

Правительство Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего
профессионального образования
«Санкт-Петербургский государственный университет»

Кафедра Системного Программирования

Пышнова Александра Витальевна

Параллельное отслеживание и составление карты для небольших рабочих
пространств

Дипломная работа

Допущена к защите.

Зав. кафедрой:

д.ф.-м.н., проф. Терехов А.Н.

Научный руководитель:

к.ф.-м.н., доц. Вахитов А.Т.

Рецензент:

асп. каф. СП Кузнецов К.О.

Санкт-Петербург

2015

SAINT-PETERSBURG STATE UNIVERSITY

Software Engineering Department

Pyshnova Alexandra

Parallel tracking and mapping for small workspaces

Graduation Thesis

Admitted for defence.

Head of the chair:

Professor A.N. Terekhov

Scientific supervisor:

Associate Professor A.T. Vakhitov

Reviewer:

Assistant K.O. Kuznetsov

Saint-Petersburg

2015

Оглавление

1.	Введение	4
2.	Постановка задачи	10
3.	Модель данных	11
4.	Основные алгоритмы	14
5.	Эксперименты	18
5.1.	Синтетические данные	18
5.1.1.	Модель	18
5.1.2.	Восстановление карты	18
5.1.3.	Восстановление траектории	20
5.2.	Реальные данные	21
5.2.1.	Описание эксперимента	21
5.2.2.	Восстановление карты	21
5.2.3.	Восстановление траектории	24
6.	Результаты	25
7.	Список литературы	26

1. Введение

Создание полностью автономных мобильных роботов или транспортных средств - одна из наиболее изучаемых областей в робототехническом сообществе. Такие технологии применяются в различных сферах, например, для управления домашними роботами, в медицинской области, для автономного управления автомобилями или подводными средствами.

Устройство должно уметь ориентироваться в полностью неизвестной обстановке, получая информацию о внешнем мире с различных датчиков. Для этого необходимо построить карту местности. Чтобы решить эту задачу, устройству необходимо знать свое точное местоположение. А для определения положения в текущий момент времени необходимо иметь точно построенную карту. Данная задача называется - задача одновременной локализации и составления карты (Simultaneous localization and mapping) [2].

Исходя из сферы применения технологии, устройство может получать данные с различных типов датчиков. Для решения задачи SLAM могут использоваться тактильные, оптические, звуковые, gps сенсоры [5], [14]. Задача построения карты, для которой используются данные только с одной камеры, называется монокулярной (monocular SLAM). Такой класс является самым распространенным, поскольку камера имеет ряд преимуществ по сравнению с другими видами датчиков:

- легкий вес;
- небольшая стоимость;
- массовое производство;
- простота в использовании.

Сложность работы с камерой заключается в том, что она является проективным сенсором, и сложно получить информацию как о положении самого устройства, так и о положении объектов, находящихся перед камерой.

Разработаны различные алгоритмы, решающие задачу monocular SLAM. [3], [4], [6] - [12].

В статье [4] для инициализации новых особенностей и реконструкции 3D координат используется техника параметризации точки величиной обратной глубине (Delayed Inverse-Depth). DI-D применяется для включения новых особенных точек в карту. Суть метода заключается в использовании стохастической триангуляции для определения начальной глубины особенной точки через задержку, что позволяет удалить “слабые” особенности и добавить в карту только “хорошие”. Авторы описали результаты работы алгоритма на двух видах данных: изображение небольшого рабочего пространства (компьютерный стол), видео последовательность, полученная с камеры робота, отснятая на улице.

В статье [3] так же, как и в [4], используется техника Delayed Inverse-Depth (DI-D) для включения новых особенных точек в карту. Данная техника минимизирует количество ошибок при реконструкции, но не исключает их полностью, поэтому разработчики также применяют технику Highest Order Hypothesis Compatibility Test (НОНСТ) для проверки соответствия данных и устранения ошибок.

В отличие от статей [3], [4], в которых используются фильтры, в статье [6] описан подход нелинейной оптимизации (в SLAM нелинейная оптимизация формулируется как поиск положения камеры и координат точек в карте с помощью минимизации ошибки перепроекции меток просматриваемых в кадре) и используется понятие ключевых кадров (поскольку изменение между

двумя последовательными кадрами небольшие, нет необходимости производить оптимизацию и добавлять новые точки после обработки каждого из них), что помогает уменьшить число дорогих операций и позволяет алгоритму работать в режиме реального времени. В данной статье ключевые кадры выбираются так: если число соответствующих точек с предыдущим ключевым кадром стало меньше 400. Для оценки качества работы алгоритма был проведен следующий эксперимент: камеру и gps сенсор прикрепили на транспортное средство, которое двигалось по центру города и пригорода. В режиме реального времени восстанавливалась траектория движения, а для определения точности построения, сравнивалась в данными, полученными с gps.

