

УДК 62-519, 001.57, 004.942, 004.946

РАСПОЗНАВАНИЕ РОБОТА В 3D ОБЛАКЕ ТОЧЕК ОТ ОЧКОВ СМЕШАННОЙ РЕАЛЬНОСТИ

В.А. Скворцова (*v.skvortsova@innopolis.university*)

М.А. Останин (*m.ostanin@innopolis.ru*)

И.М. Афанасьев (*i.afanasyev@innopolis.ru*)

Университет Иннополис, Иннополис

Аннотация. Разработка систем интерактивного программирования для промышленных роботов на основе смешанной реальности является сложной задачей робототехники, требующей комплексного подхода. Эта система имеет интуитивно понятный интерфейс и работает на основе очков смешанной реальности Microsoft HoloLens. Поскольку корректная инициализация положения робота в среде влияет на правильную работу интерфейса, важным этапом работы является калибровка системы. При запуске интерфейса Microsoft HoloLens собирает информацию об окружающей среде, создает трехмерное облако точек и получает текущую конфигурацию робота. На основе этого набора данных система инициализирует начальные координаты базы манипулятора. В данной статье описан разработанный алгоритм для инициализации пространственных координат робота путем распознавания и локализации манипулятора в трехмерном пространстве.

Ключевые слова: детектирование роботов, трехмерное облако точек, очки смешанной реальности, Microsoft HoloLens

Введение

В настоящее время увеличивается популярность интерфейсов взаимодействия человека с роботом на основе устройств смешанной реальности, использующих планшеты, смартфоны и специализированные очки. Объединение данных промышленных манипуляторов с восприятием и способностями человека-оператора может привести к появлению инновационных индустриальных приложений, посвященных взаимодействию с роботами и визуализацией их состояния, что, по мнению экспертов, способно повысить эффективность производства и даже увеличить число рабочих мест [Guhl et al., 2017]. В связи с этим, актуальным направлением в робототехнике является создание системы

интуитивно понятного и эффективного взаимодействия человека и робота. В настоящее время Университет Иннополис разрабатывает систему интерактивного программирования промышленных роботов на основе смешанной реальности [Ostanin, 2018]. Эта система имеет интуитивно понятный интерфейс и работает на основе очков смешанной реальности Microsoft HoloLens. Чтобы программное обеспечение работало правильно, оно должно корректно инициализировать положение робота в мировой системе координат, для чего требуется калибровка системы. Калибровка программного обеспечения включает локализацию трехмерной модели робота в облаке точек от MS HoloLens. Поэтому, целью данной работы является разработка быстрого алгоритма для определения локализации промышленного манипулятора в трехмерном облаке точек в соответствии с конфигурацией робота. С этой целью мы сравниваем разные методы обнаружения и локализации объектов в трехмерном облаке точек.

Анализ литературы показывает, что проблема локализации облаков точек актуальна в разных сферах человеческой жизни, таких как детектирование объектов в трехмерном облаке точек на городских улицах [Velizhev et al., 2012], биомеханический анализ походки человека [Gabbasov et al., 2015; Danilov et al., 2016], детектирование рельефа в качестве воздушного картографирования на базе трехмерного лидара [Sabirova, et al., 2019], задачи компьютерного зрения [Nath, 2014] и многие другие. Авторы некоторых статей предлагают различные способы описания моделей для их дальнейшей локализации в пространстве. Так, например, в статье [Velizhev et al., 2012] рассмотрены автоматическая локализация и распознавание объектов в трехмерном облаке точек городских улиц на базе модели неявной формы (ISM)¹, которая распознает объект, выбрав его центральное местоположение. В другом исследовании [Sfikas et al., 2011] предложен метод, основанный на сочетании свойств конформной геометрии и топологической информации на графе, выделяющий контрольные точки объекта и отношения между ними. Другие статьи предлагают методы вписывания модели суперквадратических поверхностей в 3D облако точек на основе алгоритма RANSAC² для локализации и оценки позы тела человека [Afanasyev et al., 2012] и распознавания жестов [Afanasyev et al., 2013]. Другие авторы дают свои рекомендации по записи и обработке карт, как это сделано в статье [Rusu et al., 2008], в которой получены трехмерные карты объектов домашней среды (в частности кухни), основанные на PCD³. Метод включает статистический анализ, оценку постоянных характеристик

