МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования   
**«Национальный исследовательский   
Нижегородский государственный университет им. Н.И. Лобачевского»**

**(ННГУ)**

**Институт информационных технологий, математики и механики**

**Кафедра: Программной инженерии**

Направление подготовки: «Фундаментальная информатика и информационные технологии»

Профиль подготовки: «Инженерия программного обеспечения»

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА БАКАЛАВРА**

**Тема:**

**«Система слежения за объектом в облаке точек»**

Допущена к защите Выполнил:

Заведующий кафедрой: студент группы 081406-2

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Хачатрян Корюн Агаронович

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ подпись подпись

Научный руководитель:

Доцент кафедры программной инженерии Кандидат физико-математических наук

Шапошников Дмитрий Евгеньевич

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

подпись

Нижний Новгород  
2018

Аннотация

В данной работе рассматривается метод решения задачи слежения за объектом в облаке точек, полученная в ходе лазерной съемки. Для решения задачи используется метод основанный на обнаружении особых точек объекта кадра и расчета оптического потока между кадрами. Система реализована с использованием языка С++. Для визуализации данных используется библиотека OpenSceneGraph, реализующий паттерн графа сцены с использованием библиотеки OpenGl. Проводятся также измерения метрик эффективности на облаке точек из набора данных ”The KITTI dataset”.

Оглавление

Введение

1. **Слежение за объектами**
   1. Методы основанные на предсказании
   2. Метод основанный на распознавании
2. **Метод основанный на распознавании**
   1. Поиск особенностей
   2. Описание особых точек
   3. Сопоставление
3. **Реализация**
4. **Описание экспериментальной части**

Заключение

Введение

Одной из новых активно-развивающихся областей являются автономные транспортные средства, включая подводные, наземные (роботы, машины), воздушные. Уровень автономности изменяется от полностью автономных (беспилотных) до транспортных средств, где системы, основанные на компьютерном зрении, поддерживают водителя или пилота в различных ситуациях. Такие транспортные средства используют компьютерное зрение для навигации, то есть для получения информации о месте своего нахождения, для создания карты окружающей обстановки, для обнаружения препятствий. Примерами таких систем могут быть система предупредительной сигнализации о препятствиях на машинах и системы автономной посадки самолетов. Некоторые производители машин демонстрировали системы автономного управления автомобилем, но эта технология все ещё не достигла того уровня, когда её можно запустить в массовое производство.

Цифровые данные получаются от одного или нескольких датчиков изображения, и которые помимо различных типов светочувствительных камер включают датчики расстояния, радары, ультразвуковые камеры и т. д. В зависимости от типа датчика, получающиеся данные могут быть обычным 2D изображением, 3D изображением или последовательностью изображений. Значения пикселей обычно соответствуют интенсивности света в одной или нескольких спектральных полосах (цветные или изображения в оттенках серого), но могут быть связаны с различными физическими измерениями, такими как глубина, интенсивность и др. Данная информация с сенсоров передается в систему анализа дорожной обстановки, которая и отдает команды системе управления автомобилем. В таких системах зачастую применяются алгоритмы машинного обучения, для распознавания окружающей обстановки. Например для автономных автомобилей распознаются другие участники дорожного, дорожные знаки, состояние окружающей среды, дорожную обстановку и т.д. Для их обучения требуется большое количество разнообразных данных. Обычно собирается большое количество «сырых» данных, таких как видео с камер, серия кадров с лазерного сканера , датчики геопозиционирования , скорости и многих других. Но для обучения нейронных сетей не подходят данные в необработанном виде. Например чтобы научить нейросетей распознавать машины , на вход при обучении ему необходимо подать фотографии только машин, там где доля фона очень мала , иначе обучение пройдет неудачно. Для их получения происходит разметка исходных данных, о есть выделение информативных участков на кадре.

Постановка задачи

Для обучения нейронных сетей требуется большое количество данных, но для качественного обучения они нужны быть разнообразными и информативными. По этой причине необходимо большое количество размеченных данных полученных в различной обстановке. Обычно для разметки данных объекты выделяются в ограничивающее окно. Например для 2D данных, в частности изображений, это ограничивающий прямоугольник, внутри которого содержится необходимое подпространство исходных данных без лишнего фона. Такая разметка одного и того же объекта от кадра к кадру занимает много времени. Чтобы это ускорить используется технология слежение за объектом в потоке данных. От кадра к кадру высчитывается новое положение объекта в пространстве, куда смешается ограничивающее окно. Такой подход дает значительный прирост скорости разметки данных. Заметим, что разметка облака точек занимает значительно больше времени чем разметка изображений.

В рамках данной работы планируется разработать и создать систему для слежения за объектом в облаке точек , то есть 3D данных.

Поставленные задачи:

* Исследование существующих методов слежения за выделенным объектом в потоке точек полученных с лидаров.
* Реализация оптимального метода решения задачи слежения за выделенным объектом в потоке кадров облака точек.
* Визуализация обрабатываем данных, а так же промежуточных данных используемых в алгоритме с помощью графической библиотеки.
* Исследование эффективности метода на реальных данных.

Облако точек

В описанной выше задаче потоком информации является серия кадров содержащих множество точек (размерность точки зависит от самого лидара) полученное с помощью лазерного сканирующего устройства- лазерный радар ( в дальнейшем лидара). **LIDAR** *(Light Identification Detection and Ranging)* — обнаружение, идентификация и определение дальности с помощью света - дословный перевод. Сканирующие лидары в системах машинного зрения формируют двумерную или трёхмерную картину окружающего пространства. Рассмотрим принцип действия лидара: объект, точнее его поверхность, освещается лазерным импульсом, и измеряется время, через которое сигнал вернется к источнику и зафиксируется датчиком. Зная координаты сканера и направление импульса, можно определить трехмерные координаты точки, от которой импульс отразился. Время отклика прямо пропорционально расстоянию до цели. Кроме импульсного метода измерения дистанции применяется фазовый, основанный на определении разности фаз посылаемых и принимаемых модулированных сигналов.

