

Санкт-Петербургский государственный университет

Кафедра системного программирования

Ярош Дмитрий Сергеевич

Выделение и ассоциирование плоскостей из облаков точек

Отчет по учебной практике (научно-исследовательской работе)

Научный руководитель:
доцент кафедры СП, к.т.н., Ю. В. Литвинов

Консультант:
Инженер-исследователь, Сколтех, А. В. Корнилова

Санкт-Петербург
2021

Оглавление

| | |
|--|-----------|
| Введение | 3 |
| 1. Постановка задачи | 5 |
| 1.1. Общая задача | 5 |
| 1.2. Задача на семестровую практику | 6 |
| 2. Обзор | 7 |
| 2.1. Датчики | 7 |
| 2.2. Датасеты | 9 |
| 2.3. Метрики | 11 |
| 2.4. Распознавание плоскостей | 12 |
| 2.4.1. Основанные на RANSAC методы | 13 |
| 2.4.2. Основанные на преобразовании Хафа методы | 14 |
| 2.4.3. Методы, использующие методику растущих областей | 15 |
| 3. Стенд для запуска и оценки производительности алгоритмов | 16 |
| 3.1. Подготовка данных | 16 |
| 3.2. Фреймворк для оценки точности и производительности алгоритмов | 17 |
| Текущие результаты и дальнейшие планы | 19 |
| Список литературы | 20 |

Введение

Технологии трехмерной съемки, такие как лидары, 3д-сканеры, системы Kinect и камеры глубины, активно используются в современных автономных системах, так как трехмерное представление мира, получаемое с их помощью, повышает качество работы алгоритмов для многих подзадач восприятия окружающего мира автономной системой. Ключевыми среди этих подзадач являются локализация и построение карты местности, которые обычно решаются алгоритмами *simultaneous localization and mapping* (SLAM). Согласно [50] в данных алгоритмах можно выделить две глобальные подзадачи: **frontend**, часть, которая отвечает за обнаружение ориентиров — заметных объектов в окружающей среде и их сопоставление на последовательности кадров (ассоциирование), и **backend**, алгоритм высокого уровня, который анализирует относительное положение ориентиров и на его основании вычисляет положение системы в пространстве, а также обновляет карту местности.

Внедрение технологий трехмерной съемки в алгоритмы автономных систем осложняется недостатком хорошо различимых объектов-ориентиров в облаках точек. Базовые универсальные ориентиры, которые можно использовать в таких алгоритмах, — плоскости. Их можно найти почти во всех местах возможной работы автономных систем: как внутри помещений, так и снаружи. В то время как алгоритмы backend части SLAM фреймворков, использующие плоскости в качестве ориентиров [41] [53] [31] [34], активно развиваются в последние годы, разработка frontend части для данных решений, несмотря на существование различных алгоритмов для **распознавания плоскостей**, таких как [24], [33], [15] и [62], и **ассоциирования плоскостей**, например [64], [18], [13] и [35], столкнулась с проблемой разнородных датчиков, требующих настройки множества параметров для каждого алгоритма с целью предотвращения потери точности.

Эта неоднородность в измерениях датчиков вызвана различным аппаратным оснащением и технологиями, используемыми для измерения расстояния до объекта (например, лидар использует лазерные датчи-

ки, а некоторые камеры глубины вычисляют расстояние с помощью алгоритмов триангуляции). Такое различие в конструкции датчиков и как следствие разнородность данных является причиной отсутствия универсального решения для распознавания и ассоциирования плоскостей. Более того, автономные системы также могут быть оснащены различным оборудованием, которое требует специальной оптимизации алгоритмов (например, оптимизации для контроллеров на базе ARM или на базе графических процессоров). Хотя реализация такого алгоритма зависит от платформы и является низкоуровневой, что требует разработки на таких языках, как C, C++ или Rust, данный алгоритм должен поддерживать возможность использования из языка программирования высокого уровня, например Python, поскольку он обычно используется для прототипирования вне автономных систем.

В рамках данной работы будут описаны существующие решения в области распознавания и ассоциирования плоскостей в разнородных облаках точек, предложен универсальный frontend для SLAM алгоритмов, использующий плоскости в качестве ориентиров и подготовленный как для работы на автономных системах, так и для прототипирования в лаборатории.

1. Постановка задачи

В данном разделе будет поставлена цель и описаны решаемые в работе задачи как на протяжении всей магистерской диссертации, так и в рамках семестровой практики.

1.1. Общая задача

Целью данной работы является разработка универсального frontend для SLAM фреймворка, способного распознавать и ассоциировать плоскости в разнородных облаках точек, и реализация данного решения в виде библиотеки на C++, оптимизированной для GPU и ARM, с наличием интерфейса для языка Python. Для достижения данной цели были сформулированы следующие задачи.

1. Провести обзор существующих алгоритмов **распознавания плоскостей** в облаках точек.
2. Адаптировать выбранный алгоритм для всех широкоиспользуемых датчиков либо разработать новый алгоритм для решения данной задачи.
3. Замерить производительность предложенного алгоритма распознавания плоскостей и существующих алгоритмов на стандартных тестах, использующих облака точек (например, датасеты KITTI [56], TUM [7] и т. д.).
4. Провести обзор существующих алгоритмов **ассоциирования плоскостей** в облаках точек.
5. Адаптировать выбранный алгоритм ассоциирования для всех широкоиспользуемых датчиков либо разработать новый алгоритм для решения данной задачи.
6. Замерить производительность предложенного алгоритма ассоциирования плоскостей и существующих алгоритмов на стандартных тестах, использующих облака точек.