1 от англ. implicit shape models

2 от англ. random sample consensus

3 от англ. point cloud data

гистограмм, подбор и сегментацию среды. Часто вместо описания модели или карты авторы предлагают методы вписывания одного облака точек в другое. Так авторы [Chetverikov et al., 2005] изучают проблему геометрического выравнивания двух зарегистрированных и частично перекрывающихся, зашумленных трехмерных точечных наборов. Для решения этой задачи они предлагают алгоритм последовательного использования подхода наименьших квадратов на всех этапах операции. Похожее решение было предложено в статье [Chen, 2007], использующей интегрированный дескриптор локальной поверхности для распознавания трехмерных объектов. Этот дескриптор характеризуется своим центроидом, типом локальной поверхности и двумерной гистограммой, показывающей частоту появления значений индекса формы против углов между нормалью контрольной точки функции и ее соседями. Однако эти дескрипторы рассчитываются только для характерных точек областей с большим изменением формы. Чтобы ускорить извлечение дескрипторов для большого набора объектов, локальные участки поверхности моделей индексируются в хеш-таблицу. Учитывая набор тестовых локальных наборов поверхности, голоса подаются за модели, содержащие похожие дескрипторы поверхности. На основании соответствия локальных участков поверхности выдвигаются гипотетические модели и выполняется проверка выравнивания моделей с тестовыми данными для наиболее вероятных моделей, встречающихся в сцене с помощью алгоритма итеративной ближайшей точки (ICP)⁴. Другое интересное решение этой проблемы изложено в статье [Tombari, 2010], где авторы предлагают использовать голосование Хафа для обнаружения фигур свободной формы в трехмерном пространстве со значительной степенью окклюзии и беспорядка. Этот метод сопоставляет трехмерные объекты для накопления доказательств присутствия искомым объектов в трехмерном пространстве Хафа.

1 Анализ алгоритмов и моделей для распознавания роботов в 3D облаке точек

В этой главе мы рассматриваем и сравниваем разработанные алгоритмы и модели для распознавания манипуляторов в 3D облаке точек.

1.1 Оценка пространственного положения манипуляторов на основе алгоритма случайной выборки (RANSAC)

Этап предварительной обработки включает в себя понижающую дискретизацию наборов данных модели и сцены, оценку нормалей для сцены и оценку дескрипторов для каждого облака точек. Описание модели

⁴ от англ. Iterative closest point

основано на примитивах алгоритма согласования случайных выборок (RANSAC), который является итеративным методом оценки параметров математической модели из набора наблюдаемых данных, вычисляющим точки (outliers), не принадлежащие модели. Этот метод считается надежным для установления соответствия между простой трехмерной моделью и облаком, хотя, в нашем случае, нам не удалось получить удачные результаты, в силу сложности модели промышленного манипулятора, который трудно описать простыми моделями геометрических примитивов, такими как цилиндр, куб или шар. Дополнительная проблема заключается в том, что каждая модель в этом случае требует индивидуальной конфигурации. Это требует много ресурсов и не практично с точки зрения реальных решений.

1.2. Локализация роботов на базе алгоритма группирования соответствия (Correspondence Grouping algorithm)

Предварительная обработка данных состоит из следующих этапов: оценка нормалей и описания для каждого облака точек. Обработка представляет собой определение точечных соответствий между моделью и сценой (с помощью k -мерной тройки) и кластеризацию ранее найденных соответствий. Дерево k -d или k -мерное дерево — это структура данных, используемая в информатике для организации некоторого числа точек в пространстве с k измерениями. Это бинарное дерево поиска с другими ограничениями. K -деревья очень полезны для поиска дальности и ближайшего соседа. Для наших целей мы работаем только с облаками точек в трех измерениях, поэтому все наши k -d деревья трехмерные. Каждый уровень дерева k -d разделяет всех дочерних элементов по определенному измерению, используя гиперплоскость, перпендикулярную соответствующей оси. В корне дерева все дочерние элементы разделены на основе первого измерения. Каждый уровень вниз в дереве делится на следующее измерение, возвращаясь к первому измерению, когда все остальные были исчерпаны. Самый эффективный способ построения дерева k -d использует метод разбиения, подобный быстрой сортировке, при котором срединная точка (на полдистанции от начала координат) помещается в корень, а остальные точки — близкие и дальние по дистанции: слева и справа по ветвям соответственно. Затем процедура повторяется для левого и правого поддеревьев, пока последние разделяемые деревья не будут состоять из одного элемента [Yang et al., 2017].