Световые волны подвержены рассеянию в любых средах, в том числе в воздухе, поэтому возможно не только определять расстояние до непрозрачных дискретных целей, но и фиксировать интенсивность рассеивания света в прозрачных средах. Свет движется с постоянной и известной скоростью, поэтому лидар может вычислить расстояние между ним и цели с высокой точностью.

Таким образом выходными данными лидара является набор точек пространства, заданными 3 координатами *(x,y,z) рис.1*, а так же датчиками могут фиксироваться и другие характеристики поверхности от которой отразился луч , например коэффициент отражения *(reflectance)*. Но в зависимости от лидаров количество информации об отраженном сигнале , то есть и о поверхности отражения, может отличаться.

В качестве данных для тестирования использовался *”Kitti dataset”*[1]. полученных с универсала VW для использования в мобильной робототехнике и исследованиях автономного вождения. В данных содержится различные сценарии дорожного движения на частоте 10-100 Гц с использованием различных сенсоров, таких как цветные и черно-белые стереокамеры, лазерный сканер Velodyne 3D и высокоточная система инерциальной навигации GPS / IMU. Каждый кадр полученный с лазерного сканера содержит приблизительно 120.000 точек, где каждая точка содержит координату и коэффициент отражения *(x,y,z,r)*.

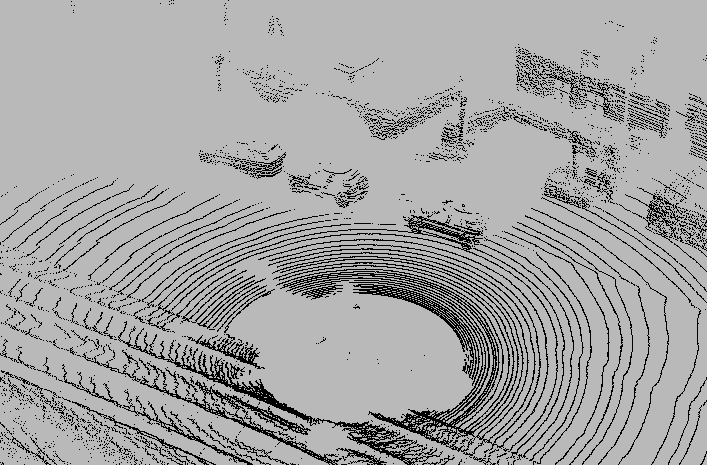


Рисунок 1. Кадр полученный с лидара , KITTI dataset.

1 Слежения за объектами

Задача слежения за объектом(**object tracking**) — отслеживание траектории движения одного или нескольких объектов, в потоке информации. В задаче классического компьютерного зрения под задачей слежения за объектом понимается, сопровождение объекта — некоторой области изображения в видео , то есть в потоке изображений. Результатом в задаче слежения является траектория движения объекта , т.е. множество позиций в которых он находился .

Методы основанные на предсказании

Идея данных методов заключается в предсказании будущего положения объекта по информации накопленной на предыдущих кадрах. Пусть на кадре ***#0*** объект ***Object*** имел положение . Методы данного типа оценивают вектор смещения *v* , такой что объект на кадре **#1** будет иметь положение . Обычно такие методы вычисляют смещение кадра относительно предыдущего кадра.

1. Оптический поток - векторное поле явного движения объектов , поверхностей и ребер в визуальной сцене между кадрами, вызванное относительным движением между наблюдателем и сценой. цены. Для каждой точки одного кадра необходимо вычислить вектор смещения текущего кадра к следующему. Можно определить область движения, приписав каждой точке вектор скорости. В некоторый момент времени точка p’ на изображении соответствует некоторой точке ***p*** . Эти две точки связаны между собой уравнениями. Точка ***p*** перемещается относительно наблюдателя (камеры, глаза) со скоростью ***v*** . Это перемещение порождает движение со скоростью ***v’*** соответствующей точки изображения ***p’***. За время точка *p* перемещается на расстояние ***vt*** , а точка изображения ***p’*** — на расстояние ***v’t*** [12]. Плотный оптический поток: эти алгоритмы помогают оценить вектор движения каждой точки в кадре. Разряженный оптический поток: эти алгоритмы, такие как Kanade-Lucas-Tomashi [10,11], отслеживают расположение нескольких точек в кадре, обычно небольшая окрестность вокруг точки слежения, например такими точками могут являться ключевые точки кадра . Минус данных методов том , что предположение о равнозначности двух соответствующих точек не всегда выполняется, к тому же метод опирается на дифференцируемость , что говорит о требовании малого смещения. Оптический поток для облака точек описан в [13].
2. Фильтры частиц. Kalman Filtering: алгоритм обработки сигналов , используемый для прогнозирования местоположения движущегося объекта на основе предшествующей информации о движении, так же использует информацию о скорости движения объекта.

Метод основанный на распознавании

Такие методы так же называются генеративными или прямыми методами. Отслеживание можно рассматривать как серию детектирования. Предположим что в кадре неким образом описан объект *Object,* и задана некоторая позиция *.* Отслеживанием объекта *Object* будет считаться поиск позиции на кадре . То есть необходимо найти объект *Object* в кадре и определить его позицию.Данную схему можно изложена в схеме на *Рисунке 2* .После того как объект найден в новом кадре информация о нем может быть скорректирована. Рассмотрим пример движущегося трафика и отслеживание в нем автомобиля. На первом кадре видео мы можем выделить объект , описать его неким образом. Затем по существующему описанию объекта найти его на следующем кадре и определить его позицию или вектор смещения. Общий алгоритм можно описать следующим образом:

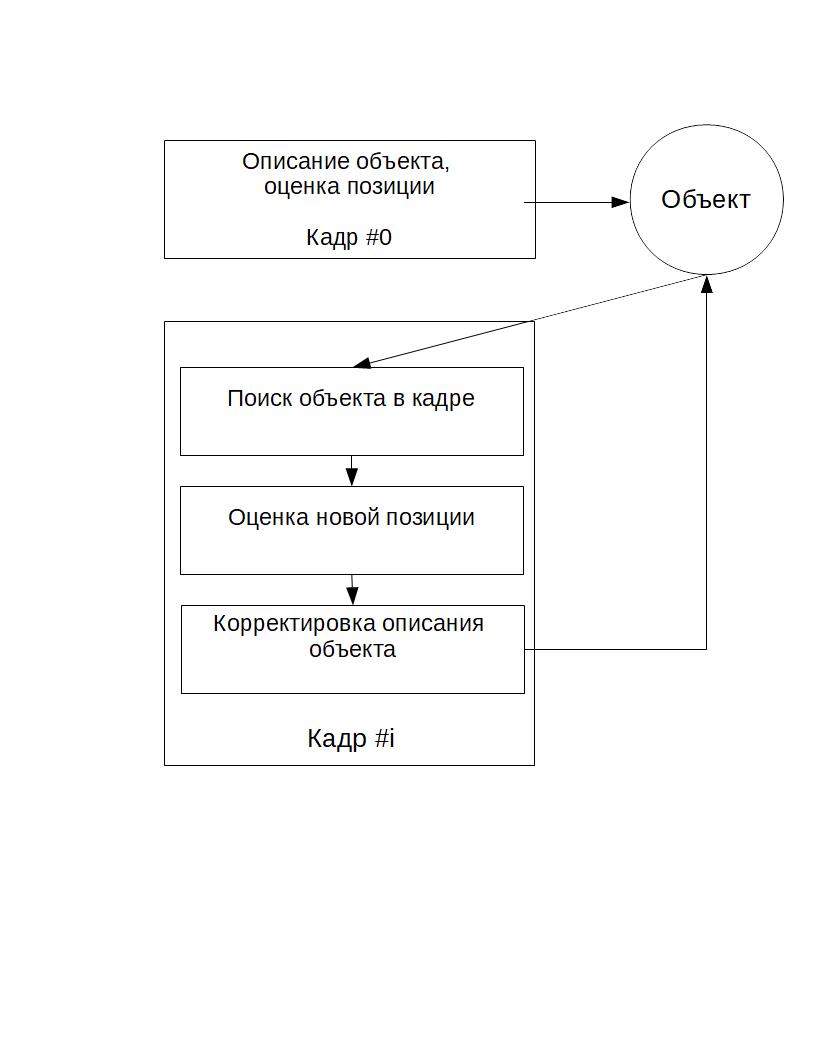
* На вход подается кадр , с выделенным в нем объектом *Object* имеющий позицию *p*.
* Описание объекта математическими признаками *D*.
* На вход подается следующий кадр и и производится поиск объекта по описанию *D*. Например, описывается весь кадр множеством описаний и cреди них ищется наиболее близкое к D описание.
* Оценивается новая позиция объекта *Object* в кадре .

Рисунок 2. Схема метода основанного на распознавании.

* Корректировка описания *D*

2 Метод основанный на распознавании

Как было описано выше , в основе данного метода лежит распознавание объекта в сцене. В данной задаче наиболее применимы методы :

* Метод основанный на воксельной модели. Наиболее известный из таких методов метод моментов. Данный метод похож на метод моментов для изображения , только в качестве бинарного изображения выступает воксельная модель пространства облака точек. Наличие в вокселе точки равносильно наличию области в пикселе. Такой метод описан в работе [].
* Метод основанный на поиске особенностей. Данный вариант будет подробно описан ниже.

2.1 Особенности

Работа с полным набором данных может быть очень трудоемким процессом, если рассматривать работу с изображением , то необходимо обрабатывать все пиксели , а при работе с облаком точек все точки пространства. При этом , далеко, не все данные могут быть полезны, некоторые могут пропасть, попасть под действием шума или просто измениться, то есть дин и тот же объект на разных кадрах может представляться различным набором данных. На разным изображениях видео у одного объекта может измениться масштаб , яркость и другие характеристики. В облаке точек может измениться набор точек описывающий его контур, однако, стоит заметить что сами черты , который задает этот контур изменяется мало.

Для решения данной проблемы стоит работать только с данными которые бы обладали некими свойства от кадра к кадру:

* **Отличимость**– данные должны явно выделяться на фоне остальной выборки и быть отличим в своей окрестности.
* **Инвариантность**– определенные данные должно быть независимо к различным преобразованиям.
* **Стабильность**– данные должно быть устойчиво к шумам и о ошибкам.
* **Уникальность(локальность)**– кроме локальной отличимости, данные должны обладать глобальной уникальностью для улучшения различимости повторяющихся, несовпадающих признаков.
* **Интерпретируемость**– данные должны определяться так, чтобы их можно было использовать для анализа соответствий и выявления интерпретируемой информации из кадра.

Точки обладающие такими свойствами назовем **особыми точками кадра**[2].

**Детектор (detector)**– алгоритм извлечения особых точек из кадра. Детектор должен обеспечивать инвариантность нахождения одних и тех же особых точек в разных кадрах, и с учетом трансформаций, которым подвергся объект.

Опираясь на принцип локальности можно утверждать что признак по которому мы выбираем ключевую точку должен быть локальным экстремумом, обычно, операторы, с помощью которых выделяются эти точки, определяют локальный максимум.

Ключевые точки различают по признакам с помощью которых извлекаются точки [3]:

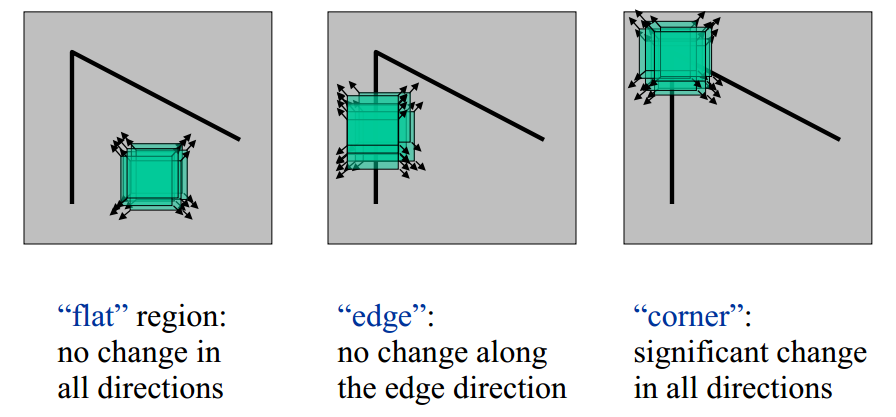
* Контурные методы — методы в которых из кадра выделяются контуры объектов , а затем происходит анализ полученных кривых. В таких методах описанием объекта может быть некое математическое описание контуров (изгибы, кривизну и т.д) из которых состоит объект. Такие методы сильно зависят от предварительного шага выделения контура и описание их кривыми. К примеру работы [5] авторы указали, что эта величина тесно связана с кривизной в вершине. Также Хо и Гиббинс [8] предложили меру, называемую кривизной, чтобы описать геометрическую информацию в вершине.
* Параметрические методы — методы параметрически описывающие поверхность , и исследующие их как математические функции. Методы, основанные на конформной параметризации в [10, 20], преобразуют трехмерную поверхность в двумерную параметризацию, которая может рассматриваться как изображение. Затем на нем могут быть обнаружены точки интереса. 3D-2D-отображение называется конформным, если углы сохраняются. После всех манипуляций обратно следует преобразование в 3D.
* Методы основании на изменениях локальной окрестности точки — методы работающие с окрестностью точки , и определяющие максимум изменения некоторых характеристик. В методах обработки изображений вычисляется изменение интенсивности , и на основах вычислений определяется является ли данная точка или окрестность особой . Многие работы основываются на классическом SIFT детекторе, например в [15] применили детектор DoG по вершинам в масштабе, и вершины с высокими откликами посчитали ключевыми. В основном такие методы можно разделить по типу найденных особенностей:
  + Капли (blobs) – или области, то есть особенности которые являются областями. Обеспечивают дополнительное описание структур изображения по областм. Тем не менее, детекторы таких особенностей часто могут содержать точку интереса (локальный максимум реакции оператора), что означает, что многие детекторы такого типа также могут рассматриваться как операторы особых точек.
  + Края (edges) — точки на границе между двумя областями. В общем случае ребро может иметь почти произвольную форму и может включать в себя переходы. На практике ребра на изображении определяются как множества точек изображения, которые имеют сильную градиентную величину, такие алгоритмы обычно устанавливают некоторые ограничения на свойства ребра, такие как форма, гладкость и значение градиента. Локально ребра имеют одномерную структуру, то есть вдоль ребра изменение не наблюдается , при переходе через ребро , происходит переход из одной области в другую , что сопровождается изменением характеристик.
  + Углы (углы) — углы точечным особенностям кадра. Среди таких алгоритмов обычно встречаются те что ищут углы в прямом смысле, то есть места пересечения двух краев, там где изменение локальных характеристик наблюдается по всем направлениям. Затем эти алгоритмы были разработаны таким образом, чтобы явное обнаружение краев больше не требовалось, например, путем поиска высоких уровней кривизны в градиенте изображения. Затем было замечено, что так называемые углы были также обнаружены на участках изображения, которые явно углами не являются.

Детекторы точек интереса для изображений достигли высокой эффективности. Причина в том, что структура изображения четко определена, а точки интереса соответствуют пикселям, которые представляют структуры на изображении. Но структура данных изображения и облака точек сильно различны:

* + Прямого применения методов поиска изображений невозможно, из-за различия в структуре данных. Облако точек можно представить как трехмерный дискретный контур объекта с . Даже как контур, он может покрывать не весь объект , а только сторону с которой происходила съемка.
  + Отличается понятие надежности помимо стандартный трансформаций, таких как перемещение, поворот масштабирование, на трехмерные объекты может действовать изменение изометрии , топологии , тесееляции ,а так же понижение дискретизации.

Рассмотрим классический вариант детектора описанный Харрисом и Стивенсоном в [4] для изображений , а затем расширение этого метода для работы с 3D данными описанного в [5]. Метод является популярным из-за его сильной инвариантности к вращению, масштабу, изменению освещенности и шуму изображения.

Детектор Харриса основан на оценке локального изменения признака вокруг точки. Идея метода заключается в том что находясь в области угла градиент изображения изменяется по всем направлениям, при этом он слабо изменяется при движении вдоль граней и совершенно не изменяется в монотонной области. На *Рисунке 3* продемонстрирована идея поиска точек в детекторе Харриса*.*

Рисунок 3. Демонстрация идем работы детектора Харриса.

Рассмотрим процесс более подробно. Исходя из идеи анализируется функция изменения интенсивности определяющаяся как:

где обозначает функцию изображения, а — координаты точек в гаусовой функции , которая обозначена как ***W***. Суммирование происходит по всем , что задает смещение во всех направлениях. Это можно видеть если переписать выражение в векторном виде.

Используя разложение Тейлора, до второй степени и запись в матричном виде получается:

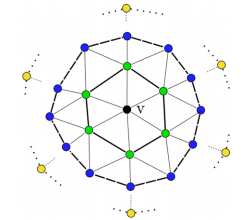
,где ***S*** как раз является вектором смещения. – производная по направлению ***x***, а производная по направлению ***y.*** Для получения численной оценки окрестности Харрис и Стивенс предложили проанализировать собственные значения матрицы ***E***, которая содержит достаточную локальную информацию, связанную с структурой окрестности. Матрица ***E*** – называется автокорреляционной (или оператором Харриса), по ее собственным значениям определяется является ли точка ключевой. Если оба собственных значения малы , то точка не является углом . Если первое собственное число близко к нулю , а второе большое , то это точка лежащая на ребре. Если оба собственных числа большие, то точка является углом. Однако, чтобы избежать дорогостоящего вычисления собственных значений, они предложили присвоить каждому пикселю изображения следующее значение:

,гдe ***k –*** константа, экспериментально установлен равным 0.04.

Понятно что исходный метод не подходит для определения особой в облаке точек, который представлен набор точек ***(x,y,z)*** , и необходима адаптация оператора.