7. Реализовать предложенные алгоритмы в виде библиотеки на C++ с интерфейсом на языке Python.
8. Исследовать и реализовать возможные оптимизации созданного фреймворка для графического процессора.
9. Исследовать и реализовать возможные оптимизации созданного фреймворка для процессора на базе ARM архитектуры.

1.2. Задача на семестровую практику

В рамках описанной цели и задач для данной работы было решено сформулировать следующую цель для семестровой практики — провести обзор существующих алгоритмов распознавания плоскостей в облаках точек, а также подготовить данные и фреймворк для оценки данных алгоритмов. Для достижения данной цели были сформулированы следующие задачи.

1. Провести обзор существующих алгоритмов **распознавания плоскостей** в облаках точек.
2. Провести обзор существующих датасетов для тестирования алгоритмов распознавания и ассоциирования плоскостей в облаках точек и наладить процесс подготовки датасета в случае отсутствия подходящих существующих решений.
3. Разработать фреймворк для тестирования и оценки производительности существующих решений в области распознавания и ассоциирования плоскостей в облаках точек.

2. Обзор

В данном разделе будут рассмотрены основные виды датчиков для трехмерной съемки, существующие алгоритмы распознавания и ассоциирования плоскостей, а также датасеты и метрики, используемые для тестирования разработанных решений в данной области.

2.1. Датчики

Существующие датчики для трехмерной съемки представлены двумя типами устройств: камерами глубины и лидарами.

Камеры глубины за исключением стереокамер используются в основном внутри помещений, так как обладают небольшой максимальной дистанцией съемки (не более 6 метров), но достаточной точностью для работы с близко расположенными друг к другу объектами в зоне их покрытия. Развитие камер глубины началось с использования простых стереокамер, например Intel D455 [29], Stereolabs ZEN 2 [52], вычисляющих дистанцию до объекта за счет совмещения изображений с обоих объективов и триангуляции. К сожалению, данные устройства не способны работать в условиях плохой освещенности из-за отсутствия источника излучения, а потому не могут быть применены в автономных системах в качестве основного датчика для построения облака точек окружения. В связи с этим в данной работе такие камеры рассматриваться не будут, а все дальнейшие упоминания камер глубины подразумевают наличие у них источника излучения для работы в условиях низкой освещенности. Более современные устройства используют дополнительный источник излучения — специальный проектор известного шаблона (structured light), позволяющий камере работать даже в условиях низкой освещенности и однородной текстуры окружающих объектов, а также увеличивающий по сравнению с обычной стереокамерой точность. Примерами таких устройств являются Xbox Kinect первого поколения [38], Zivid One+ Small [63]. Самыми современными на данный момент решениями в данной области являются камеры, использующие принцип замера времени отражения лазерного луча от поверх-

ности (TOF-камеры), например Asus Xtion 2 [4], Azure Kinect [37]. Данные приборы позволяют добиться в несколько раз большей максимальной дальности распознавания объектов и не подвержены губительному влиянию солнечного света, который способен серьезно уменьшить точность работы камер, основанных на других технологиях. Это позволяет использовать TOF-камеры даже вне помещений.

Однако основным средством для получения облаков точек вне помещений являются лидары — устройства, использующие лазерные лучи для определения дальности до окружающих объектов. Максимальная дальность работы лидаров может превышать 1 километр. Существующие лидары можно разделить на следующие категории по технологии их работы.

- Флэш-лидары — наследники TOF-камер, обладающие большей максимальной дальностью обнаружения объектов. Примером такого устройства является Continental HFL [3].
- Сканирующие механические лидары — устройства, обладающие несколькими лазерными головками, закрепленными на подвижной, быстро вращающейся платформе. Данная конструкция позволяет уменьшить число используемых лазерных датчиков и расширить угол обзора до 360 градусов. Примером такого устройства является Velodyne Puck [55].
- Сканирующие оптические лидары — аналоги механических лидаров, в которых сложная подвижная система заменена набором оптических элементов. Примерами таких устройств являются Quaenergy S3 [45], Intel RealSense L515 [30].

Таким образом, существует достаточно много различных видов приборов, позволяющих строить облака точек окружения. Несмотря на наличие небольших отличий в качестве получаемых облаков между различными видами приборов одного типа, наибольшую разность демонстрируют при сравнении облака точек, полученные с помощью камер

глубины и лидаров. Облака точек, построенные с помощью камер глубины, имеют высокую плотность и возможность прямого сопоставления с обычной rgb фотографией. В то же время облака точек, полученные с помощью лидаров, характеризуются невысокой плотностью, уменьшающейся по мере удаления объекта от точки сканирования, а также отсутствием прямого сопоставления с некоторым привычным человеку фотоснимком, что может доставлять дополнительные трудности при разметке данных.

2.2. Датасеты

Активное развитие области SLAM алгоритмов в последние годы привело к созданию большого числа датасетов и фреймворков для тестирования и замера производительности таких алгоритмов. Среди данных решений присутствуют следующие наборы облаков точек, собранные с использованием камер глубины или лидаров различных моделей.