Далее мы используем алгоритм кластеризации на базе преобразования Хафа, выделяющим признаки с поиском объектов в определенном классе фигур с помощью процедуры голосования. Эта процедура выполняется в пространстве параметров (в так называемом аккумуляторе Хафа), в котором локальные максимумы определяют объекты-кандидаты [Tombari,

2010]. В ходе реализации этой методологии выяснилось, что этот алгоритм не может работать с предложенными моделями. Алгоритм мог работать только с данными, полученными очками виртуальной реальности и обработанными вручную. Скорее всего, это происходит по двум причинам: во-первых, очки виртуальной реальности имеют большую ошибку измерения расстояния (наша проверка показала: ошибка измерения расстояния составила 9,72 см), во-вторых, этот алгоритм может работать только с точками, зависящими друг от друга, т.е. если во время выполнения алгоритма не будет достаточного количества зависимых точек, алгоритм предполагает, что в сцене нет модели. Таким образом, эта методология оказалась неработоспособной для локализации роботов.

1.3 Детектирование робота на основе алгоритма итеративной ближайшей точки

В основе следующей методологии лежит алгоритм итеративной ближайшей точки (ICP), этапами которого являются:

1. Связка точек по критерию ближайшего соседа.
2. Оценка параметров преобразования с помощью функции среднеквадратичной стоимости.
3. Преобразования точек с помощью оценочных параметров.
4. Многократные итерации (которые заново связывают точки и т.д.).

В этой работе мы рассмотрим две реализации методологий с использованием алгоритма ICP. Идея первой реализации состоит в том, что мы многократно запускаем алгоритм ICP с различными начальными позициями модели на сцене и выбираем однородное преобразование, в котором достигаем наибольшего согласия между моделью и сценой. При этом выполняется следующая последовательность для каждой итерации:

1. Устанавливаем случайную начальную позицию модели на сцене.
2. Запускаем алгоритм ICP.
3. Рассчитываем точность совпадения точек сцены и модели.
4. Если точность максимальна, то получаем наилучшее однородное преобразование.

Основным преимуществом такой реализации является то, что, многократно повторяя алгоритм при разных начальных условиях, мы уменьшаем вероятность достижения локального минимума. Этот метод оказался очень надежным по сравнению с другими алгоритмами. Из 8 тестов он сработал все 8 раз, имея 100 итераций в каждой попытке. На Рис. 1 показан результат работы этого алгоритма. Основным недостатком алгоритма является скорость его выполнения. В среднем расчеты занимают 53 минуты для каждого теста.

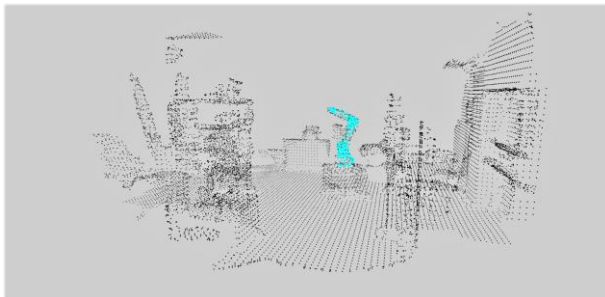


Рис 1. Детектирование модели манипулятора KUKA Agilus на основе алгоритма итеративной ближайшей точки со случайным положением модели

Вторая реализация основана на том, что мы пытаемся определить количество объектов в сцене с помощью кластерного анализа. В этом случае пользователь должен самостоятельно указать количество наблюдаемых в сцене объектов. Тогда мы сравниваем объем кластеризованных объектов с объемом модели и выбираем наиболее подходящее облако. Затем мы запускаем алгоритм ICP, который работает быстрее, поскольку размер выбранного облака будет на порядок меньше, чем размер облака сцены. Однако, эта реализация оказалась менее точной, чем предыдущая. Из 8 тестов метод правильно определил однородную трансформацию только в 3 случаях. Результат удачной работы этого алгоритма показаны на Рис. 2. Это связано с тем, что даже несмотря на уменьшение размера сцены, вероятность попадания в локальный минимум остается высокой. Также была проблема с определением количества объектов в сцене. Однако этот метод работает быстрее, демонстрируя среднее время выполнения теста – 35 секунд.

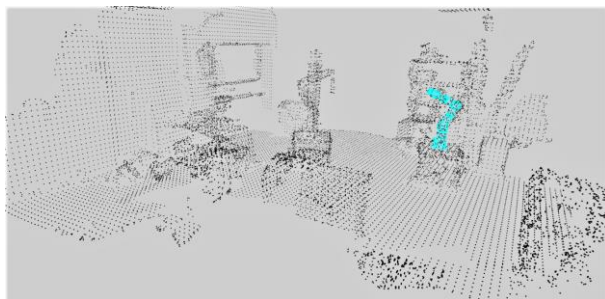


Рис 2. Детектирование модели промышленного манипулятора KUKA PWA на основе алгоритма итеративной ближайшей точки с кластеризацией сцены

Таблица. 1. Сравнение разработанных алгоритмических решений для распознавания модели манипуляторов в 3D облаке точек.

