

A.G. Leskov, S.D. Moroshkin
COMPUTER VISION SYSTEM FOR OBJECTS POSE ESTIMATION

Dmitrov branch of BMSTU
c8.df.mgtu@gmail.com

А.Г. Лесков, С.Д. Морошкин
СИСТЕМА ТЕХНИЧЕСКОГО ЗРЕНИЯ ДЛЯ ОПРЕДЕЛЕНИЯ РАСПОЛОЖЕНИЯ ОБЪЕКТОВ

ДФ МГТУ им. Н.Э. Баумана, Дмитров
c8.df.mgtu@gmail.com

Введение

Системы технического зрения (СТЗ) являются важным компонентом робототехнических систем. В манипуляционных системах СТЗ используются для обзора рабочей зоны, классификации и детектирования объектов манипулирования, а также оценки их местоположения.

Существует широкий спектр алгоритмов и технологий, лежащих в основе систем технического зрения. Например, в качестве датчиков могут использоваться 2D-камеры [1], сенсоры глубины со структурной подсветкой или лазерные дальномеры [2].

Предлагаемая система технического зрения предназначена для формирования математического представления об объектах манипулирования (ОМ), находящихся в рабочей зоне, пространственном положении этих объектов, а также для формирования карты препятствий.

В основе предлагаемой СТЗ лежат сенсоры глубины семейства PrimeSense - это датчики ASUS Xtion Pro Live и Microsoft Xbox Kinect 360. Они позволяют получать не только плоское цветное изображение, но также строить облако точек (ОТ) видимой части сцены. Облако точек представляет собой набор точек, определяемых пространственными координатами x , y и z в системе координат (СК) сенсора, а также значениями цветовых каналов в кодировке RGB.

Программно система технического зрения состоит из драйверов сенсоров, отвечающих за прием данных и формирование облаков точек, программ, реализующих алгоритмы режимов работы системы, а также базы данных (БД), в которой хранятся математические модели объектов.

п.п Алгоритмы работы системы

Основной задачей представленной СТЗ является определение пространственного расположения объектов манипулирования относительно системы координат сенсора (камеры). Под объектом манипулирования будем понимать объекты, для которых имеется трехмерная модель, а габаритные размеры не превышают 300 мм. Тогда работу системы можно разделить на три составляющих:

1. Формирование математической модели ОМ (обучение).
2. Выбор оператором ОМ в рабочей зоне.
3. Оценка пространственного положения объекта (детектирование).

Формирования математических моделей ОМ (обучения)

В режиме обучения выполняется формирование системой математической модели объекта, как некоторого упорядоченного набора данных, определяющего внутреннее представление объекта в СТЗ, и использующееся в дальнейшем для оценки расположения объекта.

Для имеющейся трехмерной модели формируются ее локальные виды. Локальный вид представляет собой такое облако точек, которое видел бы сенсор СТЗ, «смотря» на объект с определенного ракурса. Для их получения вокруг модели строится сфера, которая разбивается на n равных треугольников. Глубина разбивки n устанавливается оператором, в представленной СТЗ $n = 80$. В центр каждого треугольника устанавливается виртуальная камера, ее «взгляд» направляется в начало СК объекта и формируется локальный вид объекта. Облако точек, представляющее локальный вид, а также координаты установки камеры относительно СК объекта записываются в базу данных. Так формируется набор из n локальных видов модели объекта.

После производится расчет OUR-CVFH-дескрипторов [3] для каждого из локальных видов. OUR-CVFH дескриптор – алгоритм, описывающий облако точек локального вида гистограммой, состоящей из 308 элементов. Первые 139 элементов кодируют распределение направлений нормалей, построенных для каждой точки из облака. Следующие 104 элемента кодируют распределение точек по квадрантам относительно специальной системы координат, называемой SGURF - Semi-Global Unique Reference Frame (полуглобальная уникальная система координат), положение которой также

рассчитывается в дескрипторе. Последние 64 элемента массива кодируют направление «взгляда» камеры на облако точек. На рисунке 1 справа показан локальный вид объекта, а также расположенная на нем СК SGURF. Слева показана гистограмма OUR-CVFH дескриптора. Красным обозначены элементы, кодирующие направления нормалей, зеленым – кодирующие распределение точек, а синим – направление «взгляда» камеры.