- Датасет HILTI SLAM [22] — один из самых современных наборов данных, записанный с помощью двух лидаров специально для тестирования алгоритмов SLAM на небольших роботах.
- Датасет KITTI [20] — один из наиболее популярных наборов данных в области SLAM вне помещений, записанный с помощью лидарного комплекса, закрепленного на автомобиле.
- Датасет Waymo [48] — альтернативный набор данных для тестирования алгоритмов SLAM вне помещений от американской компании Waymo [60]. В отличие от KITTI, в данном решении используются более современные лидары, а также упрощена выгрузка данных из облачного хранилища.
- Датасет Oxford Radar RobotCar [40] — набор данных для тестирования алгоритмов SLAM вне помещений, в котором в отличие от аналогов помимо данных лидаров также присутствуют радарные данные.

- Датасет TUM RGB-D SLAM [7] — один из наиболее популярных наборов данных в области SLAM внутри помещений, записанный с использованием Xbox Kinect.
- Датасет Mobile RGBD [54] — аналог предыдущего решения, также использующий Xbox Kinect для записи внутри помещения.
- Датасет CORBS [59] — набор данных для тестирования алгоритмов SLAM внутри помещений, в котором в отличие от аналогов использована более современная версия Xbox Kinect для записи данных.
- Датасет NYUv2 [39] — аналог TUM RGB-D SLAM датасета, также записанный с использованием Xbox Kinect
- Датасет ICL-NUIM [6] — синтетический набор данных, построенный на базе заранее созданной трехмерной модели помещения, что позволяет получать высококачественные данные об удалении объектов от датчика, а также предоставляет доступ к исходной геометрии помещения.
- Датасет 3DCSR [2] — набор данных, специально подготовленный для решения задач обработки облаков точек, полученных различными датчиками. В данном датасете одни и те же сцены одновременно записаны с помощью камеры глубины и лидара.

К сожалению, общая задача SLAM алгоритмов не фиксирует ориентиры, которые необходимо использовать при построении карты окружающей местности и траектории движения автономной системы на ней. В связи с этим не существует широко используемых и общепринятых датасетов облаков точек с вручную размеченными на них плоскостями. Авторы алгоритмов вынуждены вручную размечать небольшие фрагменты существующих датасетов для замера производительности своих решений. Несмотря на наличие большого числа опубликованных наборов данных, в большинстве работ [43] [42] [17] [9] [21] в области SLAM алгоритмов используются датасеты TUM, ICL NUIM и KITTI. В связи

с этим было принято решение выбрать указанные датасеты в качестве основы для описываемого в данной работе датасета размеченных плоскостей в облаках точек.

2.3. Метрики

Для оценки качества работы алгоритмов распознавания и ассоциирования произвольных объектов, в том числе и плоскостей, используют метрики, разработанные для оценки качества решения задачи instance segmentation — распознавания различных объектов одного вида. Классическими метриками в данной области является набор из шести метрик: plane precision, plane recall, USR (under-segmentation rate), OSR (over-segmentation rate), missed и noise. Впервые данный набор метрик был представлен в работе [65]. Ключевым недостатком данных метрик является сложность определения верно сегментированной плоскости — в своей работе Ж.-Е. Deschaud и др [10] принимают плоскость как верно найденную при пересечении не менее 80 процентов точек реальной плоскости с ее позицией, определенной алгоритмом. Кроме того, для сравнения различных алгоритмов с использованием описанной метрики необходимо анализировать 6 параметров одновременно, что может помешать выявить более точный алгоритм. Для решения данной проблемы были разработаны более современные метрики, представляющие оценку алгоритма в виде одного числового параметра. Например, М. Bennewitz и др. [47] в своей работе демонстрируют результат разработанного алгоритма распознавания плоскостей на снимках, полученных с помощью камеры глубины, с использованием метрики IoU (Intersection-Over-Union)¹. Еще одной активно используемой в современных работах метрикой в данной области является коэффициент Ли Дайса (DICE)². В отличие от IoU, он менее чувствителен к ошибкам и дифференцируем.

¹https://en.wikipedia.org/wiki/Jaccard_index

²https://en.wikipedia.org/wiki/S%C3%B8rensen%E2%80%93Dice_coefficient

2.4. Распознавание плоскостей

Алгоритмы распознавания плоскостей в облаках точек могут применяться в различных сферах использования датчиков трехмерной съемки, таких как аэросъемка [61] [5], лазерное сканирование зданий и сооружений [25] [62], непосредственно SLAM алгоритмы [28]. К сожалению, разработанные для работы в других сферах алгоритмы распознавания плоскостей не всегда можно переиспользовать в задаче построения frontend для SLAM алгоритмов из-за существенных отличий в типах используемых датчиков. Например, при сканировании зданий используют стационарные лазерные системы, обеспечивающие крайне высокую плотность облака точек, недостижимую для автономной системы в движении. Однако, несмотря на существенные различия, основные идеи таких алгоритмов могут быть перенесены и в область SLAM фреймворков при должной адаптации. На данный момент существует немало алгоритмов распознавания плоскостей, которые согласно [61] и [32] можно разделить на несколько групп по ключевой идее их работы:

- стохастические подходы, такие как RANSAC и локальные статистики;
- подходы с использованием параметрических пространств, например преобразование Хафа, кластеризация в параметрическом пространстве;
- кластеризационные подходы, такие как кластеризация на основе октодеревя, кластеризация по K-средним, нечеткая кластеризация и кластеризация по сдвигу среднего значения;
- подходы с использованием методики растущих областей;
- оптимизационные подходы, использующие воксели или суперпиксели и интерпретирующие задачу распознавания плоскостей как задачу на графе.

