fast R-CNN. In IEEE International Conference on Computer Vision (ICCV), pages 1440–1448, 2015