Исследование в статье [7] сосредоточено на определении положения. Представленный алгоритм эффективно работает для роботов, перемещающихся в небольшом замкнутом пространстве, где локализация камеры может повторяться, а не для случаев, когда происходит движение по неизвестной территории. Ключевая концепция данного подхода - составление вероятностной карты на основе особенностей. При запуске карта инициализируется, а ее изменения происходят динамически: в нее могут добавляться новые особенности с помощью расширенного фильтра Калмана (ЕФК) и, по необходимости, удаляться. В статье описан эксперимент с гуманоидным роботом, движущимся в небольшом замкнутом пространстве по кругу. Представленный алгоритм применяется для построения карты. Поскольку программа работает в режиме реального времени, карты строится не плотной и состоит из редких точечных ориентиров.

В статье [9], так же как и в [7], используется вероятностный подход для реконструкции карты. В описанном методе для определения глубины карты используется объединение оценки Байеса и выпуклой оптимизации.

Реализованный метод был протестирован на видео последовательностях с изображением рабочего стола и мини-города.

В работе [8] ключевой идеей является представление карты в виде двухуровневого вложенного графа, в котором как уровни рассматриваются локальное и глобальное положения (оценка положения робота с помощью глобальных особенностей дает более точную оценку положения робота). Шаги алгоритма: определение локальных особенностей с помощью техники SIFT (scale-invariant feature transform) на полученном кадре, определение схожести текущего кадра со всеми узлами глобальной карты (схожесть определяется как количество соответствующих локальных особенностей), если схожесть выше определенного порога, то кадр добавляется к данному глобальному узлу, если таких узлов нет - текущий кадр добавляется как новый узел и привязывается к предыдущему кадру. Похожим образом рассчитывается положение кадра на уровне локальной карты, только схожесть между узлами определяется за счет сравнения 2D преобразований Хаара. Описанный алгоритм был протестирован в помещении Лаборатории компьютерного зрения и активного восприятия в Королевском технологическом институте (Стокгольм, Швеция). Управляемый вручную робот непрерывно отснял несколько помещений офиса. Каждый кадр полученной видео последовательности был помечен названием помещения, в котором он был отснят. На выходе из алгоритма была получена топологическая карта лаборатории.

В работе [10] используется новый семантический подход для решения задачи SLAM. Данная техника отличается тем, что помимо особенных точек в карту добавляются известные объекты из базы данных. Алгоритм делится на два потока, один решает задачу monocular SLAM с помощью расширенного фильтра Калмана, а второй поток распознает объекты с помощью Speeded Up

Robust Feature (SURF) дескрипторов и добавляет их в построенную карту. В статье представлены результаты эксперимента на видео последовательностях с изображением рабочего стола (распознавались такие объекты, как открытки, игрушка, пакет сока) и больничной палаты (распознавались шкаф, кровать, пакет сока).

Высокий результат показал алгоритм Параллельного отслеживания и составления карты (PTAM) [13]. Данный метод применялся не для исследования роботов, а для отслеживания движения ручной камеры в небольших пространствах дополненной реальности. Суть алгоритма заключается в разделении отслеживания особенных точек и составления карты и их выполнения параллельно в два потока. Одной из особенностей алгоритма PTAM является плотная инициализация карты за счет особенностей низкого качества, которая выполняется с помощью стерео техники (5-точечный алгоритм). Так же важным в данном подходе является понятие ключевых кадров. В алгоритме PTAM кадр считается ключевым, если выполняются следующие условия: с добавления предыдущего ключевого кадра должно быть обработано не менее 20 кадров и должно существовать минимальное расстояние между камерой и ближайшей точкой в карте. На двух последовательных ключевых кадрах ищутся соответствия с помощью эпиполярного поиска, далее по полученным проекциям точки триангулируются и добавляются в карту. Поскольку отслеживание и построение карты выполняются параллельно, для уточнения карты и параметров камеры используется дорогостоящая техника оптимизации bundle adjustment. В PTAM применяются global bundle adjustment (положение уточняется для всех ключевых кадров и всех точек карты) и local bundle adjustment (запускается для некоторого подмножества ключевых кадров и некоторого подмножества точек карты).