Несмотря на наличие большого числа возможных подходов к решению задачи распознавания плоскостей в облаке точек, наибольшее развитие в последние годы согласно [46] получили три направления исследований:

- методы, основанные на алгоритме RANSAC, использующие знания о модели плоскости для ее выделения;
- методы, основанные на преобразовании Хафа, использующие параметрическое пространство;
- методы, основанные на подходе растущих областей для использования информации о соседях точек с целью более точного определения их принадлежности той или иной плоскости.

Выбор данных направлений объясняется тем, что алгоритмы, использующие названные подходы, позволяют добиться наилучшего качества распознавания плоскостей среди всех существующих методов. В продолжении данного раздела будут описаны основные представители алгоритмов распознавания плоскостей, разработанных в последние годы и показавших превосходящую аналоги точность работы.

2.4.1. Основанные на RANSAC методы

Одним из первых решений задачи распознавания плоскостей являются методы, основанные на алгоритме RANSAC [16]. RANSAC или RANdom SAmple Consensus — метод оценки параметров некоторой математической модели на основе случайной выборки. Данный алгоритм устойчив к шумам и выбросам, а также способен напрямую использовать модель плоскости, что позволяет минимизировать необходимое для его работы количество памяти. Одну из первых реализаций метода для распознавания плоскостей в облаке точек на базе алгоритма RANSAC в своей работе предложил R. Schnabel [49]. К сожалению, главным недостатком использования алгоритма RANSAC для распознавания плоскостей является его низкая производительность, причина которой — перебор большого числа случайно выбранных для проверки

гипотез. Также данный алгоритм не позволяет разделять плоскости, имеющие одинаковые параметры математической модели, однако находящиеся далеко друг от друга в облаке точек. Существует немало решений, улучшающих данный алгоритм, например [19] и [44] в своих работах решают проблему обобщения сегментации — случая, когда несколько маленьких различных плоскостей находятся близко друг к другу и воспринимаются алгоритмом как единая плоскость с усредненными параметрами. F. Yang [27] использует ячейки с нормальным распределением для предварительной кластеризации облака точек, что позволяет уменьшить число ложно обнаруживаемых плоскостей. В [25] авторы представляют адаптацию алгоритма RANSAC для работы на данных, полученных с помощью стационарного лазерного сканера. Существуют также успешные решения, объединяющие использование алгоритма RANSAC и других подходов к распознаванию плоскостей, таких как метод роя частиц [57], метод заливки [10], принцип минимальной длины описания [62] и преобразование Хафа [12].

2.4.2. Основанные на преобразовании Хафа методы

Преобразование Хафа [26] является одним из стандартных решений для распознавания геометрических элементов, таких как прямые, круги, эллипсы на изображениях с использованием параметрического пространства. Данный метод был успешно перенесен в трехмерное пространство, и подходит для распознавания плоскостей в облаках точек. D. Borrmann и др. [1] описывают и адаптируют для данной задачи различные вариации преобразования Хафа. Дополнительные улучшения алгоритма, основанного на преобразовании Хафа, представлены в [8]. Авторы решают проблемы обработки зашумленных данных, вычислительной сложности алгоритма, а также оптимизируют количество необходимой для представления параметрического пространства памяти. F. A. Limberger и др. [36] демонстрируют адаптацию алгоритма, основанного на преобразовании Хафа, позволяющую проводить распознавание плоскостей в реальном времени. В [11] авторы представляют алгоритм, основанный на преобразовании Хафа и специализированный

для работы в реальном времени на данных, полученных с помощью датчика Kinect.

2.4.3. Методы, использующие методику растущих областей

Метод растущих областей — основанный на областях подход к сегментации изображений, в котором выбираются начальные точки — центры областей, а затем итеративно анализируются их соседи для определения необходимости их включения в данную область. Таким образом, решения, основанные на данном методе, способны использовать информацию о взаимном расположении точек в пространстве. J. Poppinga и др. [14] демонстрируют применение данного метода к задаче распознавания плоскостей, однако авторы отмечают возможные проблемы с производительностью из-за вычисления собственных векторов в ходе работы алгоритма. В [23] данная проблема решается за счет предподсчета нормалей для всех точек облака и последующей работы с центроидами растущих областей. J.-E. Deschaud и др [10] увеличивают точность работы данного алгоритма за счет подсчета уровня планарности каждой точки и более аккуратного выбора стартовых позиций для роста областей. В [43] описывается возможность предварительного разделения облака точек на ячейки, планарность которых определяется с помощью метода главных компонент, позволяющая существенно увеличить скорость работы алгоритма за счет потери точности. Альтернативный подход к применению данного метода приводится в [58]. Авторы используют графовое представление облака точек для объединения соседних регионов плоскости. Feng и др. [15] развивают графовый подход к решению данной задачи и используют агломеративную иерархическую кластеризацию для формирования плоскости в графовом представлении облака точек. Одной из последних работ в данной области является [47], авторы которой разработали алгоритм распознавания плоскостей, адаптированный под камеры глубины и использующий метод заливки для уменьшения количества вычислений нормалей, что приводит к существенному росту производительности и возможности данного решения работать в режиме реального времени.