В работе Саприна и Бастона рассматривается алгоритм адаптирующий детектор Харриса для поиска точек интереса в 3D. Схема работы алгоритма представлена на *Рисунке 5* Опишем алгоритм ниже:

* Для каждой точки v рассматривается ее окрестность , где *k* – число колец вокруг точки *v*. На *Рисунке 4* показана рассматриваемая точка покрашенная в черный и так же первая окрестность , точки которой окрашены в зеленый , вторая окрестность , точки которой окрашены в синий. Все эти точки являются ближайшими соседями к точке v.

Рисунок 4. Исследуемая точка v и ее окретность.

* Вычисляется центр масс окрестности где каждая точка относящаяся имеет одинаковый вес. Далее весь набор точек переносится так , чтобы центр масс находился в начале координат.
* Отыскивается наилучшая плоскость аппроксимирующая данный набор точек. Например применив метод наименьших квадратов, минимизирующую расстояние от каждой точки до проекции этой точки на плоскость. Или можно применить анализ главных компонентов к множеству точек и выбираем собственный вектор с наименьшим собственным значением как в качестве нормали к искомой плоскости.
* Вращение набор точек так, что нормаль плоскости совпала с осью *OZ*. И система точек переносится так чтобы точка v стала центром системы координат.
* Для вычисления производных сопоставляется квадратичная поверхность множеству преобразованных точек. Используя метод наименьших квадратов, получается параболоид вида:

Кроме того, такая поверхность позволяет просто вычислить производные.

* Для получения устойчивости к шуму авторы предлагают применить гауссову функцию. Однако возникает трудность, поскольку в исходном выражении производные являются дискретными функциями, а в полученном производные являются непрерывными функциями. Для решения этой задачи предлагается интегрирование производных с непрерывной гауссовой функцией следующим образом:

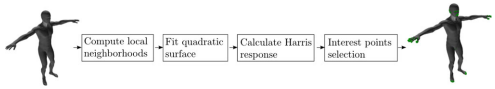
где σ - дисперсия гауссовой функции

* Подставив значение функции и вычислив интеграл получается:

Результатом работы вышеуказанных детекторов является множество особых точек, в частности, углов.

Оператор Харриса определяется как:

Отклик матрицы вычисляется способом описанным выше.

 Рисунок 5. Схема работы детектора Харриса для 3D.

2.2 Описание особой точки

Простой работы с особыми точками не достаточно. Хоть точки и отличимы локально , но глобально они не отличаются от других особых точек. Чтобы получить возможность их каким либо образом идентифицировать и сравнить между собой необходимо некое математическое описание точки. Обычно описание основывается на информации об окрестности точки.

**Дескриптор(descriptor)** – идентификатор, описание особой точки. Некоторый набор данных, основанный на информации об окрестности точки , который описывает данную ключевую точку и позволяет отличить ее от других точек. Дескрипторы должны обеспечивать инвариантность нахождения соответствия между особыми точками относительно преобразований кадра.

Понятно что дескриптор одной ключевой точки на разных кадрах должен совпадать или незначительно отличаться.

Описание точки может представлять из себя вектор заполненный извлеченной из окрестности информации, в таком случае его можно рассмотреть как точку в многомерном пространстве , и определять схожесть двух дескрипторов по расстоянию между этими точками .

Гистограмма особенностей точек (Point Feature Histograms)

Идея алгоритма основывается на том, что при трансформации объект или некоторые его части сохраняют свою форму, и поэтому нормали в таких областях будут похожи. Метод работает с точками (x,y,z), и описывает точку исходя из нормалей поверхности, образованной окрестностью данной точки. Рассмотрим алгоритм подробнее:

* Задается радиус *r* и рассматривается окрестность данного радиуса точки *p* -
* Для каждой точки из данной окрестности оценивается нормали соответствует точке (с помощью анализа главных компонент).
* Полученные нормали переориентируются используя *v* точку откуда производилась съемка.
* Для каждой пары точек и *(i,j=1-6, j <i)* в окрестности и соответствующих нормалей и мы выбираем точку и точку , причем является тот, который имеет меньший угол между соответствующей нормалью и линией, соединяющей точки *Рисунок 6*:

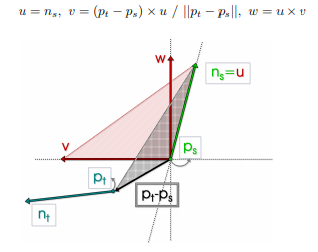
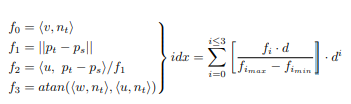


Рисунок 6. Вычисление векторов для функций дескриптора.

* Гистограмма для точки *p* будет вычисляться с использованием четырех геометрических признаков:

,где [] - оператор обозначает операцию целочисленной части, d - количество подразделений максимального теоретического диапазона характеристик (fimax - fimin), а idx - индекс гистограммного бина, в который падает пара точек .

* Значение бина увеличивается на 1 и нормализуется каждый бин с общим числом пар точек () для достижения точечной инвариантности плотности.

Количество бинов в гистограмме, которые могут быть сформированы с использованием этих четырех геометрических признаков — d^4. Авторы эмпирически получили хорошие результаты, разделив значения признаков на три части d = 3, поэтому общей сложности 81 бин (количества комбинаций между четырьмя функциями). Как замечают авторы , еще большее увеличение количества бинов не улучшило результаты значительно, а количество вычислений увеличивается экспоненциально. Использование более 3 подразделов приведет к большому количеству дополнительных вычислений для каждой точки (например, 256D).

Можно сказать что данный алгоритм обобщает среднюю кривизну в точке которую описывает.

2.3 Сопоставление

Имея особую точку и ее описание можно сопоставлять точки объекта с точками сцены где мы объект ищем. Ключевые точки описаны вектором, и можно их рассматривать как точки в многомерном пространстве , соответственно можно найти расстояние между ними. Точку можно сравнить с точкой : , где -метрика.

Чаще всего используются метрики:

Sum of Absolute Differences (SAD) где i итерация по всем компонентам

Sum of Squared Differences (SSD) где i итерация по всем компонентам

Каждой особой точке сцены сопоставляем особую точку объекта. Таким образом одной ключевой точке объекта может соответствовать несколько ключевых точек сцены среди них предпочтительнее выбирать самую маленькую.