Решение задачи SLAM является актуальной темой. Было разработано множество алгоритмов для разных сфер применений, но нет отдельного модуля геометрической реконструкции для однокамерной системы. Поэтому в качестве задачи было решено взять создание такого модуля с помощью подхода нелинейной оптимизации, чтобы его можно было применять на практике для решения. За основу разрабатываемого метода был выбран алгоритм PTAM, поскольку у него есть ряд преимуществ по сравнению с другими алгоритмами: высокая точность за счет применяемых техник оптимизации, управление скоростью работы алгоритма за счет изменения точности в Bundle Adjustment. Так же, поскольку PTAM не является рекурсивным (оценка положения камеры в каждый момент времени происходит по данным с текущего кадра и восстановленной карты и не зависит от оценки положения в предыдущий момент времени), ошибка не будет накапливаться от кадра к кадру.

2. Постановка задачи

Цель диплома --- создание библиотеки геометрической реконструкции для однокамерной системы. Для достижения этой цели необходимо решить следующие задачи.

1. Создать синтетическую модель для тестирования алгоритма.
2. Реализовать геометрический модуль реконструкции на основе алгоритма РГАМ
3. Провести испытания на синтетических и реальных данных.

3. Модель камеры.

Камеру можно описать с помощью проективной модели [1]. Проективная камера определяется центром камеры, проективной плоскостью и главной осью (определяет направление камеры).

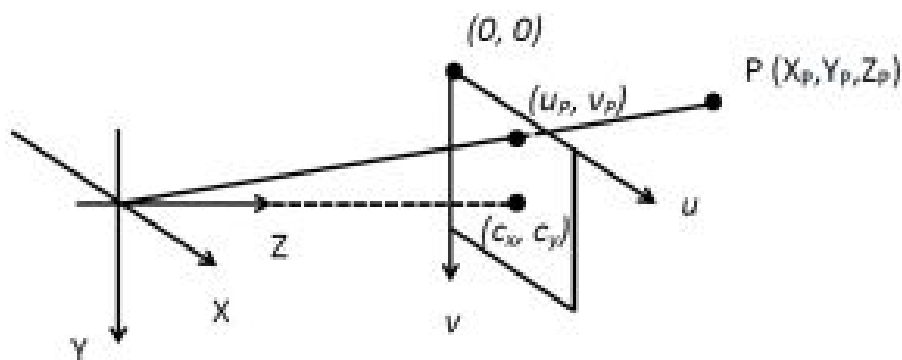


Рис. 1 Проективная модель.

На рис. 1 - XYZ - система мировых координат, uv - система координат камеры, (c_x, c_y) - центр камеры, f - фокусное расстояние, точка с координатами (X_p, Y_p, Z_p) и ее проекция на область камеры (u_p, v_p) . Проекции точек ищутся по формулам :

$$u = f \frac{X}{Z} + c_x; \quad v = f \frac{Y}{Z} + c_y$$

Описание в терминах линейной алгебры выглядит следующим образом: пусть X - координаты точки в пространстве, x - однородные координаты в плоскости изображения, тогда $x = PX$, где P - матрица, имеющая вид $P = K[R|t]$, K - треугольная матрица калибровки камеры размера 3×3 , R - ортогональная

матрица размера 3×3 , определяющая поворот системы координат камеры относительно мировых координат, t - вектор смещения размера 3.

Матрица K имеет вид:

$$K = \begin{pmatrix} f_x & 0 & x_0 \\ 0 & f_y & y_0 \\ 0 & 0 & 1 \end{pmatrix},$$

где f_x, f_y - фокусное расстояние по x и y , а x_0, y_0 - координаты центра камеры.

Первая подзадача диплома - создание модели для тестирования алгоритма. Программа работает следующим образом:

1. На вход подается траектория движения, каждая точка которой определяется вектором сдвига и параметрами матрицы поворота. Для матрицы калибровки значения фокусного расстояния и координат центра камеры взяты значения, приближенные к реальным условиям.

$$K = \begin{pmatrix} 320 & 0 & 320 \\ 0 & 320 & 240 \\ 0 & 0 & 1 \end{pmatrix}$$

2. Генерируется случайный набор начальных точек (3D координаты).
3. Для точки траектории по вектору сдвига, матрице поворота и матрице калибровки считаются проекции. Координаты точек, не попавших в плоскость камеры, приравниваются 0.
4. По необходимости добавляется шум (нормальное распределение)

С такой моделью решение задачи сводится к формуле:

$$\sum_c \sum_p \| P_c x' - (u_p \ v_p \ 1) \|^2 \rightarrow \min ,$$

где $P_c = K(R_c | t_c)$, K – матрица камеры, R_c , t_c – найденные матрица поворота и вектор сдвига текущего кадра c , x' – восстановленные 3D координаты точки p , (u_p, v_p) – данные координаты проекции точки p на плоскость камеры.