3. Стенд для запуска и оценки производительности алгоритмов

В данном разделе будет описан реализованный фреймворк для замера производительности и точности алгоритмов распознавания и ассоциирования плоскостей в облаках точек, а также организация процесса создания датасета для тестирования рассматриваемых алгоритмов.

3.1. Подготовка данных

В связи с отсутствием существующих общеиспользуемых датасетов для проверки качества работы алгоритмов распознавания плоскостей в облаках точек было решено подготовить набор данных, который позволит протестировать существующие решения, а также оценить эффективность предложенного в данной работе алгоритма. Для создания датасета было необходимо подобрать наборы облаков точек, записанных с помощью различных камер глубины и лидаров, а затем разметить существующие плоскости в исходных облаках точек. Было принято решение взять датасеты KITTY, HILTI SLAM в качестве источников данных для лидарных облаков точек и датасеты TUM, ICL NUIM в качестве исходных данных облаков точек камер глубины. Данный выбор обусловлен тем, что отобранные наборы данных являются общеиспользуемыми в сфере тестирования SLAM алгоритмов и содержат достаточное для решаемой задачи количество облаков точек, записанных различными датчиками. Для разметки данных лабораторией робототехники Skoltech [51] были предоставлены ресурсы в виде оплачиваемого специалиста-разметчика. С целью ускорения и повышения качества разметки было решено использовать полуавтоматические инструменты для помощи человеку в процессе разметки плоскостей. Создание данных инструментов было проведено при участии студента 2го курса СПбГУ математико-механического факультета по направлению «Технологии программирования» Павла Мокеева. В ходе работы были подобраны подходящие для решаемой задачи средства размет-

ки облаков точек для лидаров и камер глубины, а также предложены полуавтоматические решения для ускорения разметки облаков точек лидаров за счет использования автоматической генерации плоскости по частичному набору ее точек.

3.2. Фреймворк для оценки точности и производительности алгоритмов

Для оценки качества работы и производительности существующих алгоритмов распознавания и ассоциирования плоскостей в облаках точек был разработан фреймворк, позволяющий проводить тестирование различных алгоритмов в данной области на подготовленных наборах данных с использованием существующих метрик. В ходе разработки использовался язык программирования Python, так как он позволяет быстро прототипировать решения, а также удобен для быстрого запуска решения сторонними пользователями. Для работы с облаками точек и изображениями в фреймворке было решено использовать библиотеки Open3D и OpenCV соответственно, так как они имеют интерфейс для языка Python, являются современными поддерживаемыми решениями, а также широко используются для аналогичных задач в данной области. Разработанный фреймворк имеет конвеерную архитектуру, схема работы данного фреймворка представлена на рис. 1.

Модуль чтения данных отвечает за взаимодействие с датасетами, данные которых представлены в различных форматах. Разработанное решение позволяет легко расширять список поддерживаемых форматов хранения исходных облаков точек, а также добавлять возможность работы с новыми форматами файлов, содержащих информацию о разметке плоскостей в облаке точек. На данный момент в фреймворке поддерживаются следующие форматы датасетов: TUM, ICL NUIM. В связи с тем, что разметка облаков точек на плоскости при работе над созданием датасета производилась в разных инструментах для облаков, полученных с помощью лидара и камеры глубины, в фреймворке поддерживаются форматы хранения разметки, используемые в данных инструментах.



Рис. 1: Схема работы фреймворка для оценки точности и производительности алгоритмов

Модуль оценки алгоритмов позволяет сравнить результат работы исследуемого решения и предварительно верно размеченные данные с использованием различных метрик. В рамках данного модуля реализованы такие метрики как IoU, DICE, а также набор из 6 метрик широко используемый в существующих работах в данной области и представленный в [65]. Дополнительно модуль поддерживает простое добавление новых метрик при необходимости.

Текущие результаты и дальнейшие планы

В ходе данного семестра были выполнены следующие задачи.

- Проведен обзор существующих алгоритмов распознавания плоскостей из облаков точек.
- Проведена подготовка датасета для тестирования алгоритмов распознавания и ассоциирования плоскостей в облаках точек с привлечением специалиста-разметчика и студента 2 курса Павла Мокеева.
- Разработан расширяемый фреймворк для тестирования и оценки производительности существующих решений в области распознавания и ассоциирования плоскостей в облаках точек на языке Python.

В следующем семестре планируется опубликовать результаты работы по созданию фреймворка для оценки рассматриваемых алгоритмов на конференции IROS'22³, а также продолжить работу по тестированию существующих алгоритмов для распознавания плоскостей и адаптацию выбранного решения для всех активно используемых в современных автономных системах датчиков трехмерной съемки.