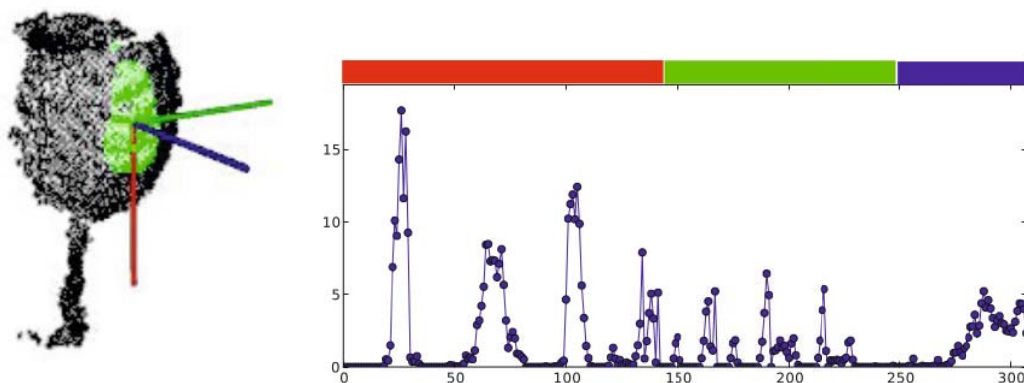


Рисунок 1 – OUR-CVFH-гистограмма для «гладкой» области облака точек

Каждой гистограмме в соответствие ставится матрица 4x4 с координатами установки камеры и матрица 4x4 расположения СК SGURF. Этот набор данных записывается в базу данных под определенным номером, указывающим на номер локального вида.

Сформированная таким образом математическая модель объекта представляет собой упорядоченный набор данных, состоящий из OUR-CVFH-гистограмм и прикрепленных к каждой гистограмме двух матриц 4x4, определяющих координаты установки камеры и расположение СК SGURF.

Выбор оператором ОМ в рабочей зоне

Используемые в СТЗ сенсоры позволяют получать цветное облако точек видимой части сцены. На рисунке 2 представлен вид системы визуализации с облаком точек сцены. На нем видно, что в поле зрения попадают различные объекты манипулирования, а также стол и объекты окружения.

Для определения координат ОМ необходимо решить две задачи: классифицировать объект и выделить из облака точек те точки, которые принадлежат объекту. На данном этапе первая задача решается оператором путем указания, с каким именно объектом будет происходить работа. Вторая задача решается при помощи элементов графического интерфейса.