4. Основные алгоритмы.

Модуль геометрической реконструкции был реализован на языке C++ с использованием библиотеки OPENCV [17]. На вход программе подаются треки с проекциями отслеживаемых точек. На выходе - реконструированные траектория и карта и информация об ошибках реконструкции.

Модуль представлен в виде процессора, в котором выполняется цикл покадрового чтения. Для каждого кадра выполняется запись в конкретные структуры (точка, кадр, трек) и производится обработка. Процессор может пребывать в двух состояниях:

1. инициализация карты;
2. обработка кадра.

Состояние “Инициализация карты”.

Находясь в первом состоянии, процессор получает последовательно два первых кадра и с помощью эпиполярной геометрии [15] определяет положение камеры на втором кадре относительно первого и реконструирует карту.

Алгоритм:

1. Отбрасывание плохих точек. Выполняется с помощью метода RANSAC
2. Поиск фундаментальной матрицы. Фундаментальная матрица - отношение между двумя изображениями одной и той же сцены, которое ограничивает возможность пребывания проекций точек в обеих плоскостях. То есть если известны проекции на плоскости изображений, то можно проверить будут ли они действительно проекциями одной точки. В терминах линейной алгебры фундаментальная матрица F - матрица размера 3×3 , тогда проекции x_l и x_r - стереопара тогда и только

тогда, когда выполняется соотношение $x_r F x_l = 0$ (x_l и x_r в однородных координатах).

3. Уточнение фундаментальной матрицы.
4. Поиск существенной матрицы. Существенная матрица - матрица E размера 3×3 , аналогично фундаментальной матрице связывает соответствующие точки на стерео изображениях. x_l и x_r - стереопара тогда и только тогда, когда выполняется соотношение $x_r E x_l = 0$ (x_l и x_r в нормализованных координатах). Существенная матрица вычисляется по формуле : $E = K'^T F K$, где K' и K – матрицы камеры на втором и первом кадре соответственно.
5. Поиск матрицы поворота и вектора сдвига по существенной матрице [18].
 - a. SVD разложение существенной матрицы в виде:

$$E = U \text{diag}(1 \ 1 \ 0) V^T .$$

- b. Составление возможных решений (рис. 2) по формулам:

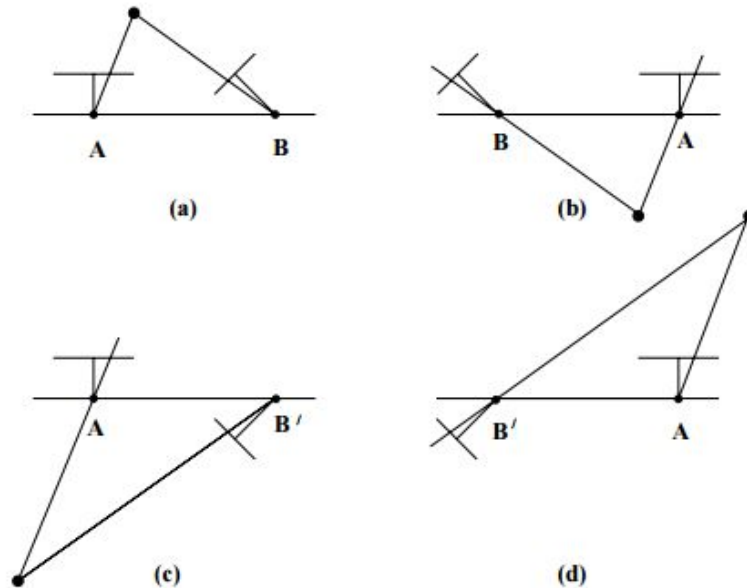


Рис. 2 Возможные расположения камер.

$$P'_1 = [U W V^T \mid + u_3]$$

$$\begin{aligned}
 P'_2 &= [UWV^T \mid -u_3] \\
 P'_3 &= [UW^T V^T \mid +u_3] \\
 P'_4 &= [UW^T V^T \mid -u_3],
 \end{aligned}$$

$$\text{где } u_3 = U(0 \ 0 \ 1)^T \quad \text{и} \quad W = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