3 Реализация

При реализации системы слежения использовался метод прямого трекинга о котором изложено выше. Для нулевого кадра , содержащего облако точек, строится дерево пространственного разбиения. Происходит выделении подпространства , содержащего точки объекта, с помощью прямоугольника *(x,y,z,w,l,h)* (Рисунок №). Благодаря построенному дереву, построенному на предыдущем шаге, поиск точек входящих в область прямоугольника(т.е. объект слежения) занимает небольшое время. После,к полученному множеству точек применяется детектор углов Харриса и на выходе получается множество ключевых точек . Данный шаг нужен для сокращения объема точек с помощью которых мы затем опишем наш объект. Затем для каждой ключевой точки *kp с* помощью дескриптораописывается окрестность и на выходе получается описание для точки . После данного шага мы имеем множество векторов, описывающих объект.

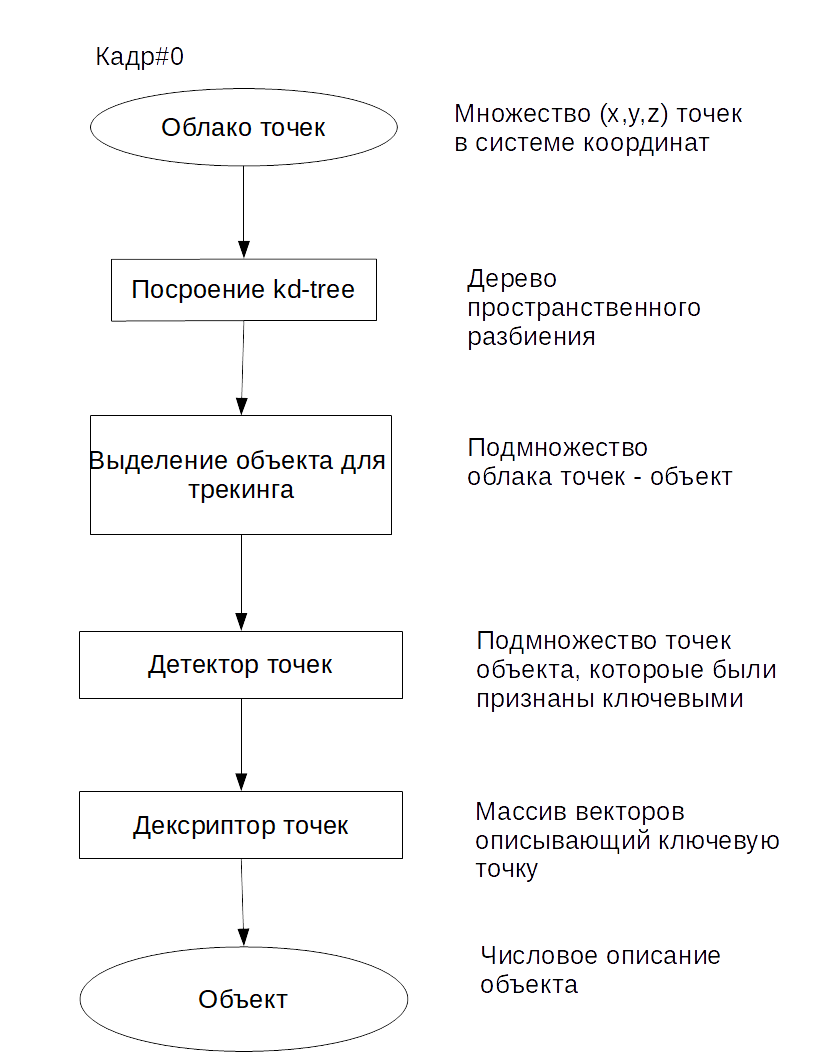
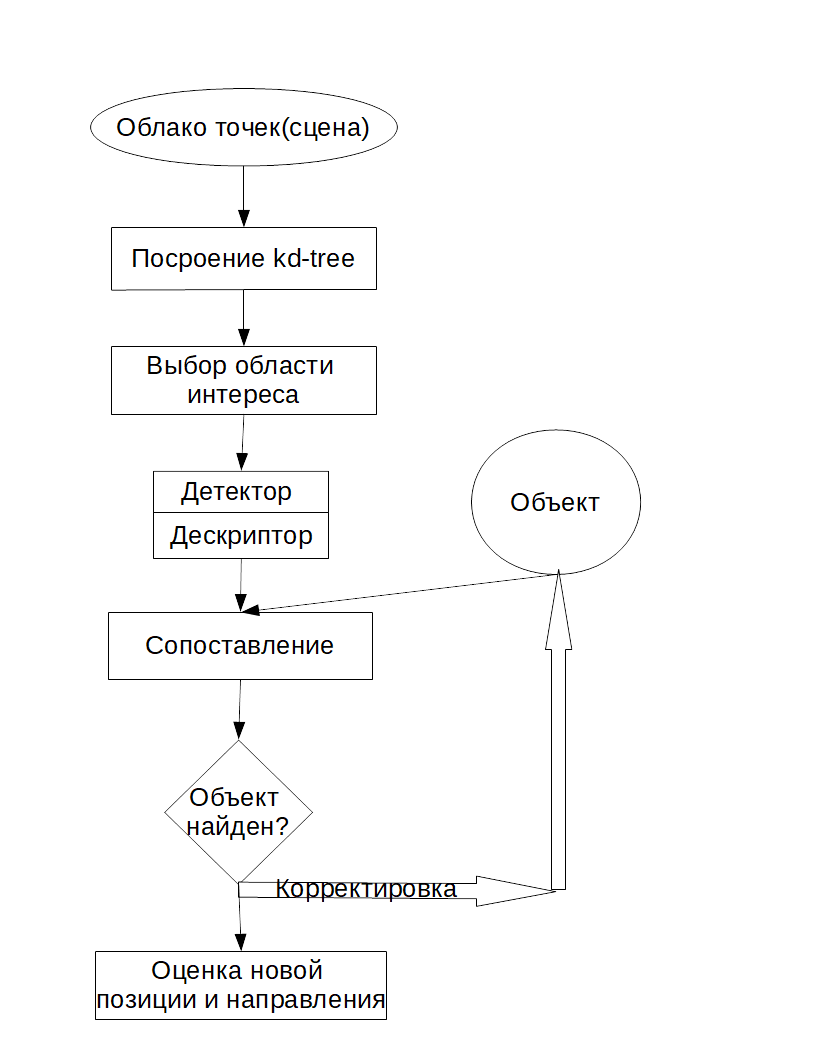


Рисунок 7. Схема работы на нулевом кадре.