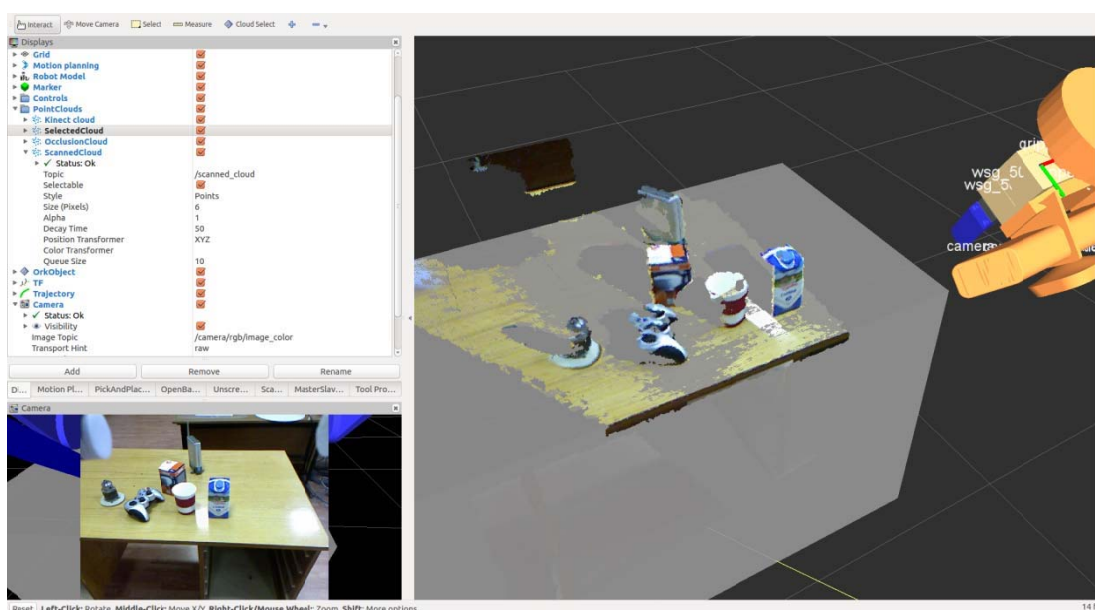


Рисунок 2 – Облако точек сцены

Оператор при помощи мыши указывает точку на поверхности нужного объекта в представленном в окне облаке точек, тем самым определяя двумерные координаты точки на экране. Для определения координат указанной точки в трехмерном облаке точек строится луч, проходящий через указанную пользователем двумерную точку и перпендикулярный плоскости виртуальной камеры. Та точка из трехмерного облака точек, расстояние от которой до луча будет минимальным, считается выбранной оператором точке ОМ.

После того, как на облаке точек определена одна точка, принадлежащая ОМ, системой выделяется кластер точек, принадлежащих объекту. Для этого вокруг вычисленной точки строится куб, размер которого зависит от размеров ОМ и задается при помощи интерфейса пользователя. Все точки *исходного* облака точек, находящиеся вне куба, удаляются. Среди *оставшихся* определяются точки, принадлежащие самой нижней горизонтальной плоскости (обычно это плоскость стола). Затем эти точки удаляются. Если в результате удаления остался один кластер точек, то есть, расстояние между каждой из точек и ее ближайшим соседом не превышает 10мм, этот кластер считается кластером точек объекта. Если кластеров несколько, выбирается тот, к которому наиболее близка указанная оператором точка (на рисунке 3 отмечен зеленым).

После определения кластера объекта выделяются кластеры препятствий. Для этого из исходного облака точек удаляются все точки, лежащие в заданном радиусе от каждой точки ОТ объекта. Оставшиеся точки считаются точками препятствий (на рисунке 3 отмечены красным).

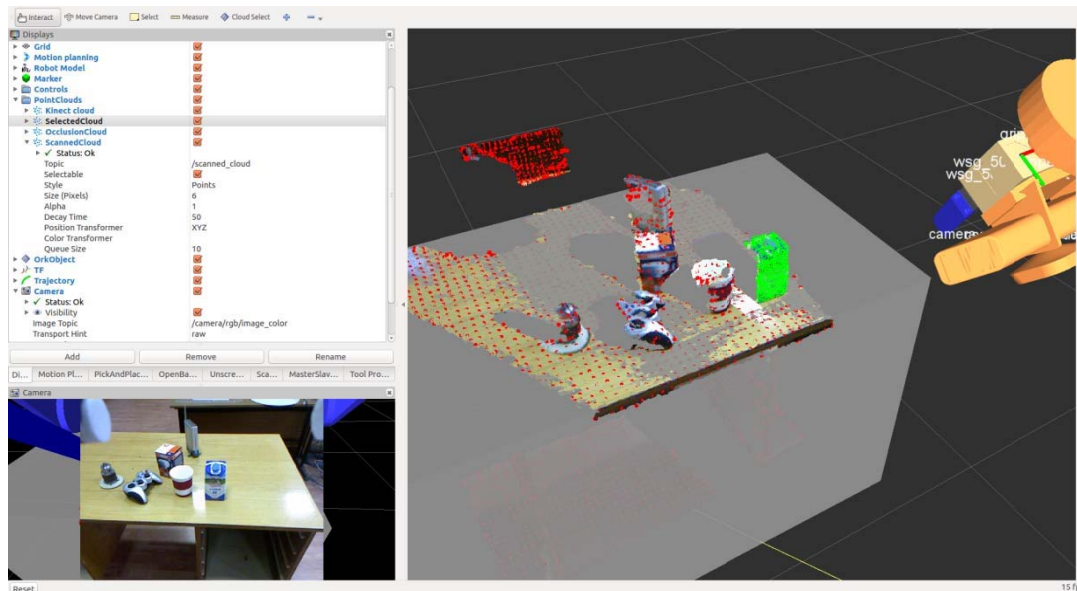


Рисунок 3 – Интерфейс выбора объекта на сцене

Результатом выбора объекта являются сформированные облака точек поверхностей, принадлежащих выбранному объекту манипулирования, а также других объектов на сцене, которые могут быть рассмотрены как препятствия.

Оценка расположения объекта

После выделения точек объекта из исходного ОТ можно определить расположение системы координат, связанной с трехмерной моделью объекта, относительно системы координат сенсора. Алгоритм работы следующий.