³<https://www.iros2021.org/iros-2022>

Список литературы

- [1] Borrmann Dorit, Elseberg Jan, Lingemann Kai, and Nuchter Andreas. The 3D Hough Transform for plane detection in point clouds: A review and a new accumulator design // 3D Research. — 2011. — 06. — Vol. 0202.
- [2] Huang Xiaoshui, Mei Guofeng, Zhang Jian, and Abbas Rana. A comprehensive survey on point cloud registration. — 2021.
- [3] AG Continental. Continental's High-Resolution 3D Flash LIDAR // официальный сайт. — Access mode: <https://www.continental-automotive.com/en-gl/Passenger-Cars/Autonomous-Mobility/Enablers/Lidars/3D-Flash-Lidar> (online; accessed: 16.12.2021).
- [4] ASUSTeK Computer Inc. Asus Xtion 2 // официальный сайт. — Access mode: <https://www.asus.com/ch-en/Networking-IoT-Servers/Smart-Home/Security-Camera/Xtion-2/?ref=hackernoon.com> (online; accessed: 16.12.2021).
- [5] Awrangjeb Mohammad and Fraser Clive S. Automatic Segmentation of Raw LIDAR Data for Extraction of Building Roofs // Remote Sensing. — 2014. — Vol. 6, no. 5. — P. 3716–3751. — Access mode: <https://www.mdpi.com/2072-4292/6/5/3716>.
- [6] Handa A., Whelan T., McDonald J.B., and Davison A.J. A Benchmark for RGB-D Visual Odometry, 3D Reconstruction and SLAM // IEEE Intl. Conf. on Robotics and Automation, ICRA. — Hong Kong, China. — 2014. — May.
- [7] Sturm J., Engelhard N., Endres F., Burgard W., and Cremers D. A Benchmark for the Evaluation of RGB-D SLAM Systems // Proc. of the International Conference on Intelligent Robot Systems (IROS). — 2012. — ДД°ÑÑ.

- [8] Hulík Rostislav, Španěl Michal, Šmrz Pavel, and Materna Zdeněk. Continuous plane detection in point-cloud data based on 3D Hough Transform // Journal of Visual Communication and Image Representation. — 2014. — Vol. 25, no. 1. — P. 86–97. — Visual Understanding and Applications with RGB-D Cameras. Access mode: <https://www.sciencedirect.com/science/article/pii/S104732031300062X>.
- [9] Cvišić Igor, Marković Ivan, and Petrović Ivan. Recalibrating the KITTI Dataset Camera Setup for Improved Odometry Accuracy // European Conference on Mobile Robots (ECMR). — 2021.
- [10] Deschaud Jean-Emmanuel and Goulette Francois. A Fast and Accurate Plane Detection Algorithm for Large Noisy Point Clouds Using Filtered Normals and Voxel Growing. — 2010. — May.
- [11] Dubé Daniel and Zell Andreas. Real-time plane extraction from depth images with the Randomized Hough Transform. — 2011. — 11. — P. 1084–1091.
- [12] Gaspers Bastian, Stückler Jörg, Welle Jochen, Schulz Dirk, and Behnke Sven. Efficient Multi-resolution Plane Segmentation of 3D Point Clouds. — 2011. — 12. — P. 145–156.
- [13] Wang Yunbiao, Xiao Jun, Liu Lupeng, and Wang Ying. Efficient Rock Mass Point Cloud Registration Based on Local Invariants // Remote Sensing. — 2021. — 04. — Vol. 13. — P. 1540.
- [14] Poppinga Jann, Vaskevicius Narunas, Birk Andreas, and Pathak Kaustubh. Fast plane detection and polygonalization in noisy 3D range images // 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems. — 2008. — P. 3378–3383.
- [15] Feng Chen, Taguchi Yuichi, and Kamat Vineet R. Fast plane extraction in organized point clouds using agglomerative hierarchical clustering // 2014 IEEE International Conference on Robotics and Automation (ICRA). — 2014. — P. 6218–6225.

- [16] Fischler Martin A. and Bolles Robert C. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography // Commun. ACM. — 1981. — jun. — Vol. 24, no. 6. — P. 381–395. — Access mode: <https://doi.org/10.1145/358669.358692>.
- [17] Fontan Alejandro, Civera Javier, and Triebel Rudolph. Information-Driven Direct RGB-D Odometry. — 2020. — 06. — P. 4928–4936.
- [18] Förstner Wolfgang and Khoshelham Kourosh. Efficient and Accurate Registration of Point Clouds with Plane to Plane Correspondences // 2017 IEEE International Conference on Computer Vision Workshops (ICCVW). — 2017. — P. 2165–2173.
- [19] Gallo Orazio, Manduchi Roberto, and Rafii Abbas. CC-RANSAC: Fitting planes in the presence of multiple surfaces in range data // Pattern Recognition Letters. — 2011. — Vol. 32, no. 3. — P. 403–410. — Access mode: <https://www.sciencedirect.com/science/article/pii/S0167865510003557>.
- [20] Geiger Andreas, Lenz Philip, and Urtasun Raquel. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite // Conference on Computer Vision and Pattern Recognition (CVPR). — 2012.
- [21] Koide Kenji, Yokozuka Masashi, Oishi Shuji, and Banno Atsuhiko. Globally Consistent 3D LiDAR Mapping with GPU-accelerated GICP Matching Cost Factors // IEEE Robotics and Automation Letters. — 2021.
- [22] Helmberger Michael, Morin Kristian, Kumar Nitish, Wang Danwei, Yue Yufeng, Cioffi Giovanni, and Scaramuzza Davide. The Hilti SLAM Challenge Dataset. — 2021. — arXiv:2109.11316.
- [23] Holz Dirk and Behnke Sven. Fast Range Image Segmentation and Smoothing Using Approximate Surface Reconstruction and Region Growing. — 2012. — 06. — Vol. 194.