На *Рисунке 7* представлена схема предварительной обработки на кадре #0:

После того как объект описан можно приступать к поиску объекта на сцене и оценки его позиции. На *Рисунке 8* представлена общая схема работы алгоритма слежения.

 Рисунок 8. Схема работы алгоритма слежения.

После построение дерева пространственного разбиения по предыдущей позиции объекта оценивается область интереса в которой будет происходить поиск объекта. После предварительного шага мы имеем только позицию объекта и не имеем информации о направлении движения, поэтому поиск будет происходить только в небольшой окрестности ограничивающего прямоугольника. Интересующая область будет представлять подмножество облака точек, в ней и будет происходить первичный поиск объекта. При отсутствии объекта в данной области поиск будет происходить по всему кадру. Сопоставление точек происходит по описанному выше алгоритму.

После сопоставления мы имеет соответствие между подмножеством точек объекта и подмножеством точек сцены .

Выбор области интереса

Операция поиска ключевых точек и их описания трудоемкий процесс, поэтому при описании сцены нужно избегать работы со всеми точками, для этого желательно обрабатывать некоторое подмножество сцены. Решает данную проблему попытка предсказания области интереса. Будем пытаться предсказывать новую позицию объекта и искать в области около нее . На первом шаге когда еще не известна скорость объекта, произведем попытку поиска в окрестности позиции p, где *w,l* – ширина и длинна ограничивающей области . Окрестность равная 3w и 3l в ширину и длину соответственно. После оценки новой позиции *p’* объекта на следующем кадре , мы можем вычислить скорость *v = p’ - p* смещения объекта на данном кадре, где p – позиция объекта на предыдущем кадре. После чего *p=p’*. Поиск в следующем кадре будет производиться в окрестности точки *p + v* , то есть мы предполагаем , что объект движется с постоянной скоростью.

Корректировка

Так как со временем множество точек регистрируемых лидаром меняется, то некоторые точки объекта могут больше не появляться. Для этого стоит корректировать имеющуюся информацию об объекте. Для этого каждой точке *p* описывающий объект присваивается счетчик актуальности *h*, имеющий некоторое начальное значение .

* Если для точки объекта *p* найдена соответствующая в ей точка сцены и значение счетчика *p.h* максимально , то значение не изменяется, иначе уменьшается на 1.
* Если для точки объекта *p* найдено соответствие и счетчик *p.h* не максимален, то увеличиваем значение на 1*.*
* На каждом шаге корректировки проверяются счетчики актуальности всех точек объекта , если значение равно нулю , то точка считается неактуальной и удаляется из множества описывающего объект.

Таким образом учитывается изменение в структуре объекта.

Оценка позиции

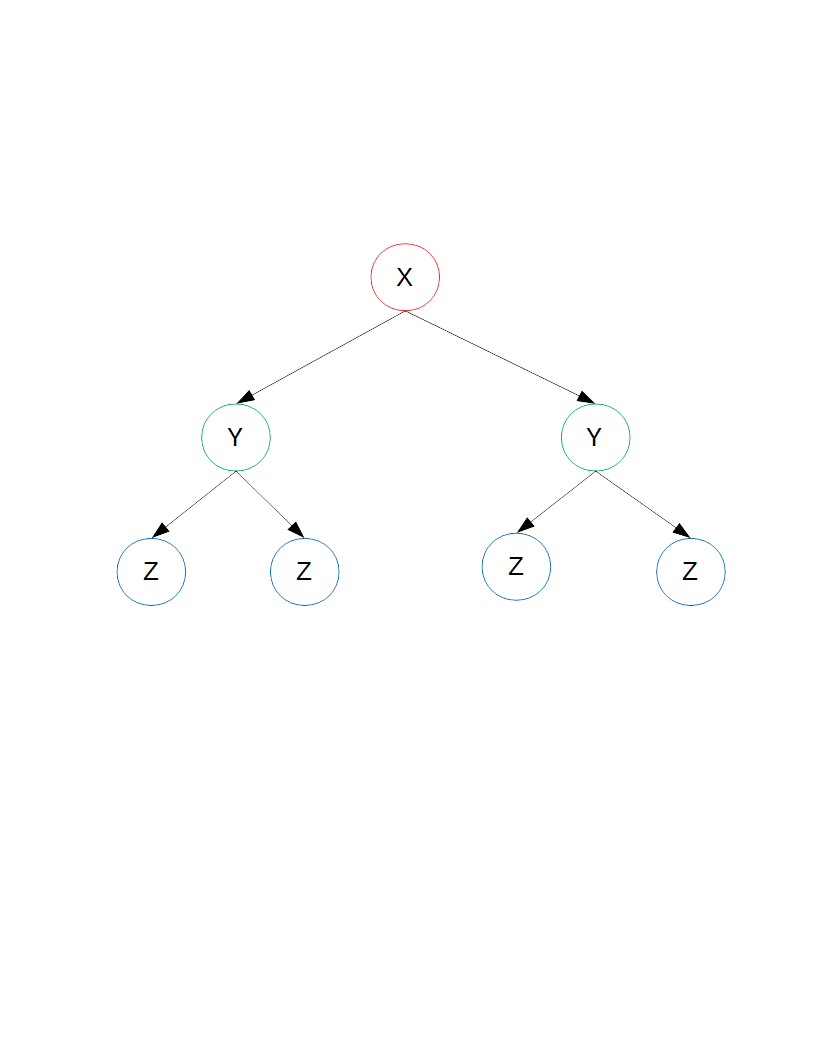
Для перемещения ограничивающего прямоугольника необходимо указать его новую позицию угол поворота. Чтобы получить новую позицию высчитаем вектор смещения *v’ .* Посчитаем смещение всех ключевых точек объекта: *vi = pi“ – pi* где *vi* – вектор смещения i-ой точки , а *pi’* - новое положение точи *pi*. Вычислим вектор средней длинны . Новая позиция прямоугольника *p’= p + v’* где *p*- старая позиция. После данного шага требуется еще одна корректировка.