- с. Выбор правильного решения из четырех. Для этого необходимо восстановить одну точку по ее проекциям на плоскости изображений и убедиться, что она будет лежать перед обеими камерами. Вариант, для которого это будет выполняться, только один. Таким способом можно найти матрицу поворота R и нормированный вектор смещения t .
6. Добавление найденных вектора сдвига и матрицы поворота в траекторию.
 7. По найденным матрице поворота, вектора сдвига и проекциям точек на две плоскости - восстановление карты с помощью триангуляции. Триангуляция - процесс восстановления 3D координат точек по известным проекциям. Был выбран метод линейной триангуляции [16]. Составляется система решается с помощью метода наименьших квадратов.
 8. Переход в состояние “обработка кадра”.

Состояние “обработка кадра”.

Пребывая в этом состоянии, процессор читает кадр по одному и для каждого из них выполняет следующие действия:

1. Поиск матрицы поворота и вектора сдвига с помощью метода Perspective-n-point по проекциям точек на плоскость текущего кадра и реконструированной карте.
2. Добавление параметров текущего кадра в траекторию.

В алгоритме два важных момента - техника sparse bundle adjustment и понятие ключевых кадров. SBA - уточнение параметров камер на всех кадрах и уточнение карты. Данная операция требует много вычислений и выполнять ее после определения параметров каждого кадра невыгодно. Поэтому появляются ключевые кадры, после обработки которых и запускается SBA.

5. Эксперименты.

Эксперименты проводились на двух видах данных.

1. Синтетическая модель.
2. Реальные данные.

Для оценки качества выполнения реконструкции применялись следующие метрики.

1. Ошибка перепроектирования. Считается по формуле $\| Px' - (u_p \ v_p \ 1) \|$, где x' - реконструированные точки, $P = K[R|t]$, R , t - найденные матрица поворота и вектор сдвига, (u_p, v_p) - начальные координаты проекций точек.
2. Ошибка в определении параметров.

5.1. Синтетические данные.

5.1.1. Модель.

Была описана траектория и созданы проекции точек с помощью реализованной модели. Для начала алгоритм был протестирован на данных без добавления шума. Далее в качестве шума было взято нормальное распределение с дисперсиями от 0.5 до 3 пикселей.

5.1.2. Восстановление карты.

Было сгенерировано 200 точек (рис. 3), значения x и y варьировались от 300 до 800, z - от 600 до 1500. Далее были удалены точки, проекции которых не попадали на первый кадр. По описанной траектории были созданы проекции оставшихся точек и к этим проекциям добавлен шум (нормальное

распределение с дисперсией 1 пиксель) (рис. 4а и рис. 4б - проекции точек на первый и второй кадр соответственно). На рис. 5. изображена восстановленная карта.

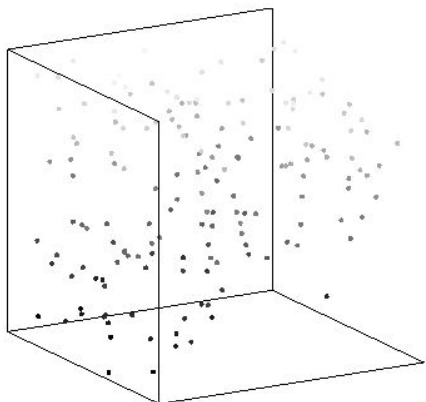
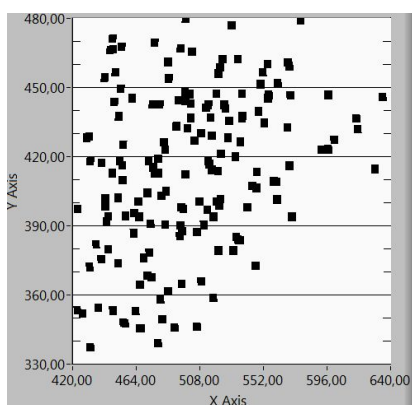
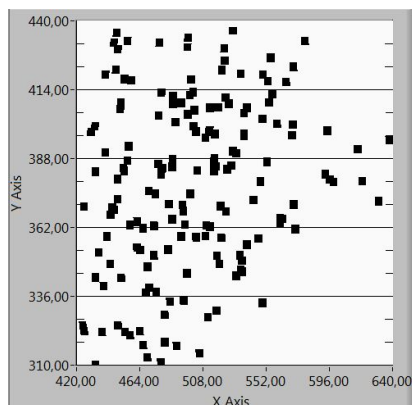


Рис. 3 Начальные точки.