- [24] Honti Richard, Erdélyi Ján, and Kopacik Alojz. Plane Segmentation from Point Clouds // Pollack Periodica. — 2018. — 08. — Vol. 13. — P. 159–171.
- [25] Honti Richard, Erdélyi Ján, and Kopacik Alojz. Plane Segmentation from Point Clouds // Pollack Periodica. — 2018. — 08. — Vol. 13. — P. 159–171.
- [26] Illingworth J. and Kittler J. A survey of the hough transform // Computer Vision, Graphics, and Image Processing. — 1988. — Vol. 44, no. 1. — P. 87–116. — Access mode: <https://www.sciencedirect.com/science/article/pii/S0734189X88800331>.
- [27] Li Lin, Yang Fan, Zhu Haihong, Li Dalin, Li You, and Tang Lei. An Improved RANSAC for 3D Point Cloud Plane Segmentation Based on Normal Distribution Transformation Cells // Remote Sensing. — 2017. — 05. — Vol. 9.
- [28] Lee Tae-kyeong, Lim Seungwook, Lee Seongsoo, An Shounan, and Oh Se-young. Indoor mapping using planes extracted from noisy RGB-D sensors // 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems. — 2012. — P. 1727–1733.
- [29] Intel Corporation. Intel RealSense Depth Camera D455 // официальный сайт. — Access mode: <https://www.intelrealsense.com/depth-camera-d455> (online; accessed: 16.12.2021).
- [30] Intel Corporation. Intel RealSense LiDAR Camera L515 // официальный сайт. — Access mode: <https://www.intelrealsense.com/lidar-camera-1515> (online; accessed: 16.12.2021).
- [31] Kaess Michael. Simultaneous localization and mapping with infinite planes // Proceedings - IEEE International Conference on Robotics and Automation. — 2015. — 06. — Vol. 2015. — P. 4605–4611.
- [32] Kaiser Adrien, Zepeda Jose Alonso Ybanez, and Boubekour Tamy. A Survey of Simple Geometric Primitives Detection Methods for

Captured 3D Data // Computer Graphics Forum. — 2019. — Vol. 38, no. 1. — P. 167–196.

- [33] Kotthäuser Tobias and Mertsching Bärbel. Triangulation-Based Plane Extraction for 3D Point Clouds // Intelligent Robotics and Applications / ed. by Su Chun-Yi, Rakheja Subhash, and Liu Honghai. — Berlin, Heidelberg : Springer Berlin Heidelberg. — 2012. — P. 217–228.
- [34] Geneva Patrick, Eckenhoff Kevin, Yang Yulin, and Huang Guoquan. LIPS: LiDAR-Inertial 3D Plane SLAM. — 2018. — 10.
- [35] Li Jun, Qian Fei, and Chen Xianfu. Point Cloud Registration Algorithm Based on Overlapping Region Extraction // Journal of Physics Conference Series. — 2020. — Sep. — Vol. 1634 of Journal of Physics Conference Series. — P. 012012.
- [36] Limberger Frederico A. and Oliveira Manuel M. Real-time detection of planar regions in unorganized point clouds // Pattern Recognition. — 2015. — Vol. 48, no. 6. — P. 2043–2053. — Access mode: <https://www.sciencedirect.com/science/article/pii/S0031320315000072>.
- [37] Microsoft. Azure Kinect DK // официальный сайт. — Access mode: <https://azure.microsoft.com/en-us/services/kinect-dk> (online; accessed: 16.12.2021).
- [38] Microsoft. Xbox One Kinect Sensor v1 // сайт распространителя. — Access mode: <https://www.amazon.com/Microsoft-360-Kinect-Sensor-Certified-Refurbished/dp/B005GA1H4C> (online; accessed: 16.12.2021).
- [39] Nathan Silberman Derek Hoiem Pushmeet Kohli and Fergus Rob. Indoor Segmentation and Support Inference from RGBD Images // ECCV. — 2012.
- [40] Barnes Dan, Gadd Matthew, Murcutt Paul, Newman Paul, and Posner Ingmar. The Oxford Radar RobotCar Dataset: A Radar

- Extension to the Oxford RobotCar Dataset // Proceedings of the IEEE International Conference on Robotics and Automation (ICRA). — Paris. — 2020. — Access mode: <https://arxiv.org/abs/1909.01300>.
- [41] Zhang Xiaoyu, Wang Wei, Xianyu Qi, Ziwei Liao, and Wei Ran. Point-Plane SLAM Using Supposed Planes for Indoor Environments // Sensors. — 2019. — 09. — Vol. 19. — P. 3795.
- [42] Zhang Xiaoyu, Wang Wei, Xianyu Qi, Ziwei Liao, and Wei Ran. Point-Plane SLAM Using Supposed Planes for Indoor Environments // Sensors. — 2019. — 09. — Vol. 19. — P. 3795.
- [43] Proença Pedro F. and Gao Yang. Fast Cylinder and Plane Extraction from Depth Cameras for Visual Odometry // 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). — 2018. — P. 6813–6820.
- [44] Qian Xiangfei and Ye Cang. NCC-RANSAC: A fast plane extraction method for 3-D range data segmentation // IEEE transactions on cybernetics. — 2014. — 04. — Vol. 44.
- [45] Quanergy Systems Inc. S3 Solid State LiDAR Sensor // официальный сайт. — Access mode: <https://quanergy.com/products/s3> (online; accessed: 16.12.2021).
- [46] Xu Jinxuan, Xie Qian, Chen Honghua, and Wang Jun. Real-Time Plane Detection with Consistency from Point Cloud Sequences // Sensors. — 2021. — Vol. 21, no. 1. — Access mode: <https://www.mdpi.com/1424-8220/21/1/140>.
- [47] Roychoudhury Arindam, Missura Marcell, and Bennewitz Maren. Plane Segmentation in Organized Point Clouds using Flood Fill // 2021 IEEE International Conference on Robotics and Automation (ICRA). — 2021. — P. 13532–13538.
- [48] Sun Pei, Kretzschmar Henrik, Dotiwalla Xerxes, Chouard Aurelien, Patnaik Vijaysai, Tsui Paul, Guo James, Zhou Yin, Chai Yuning,