Для выделенного кластера точек (зеленый на рисунке 3) рассчитывается OUR-CVFH дескриптор, а также матрица расположения СК SGURF относительно СК датчика $T_{sensor, SGURF_0}^T$.

При помощи алгоритмов поиска ближайших соседей в сформированной ранее математической модели отыскивается набор из M гистограмм, максимально близких к гистограмме данного кластера (количество определяется оператором и обычно составляет 10-15 штук). При этом каждой гистограмме соответствуют матрицы расположения камеры относительно СК объекта $T_{obj_m, sensor}$ и матрицы расположения СК SGURF относительно СК датчика $T_{sensor, SGURF_m}$.

Затем производится расчет предварительных преобразований от СК датчика к СК объекта для каждого из найденных соседей (M штук):

$$T_{obj, sensor} = T_{obj_m, sensor} * T_{sensor, SGURF_m} * T_{sensor, SGURF_0}^T \quad (1)$$

В целях уточнения преобразований производится выравнивание модели объекта и уточнение координат. Для этого для каждого из предполагаемых вариантов из БД загружается модель объекта, устанавливается в начало СК сенсора, и к ней применяются преобразования вида (1). Далее при помощи итеративного алгоритма ближайших точек [4] производятся попытки совместить геометрическую модель объекта и кластер точек, причем перемещается только загруженная модель объекта, тогда как кластер точек остается неподвижным (рисунок 4). Результатом выравнивания является матрица смещения T_{icp} . Окончательные матрицы расположения объекта будут рассчитываться по формуле:

$$T_{fin} = T_{icp} * T_{obj, sensor} \quad (2)$$

На рисунке 4 изображены гипотезы расположения объекта. Красным цветом показан кластер точек, принадлежащий объекту и сформированный во время выбора ОМ в рабочей зоне. Синим показаны трехмерные модели объекта, к которым применили преобразования вида (1), то есть гипотезы расположения объекта по «грубым прикидкам». Зеленым цветом изображены модели объекта с преобразованиями вида (2), то есть уточненные гипотезы расположения.

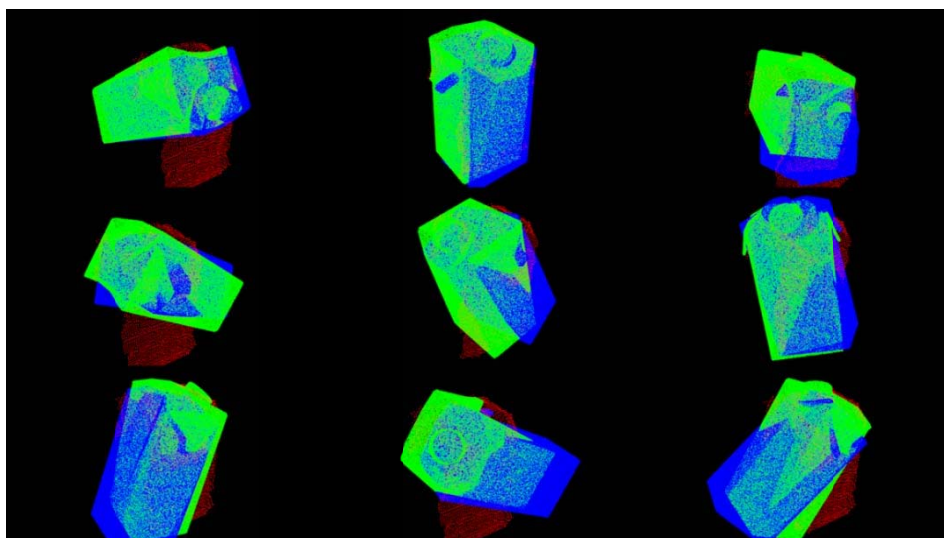


Рисунок 4 – Гипотезы расположения объекта

Чтобы из всех гипотез выбрать наиболее подходящую, для каждой из гипотез производится проверка того, насколько точки кластера совпадают с геометрической моделью объекта. Если точки лежат вблизи поверхности объекта в заданных оператором пределах, то гипотеза принимается, если нет – гипотеза отбрасывается. Если проверку прошли несколько гипотез, то выбирается та, у которой среднее отклонение точек минимально. На рисунке 5 изображена трехмерная модель объекта, расположенная в соответствии с наилучшей гипотезой расположения объекта.