Так как объект может изменить свое направление, то ограничивающий прямоугольник так же необходимо повернуть . Чтобы скорректировать угол поворота вычислим угол между разностью позиций *p’-p*  и предыдущим вектором направления *d* , для этого посчитаем скалярное произведение *a = (p’-p,d)* , где вектора предварительно нормированные. Величина a будет характеризовать угол на который повернут бокс в новом кадре . Однако, считать, что объект поменял направление можно только при условии относительной непрерывности кадров. Например, если объект поменяет направление , после чего сильно изменит позицию и восстановит направление, то ограничительная область будет иметь неверное направление. Данная проблема решается вводом константы, оценивающей степень изменения позиции, если позиция сильно изменилась , то можно сохранить направление. Однако на реальных примерах частоты потока данных лидара обеспечивает такого рода непрерывность.

Дерево пространственного разбиения

K-мерное дерево (*KD-Tree*) [14] - геометрическая структура данных, для разбиения к-мерного пространства на подмножества, посредством сечения гиперплоскостями. Такие деревья являются одним из видов бинарных поисковых деревьев (BST). При K = 3 трехмерное дерево, позволяющее разбить 3хмерное пространство, с помощью плоскостей. При K = 2 квадро-деревья позволяющее разбить плоскость с помощью прямых. При K=1, дерево превращается в обыкновенное бинарное дерево, разбивающее одномерное пространство (линию). Такое разбиение используют для сужения диапазона поиска в K-мерном пространстве. С помощью данной структуры решаются задачи поиск ближнего объекта, поиск пересечения , проецирование точек на 3D сетку, трассировка лучей. При этом объекты пространства помещаются в специальные параллелепипеды-*bounding box-ы*(bounding box-ом назовем такой параллелепипед, который описывает исходное множество объектов или сам объект, если мы строим bounding box лишь для него. У точек в качестве bounding box-а берется bounding box с нулевой площадью поверхности и нулевым объемом), стороны которых параллельны осям координат.

В нашем случае K-мерное (K=3) дерево - это двоичное дерево, в котором каждый узел является 3-мерной точкой. Каждый нелистовой узел можно рассматривать как неявную разбивающую гиперплоскость, которая делит пространство на две части, левое меньше ,а правое больше. Точки слева от этой гиперплоскости представлены левым поддеревом этого узла и указывают справа от гиперплоскости, представлены правым поддеревом. Направление гиперплоскости выбирается следующим образом: каждый узел в дереве связан с одной из 3-мерной, причем гиперплоскость перпендикулярна оси этого измерения. Так, например, если для конкретного разделения выбрана ось «x», все точки в поддереве с меньшим значением компоненты «x», чем узел, будут лежать в левом поддереве, и все точки с большим значением компоненты «x» будут в правом поддереве. В таком случае гиперплоскость будет задаваться значением OX точки, а его нормаль будет единичной осью OX. На *Рисунке 9* представлена схема 1 шага разбиения:

Рисунок 9. Схема разбиения одного подпространства.

Описание экспериментальной части

Данная реализация не предполагалась как система реального времени по этой причине оценка времени не так важна. Однако надо учесть , что объем реальных данных очень велико и исчисляется петабайтами, и в глобальном смысле, слежение за объектом может занять много времени .

Приведем результаты времени работы всего алгоритма на файлы различной длинны.

Можно проанализировать время затрачиваемое каждой компонентой, требующей большое количество математических вычислений.

Важной характеристикой при поиске ключевых точек является их количество. Этот пункт может быть полезен как эмпирическая оценка минимального количества точек с помощью которого можно описать объект, к примеру , если в области интереса не находится минимального количества точек , то можно искать точки на всей сцене.

Список литературы

1. *Andreas Geiger, Philip Lenz, Christoph Stiller and Raquel Urtasun:* Vision meets Robotics: The KITTI Dataset
2. *V. Rodehorst, A. Koschan*.: Comparison and evaluation of feature point detectors, 2006.
3. *Cordelia Schmid, Roger Mohr, Christian Bauckhage.*: Evaluation of Interest Point Detectors
4. *Harris, C., Stephens*, M.: A combined corner and edge detection
5. *Ivan Sipiran, Benjamin Bustos* : Harris 3D: a robust extension of the Harris operator for interest point detection on 3D meshes
6. Persistent Point Feature Histograms for 3D Point Clouds
7. Learning Informative Point Classes for the Acquisition of Object Model Maps
8. *A. M. and A. A.:* “On normals and projection operators for surfaces defined by point sets,” in Proceedings of Symposium on Point-Based Graphics
9. *Radu Bogdan Rusu, Zoltan Csaba Marton, Nico Blodow, Michael Beetz*: Learning Informative Point Classes for the Acquisition of Object Model Maps
10. *И.О. Сакович, Ю.С. Белов:* Применение метода Лукаса — Канаде для вычисления оптического потока
11. Vychislenie opticheskogo potoka metodom Lukasa–Kanade [Calculation of optical flow by Lucas-Kanade method]. Multifunctional site. Available at: http://habrahabr.ru/post/169055
12. *Li L., Huang W., Gu I. Y. H., Tian Q.:* Foreground object detection in changing background based on color co-occurrence statistics. Applications of Computer Vision
13. *Ayush Dewan Tim Caselitz Gian Diego Tipaldi Wolfram Burgard****:*** Rigid Scene Flow for 3D LiDAR Scans
14. Bentley J. L. :Multidimensional binary search trees used for associative searching
15. Castellani, U., Cristani, M., Fantoni, S., Murino, V. : Sparse points matching by combining 3D mesh saliency with statistical descriptors