- Caine Benjamin, et al. Scalability in perception for autonomous driving: Waymo open dataset // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. — 2020. — P. 2446–2454.
- [49] Schnabel Ruwen, Wahl Roland, and Klein Reinhard. Efficient RANSAC for point-cloud shape detection // Comput. Graph. Forum. — 2007. — 06. — Vol. 26. — P. 214–226.
- [50] Cadena Cesar, Carlone Luca, Carrillo Henry, Latif Yasir, Scaramuzza Davide, Neira Jose, Reid Ian, and Leonard John. Simultaneous Localization And Mapping: Present, Future, and the Robust-Perception Age // IEEE Transactions on Robotics. — 2016. — 06. — Vol. 32.
- [51] Skoltech. Skoltech Mobile Robotics // официальный сайт. — Access mode: <https://sites.skoltech.ru/mobilerobotics> (online; accessed: 16.12.2021).
- [52] Stereolabs Inc. Stereolabs ZED 2 // официальный сайт. — Access mode: <https://www.stereolabs.com/zed-2/> (online; accessed: 16.12.2021).
- [53] Trevor Alexander J. B., Rogers John G., and Christensen Henrik I. Planar surface SLAM with 3D and 2D sensors // 2012 IEEE International Conference on Robotics and Automation. — 2012. — P. 3041–3048.
- [54] Vaufreydaz Dominique and Nègre Amaury. MobileRGBD, An Open Benchmark Corpus for mobile RGB-D Related Algorithms // 13th International Conference on Control, Automation, Robotics and Vision. — Singapour, Singapore. — 2014. — Dec. — Access mode: <https://hal.inria.fr/hal-01095667>.
- [55] Velodyne Lidar Inc. Velodyne’s Puck lidar sensor // официальный сайт. — Access mode: <https://velodynelidar.com/products/puck> (online; accessed: 16.12.2021).

- [56] Geiger Andreas, Lenz Philip, Stiller Christoph, and Urtasun Raquel. Vision meets Robotics: The KITTI Dataset // International Journal of Robotics Research (IJRR). — 2013.
- [57] Wang Lin, Cao Jianfu, and Han Chongzhao. Multidimensional particle swarm optimization-based unsupervised planar segmentation algorithm of unorganized point clouds // Pattern Recognition. — 2012. — 11. — Vol. 45. — P. 4034–4043.
- [58] Wang Tingqi, Chen Lei, and Chen Qijun. A graph-based plane segmentation approach for noisy point clouds // 2013 25th Chinese Control and Decision Conference (CCDC). — 2013. — P. 3770–3775.
- [59] Wasenmuller Oliver, Meyer Marcel, and Stricker Didier. CoRBS: Comprehensive RGB-D Benchmark for SLAM using Kinect v2 // IEEE Winter Conference on Applications of Computer Vision (WACV) / IEEE. — 2016. — March. — Vol. . — P. . — Access mode: <http://corbs.dfki.uni-kl.de/>.
- [60] Waymo LLC. Waymo // официальный сайт. — Access mode: <https://waymo.com/> (online; accessed: 16.12.2021).
- [61] Wu Teng, Hu Xiangyun, and Ye Lizhi. Fast and Accurate Plane Segmentation of Airborne LiDAR Point Cloud Using Cross-Line Elements // Remote Sensing. — 2016. — Vol. 8, no. 5. — Access mode: <https://www.mdpi.com/2072-4292/8/5/383>.
- [62] Plane Detection in Point Cloud Data : WorkingPaper / University of Bonn ; executor: Yang Michael Ying and Förstner Wolfgang : 2010. — P. 1–16.
- [63] Zivid. Zivid One+ Small // официальный сайт. — Access mode: <https://www.zivid.com/zivid-one-plus-small-3d-camera> (online; accessed: 16.12.2021).
- [64] Fontana Simone, Cattaneo Daniele, Ballardini Augusto L., Vaghi Matteo, and Sorrenti Domenico G. A benchmark for point clouds

registration algorithms // Robotics and Autonomous Systems. — 2021. — Vol. 140. — P. 103734. — Access mode: <https://www.sciencedirect.com/science/article/pii/S0921889021000191>.

- [65] Hoover A., Jean-Baptiste G., Jiang X., Flynn P.J., Bunke H., Goldgof D.B., Bowyer K., Eggert D.W., Fitzgibbon A., and Fisher R.B. An experimental comparison of range image segmentation algorithms // IEEE Transactions on Pattern Analysis and Machine Intelligence. — 1996. — Vol. 18, no. 7. — P. 673–689.