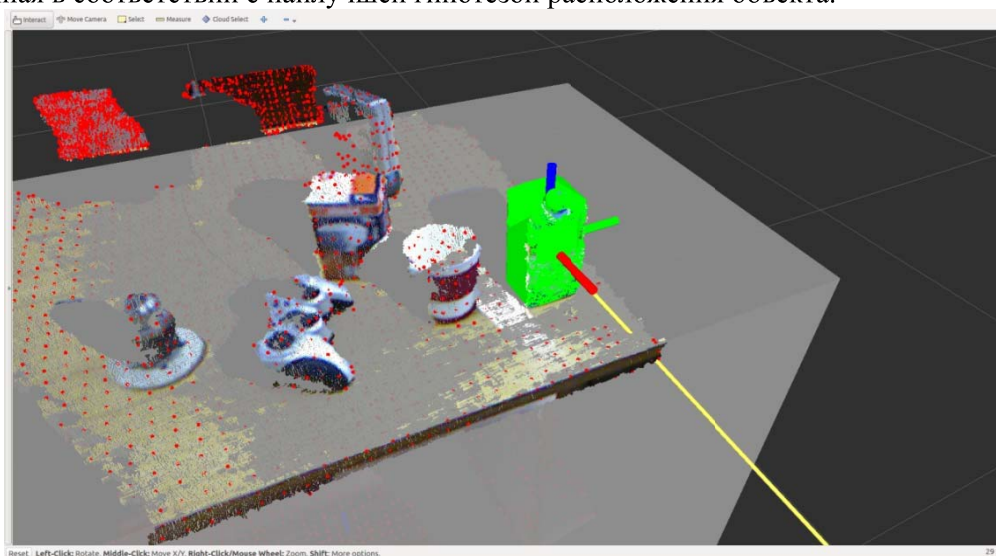


Рисунок 5 – Результат работы алгоритма оценки координат объекта

Представленная система технического зрения была реализована на функционально-моделирующем стенде Дмитровского Филиала МГТУ им. Н.Э. Баумана. Аппаратная часть СТЗ состоит из сенсоров, в качестве которых использовались датчики глубины семейства PrimeSense, такие как Microsoft Xbox Kinect 360 и Asus Xtion Pro Live.

Программная часть реализована с использованием мета-операционной системы ROS[5], набора библиотек для работы с облаками точек Point Cloud Library (PCL) [6] и фреймворка для создания алгоритмов систем технического зрения Object Recognition Kitchen (ORK) [7].

PCL – библиотека алгоритмов для обработки облаков точек и трехмерной геометрии с открытым исходным кодом. PCL содержит алгоритмы фильтрации, сегментации, сглаживания и анализа облаков точек, а также алгоритмы расчета различных дескрипторов.

ORK представляет собой удобный фреймворк для создания алгоритмов работы систем технического зрения. ORK содержит несколько готовых алгоритмов распознавания объектов, а также позволяет разрабатывать и встраивать новые методы и способы распознавания. В ORK реализован ряд необходимых при работе функций, таких как работа с базой данных, работа с сенсорами, интеграция с ROS и т.д. ORK реализован в виде набора программ, которые взаимодействуют как ориентированный ациклический граф.

Разработанные алгоритмы и программное обеспечение позволяют производить оценку расположения и пространственной ориентации объектов манипулирования с точностью порядка 5 мм по линейным координатам и 2 градусов по угловым. В дальнейшем планируется развитие СТЗ в направлении автоматической классификации объектов, а также увеличение точности выбора гипотез путем применения алгоритмов машинного обучения.

1. <http://fms.bmstu.ru/>
2. <https://www.willowgarage.com/pages/pr2/overview>
3. Aldoma A., Tombari F., Rusu R., and Vincze M. OUR-CV FH Oriented, Unique and Repeatable Clustered Viewpoint Feature Histogram for Object Recognition and 6DOF Pose Estimation. //DAGM/OAGM Symposium. Lecture Notes in Computer Science Volume 7476, pp. 113-122. Springer, 2012.
4. http://ru.wikipedia.org/wiki/Итеративный_алгоритм_ближайших_точек
5. <http://wiki.ros.org/>
6. <http://pointclouds.org/>
7. http://wg-perception.github.io/object_recognition_core/