Название алгоритма	Применимость к решению поставленной задачи	Точность	Время обработки
Оценка положения манипуляторов на основе алгоритма случайной выборки	Нет	-	-
Локализация роботов на базе алгоритма соответствующей группировки	Нет	-	-
Детектирование робота на основе алгоритма итеративной ближайшей точки со случайным положением модели	Да	0.03 м	53 минуты
Детектирование робота на основе алгоритма итеративной ближайшей точки с кластеризацией сцены	Да	от 0.03 до 2.5 м	35 секунд
Локализация роботов в 3D облаке точек на базе алгоритма плотности шума (DBSCAN)	Да	от 0.04 м	23 секунды

2 Разработка алгоритма локализации роботов в 3D облаке точек на базе алгоритма плотности шума (DBSCAN)

Таким образом, после имплементации и анализа работы всех методологий (описанных в Разделе 1) мы создали наш собственный алгоритм, в котором учли слабые стороны алгоритмов на базе ICP и использовали их преимущества. Блок-схема нашего алгоритма приведена на Рис. 3. Предварительные обработки облаков точек сцены и модели отличаются. Внутренние точки были обнаружены в большинстве модельных облаков точек. Это вносит большие ошибки в работу любого алгоритма распознавания. Мы разделяем облако точек модели на слои вдоль одной из осей координат и удаляем внутренние точки, используя алгоритм Джарвиса, чтобы построить минимальную выпуклую оболочку

для каждого слоя. Затем, мы удаляем пол в облаке точек сцены, чтобы упростить процедуру кластеризации. Мы используем алгоритм RANSAC для инициализации плоскостей. После этого убираем плоскость с минимальной z-координатой. Следующим этапом предварительной обработки сцены является кластеризация точек сцены в объектах. Для этой цели мы используем пространственную кластеризацию на основе алгоритма плотности шума (DBSCAN) [Schubert et al., 2017], который представляет собой алгоритм кластеризации на основе плотности: при заданном наборе точек в некотором пространстве он группирует точки, которые тесно упакованы вместе (точки с множеством соседей), помечая выпадающие точки, которые лежат одни в областях с низкой плотностью.



Рис. 3 Блок-схема предложенного алгоритма.

После этапа предварительной обработки мы измеряем объем модели и каждого кластерного объекта. Для этого мы подбираем наиболее похожие объекты по объему. Установив начальное положение модели для объекта (связывая средний вектор каждого из облаков точек) и запустив ICP-алгоритм, мы повторяем этот шаг для каждого подходящего объекта с точки зрения объема и записываем значение ошибки для каждого случая. После чего выбираем преобразование, при котором ошибка была наименьшей. Результат работы показал, что алгоритм работает правильно как для разреженных, так и для плотных облаков точек (Рис. 4). Среднее

время работы алгоритма составляет 23 секунды, что быстрее, чем работа других методологий. Из 8 тестов он правильно определил трансформацию в 7 случаях с ошибкой в 0.04 м. Во время работы алгоритм может сообщать об ошибке, если в сцене присутствуют два робота одинакового размера в одинаковой конфигурации.

Результаты работы разных алгоритмов для распознавания модели манипуляторов в 3D облаке точек, включая предложенный нами метод на базе алгоритма плотности шума (DBSCAN), представлены в Таблице 1.

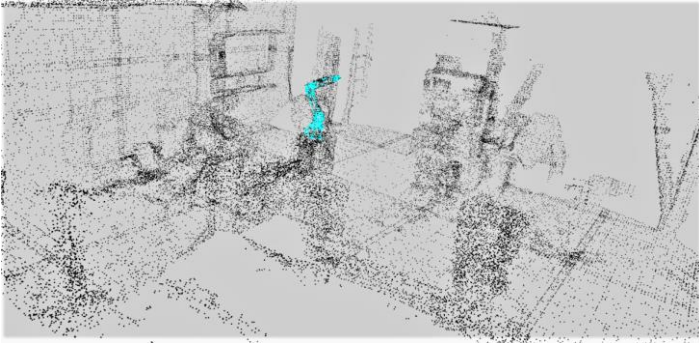


Рис. 4 Результат работы алгоритма на плотном облаке точек.

Заключение

В статье был предложен разработанный быстрый и точный алгоритм локализации 3D модели промышленного манипулятора в сцене, которая была получена с помощью очков смешанной реальности Microsoft HoloLens. На основе 3D облаков точек внешней среды и модели алгоритм инициализировал координаты базы манипулятора. Этот проект решает проблему с калибровкой системы интерактивного программирования промышленных манипуляторов, используя очки смешанной реальности. Мы достигли быстрой работы алгоритма (порядка 20 секунд) и точности в 0.04 м. Алгоритм надежно работает в ситуациях, когда в сцене присутствует один детектируемый робот необходимой конфигурации, в противном случае точность алгоритма не гарантируется.

Список литературы

- [Afanasyev et al., 2012] I. Afanasyev et al. 3D Human Body Pose Estimation by Superquadrics. VISAPP (2): 294–302, 2012.
- [Afanasyev et al., 2013] I. Afanasyev and M. De Cecco. 3D Gesture Recognition by Superquadrics. VISAPP (2): 429–433, 2013.

- [**Chen, 2007**] H. Chen and B. Bhanu. 3D free-form object recognition in range images using local surface patches. *Pattern Recognit. Lett.*, 28(10), 2007.
- [**Chetverikov et al., 2005**] D. Chetverikov, D. Stepanov, and P. Krsek. Robust Euclidean alignment of 3D point sets: The trimmed iterative closest point algorithm. *Image and Vision Computing*, 23(3): 299–309, 2005.
- [**Danilov et al., 2016**] I. Danilov, B. Gabbasov, I. Afanasyev, and E. Magid, ZMP Trajectory from Human Body Locomotion Dynamics Evaluated by Kinect-based Motion Capture System. In *VISAPP*, pp. 162-168, 2016.
- [**Gabbasov et al., 2015**] B. Gabbasov, I. Danilov, I. Afanasyev, and E. Magid Toward a human-like biped robot gait: Biomechanical analysis of human locomotion recorded by Kinect-based Motion Capture system, *ISMA*, 2015.
- [**Guhl et al., 2017**] J. Guhl, S. Tung, J. Kruger, Concept and architecture for programming industrial robots using augmented reality with mobile devices like Microsoft HoloLens. In *IEEE ETFA*, pp. 1-4, 2017.
- [**Nath, 2014**] V. Nath and S. E. Levinson. *Computer vision*. Springer Briefs in Computer Science, (9783319056050):33–38, 2014.
- [**Ostanin, 2018**] M. Ostanin and A. Klimchik. Interactive Robot Programing Using Mixed Reality. *IFAC Symposium on Robot Control*, 2018.
- [**Pomelau et al., 2015**] F. Pomerleau, F. Colas, and R. Siegwart. A Review of Point Cloud Registration Algorithms for Mobile Robotics. *Foundations and Trends in Robotics*, 4(1):1–104, 2015.
- [**Rusu et al., 2008**] R. B. Rusu et al Towards 3D Point cloud based object maps for household environments. *Robot Auton Syst*, 56(11):927–941, 2008.
- [**Sabirova et al., 2019**] A. Sabirova, et al., Ground Profile Recovery from Aerial 3D LiDAR-based Maps. In *FRUCT*, arXiv:1903.11097, 2019.
- [**Schubert et al., 2017**] Schubert Erich et al. DBSCAN Revisited, Revisited: Why and How You Should (Still) Use DBSCAN. *ACM TODS*, 10197, 2017.
- [**Sfikas, 2011**] K. Sfikas, et al. ConTopo: Non-rigid 3D object retrieval using topological information guided by conformal factors. *EG 3DOR*, 2011.
- [**Skibbe, 2012**] H. Skibbe, et al. Fast rotation invariant 3D feature computation utilizing efficient local neighborhood operators. *IEEE TPAMI*, 34(8), 2012.
- [**Tombari, 2010**] F. Tombari and L. Di Stefano. Object recognition in 3D scenes with occlusions and clutter by Hough voting. *PSIVT*, 2010.
- [**Velizhev et al., 2012**] A. Velizhev, et al. Implicit shape models for object detection in 3D point cloud. *ISPRS Annals*:179–184, 2012.
- [**Wohllhart, 2015**] P. Wohllhart and V. Lepetit. Learning descriptors for object recognition and 3D pose estimation. *IEEE CVPR*, 3109–3118, 2015.
- [**Yang et al., 2017**] J. Yang, K. Xian, Y. Xiao, Z. Cao. Performance evaluation of 3D correspondence grouping algorithms. In *IEEE 3DV*, 467-476, 2017.