а



б

Рис. 4. Проекция сгенерированных точек.

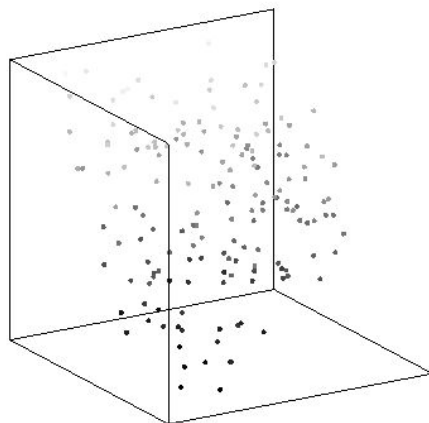


Рис. 5. Восстановленная карта.

На рис. 6 изображена гистограмма с ошибкой перепроектирования на второй кадр.

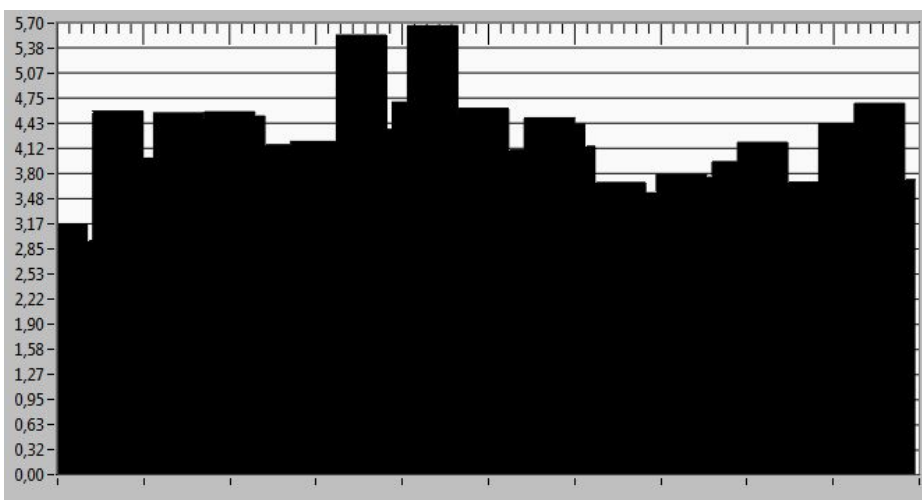


Рис. 6. Ошибка перепроектирования на второй кадр.

5.1.3. Восстановление траектории.

На рис. 7 и рис. 8 изображены - начальная траектория и восстановленная по проекциям без шума.

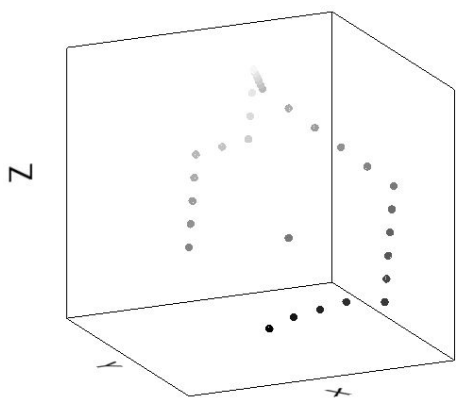


Рис. 7. Начальная траектория.

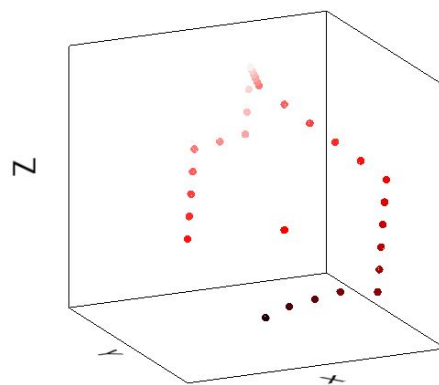


Рис. 8. Восстановленная траектория. Шум = 0

5.2. Реальные данные

5.2.1. Описание эксперимента

Для проведения экспериментов с реальными данными было отснято видео с камеры робота, который двигался по комнате (кадр из видео представлен на рис. 9). Далее были созданы треки с проекциями отслеживаемых точек. По этим трекам восстанавливалась карта и траектория.



Рис. 9. Кадр из видео с камеры робота.

5.2.2. Восстановление карты

На рис. 10 изображена восстановленная карта, а на рис. 11 и рис 12 изображены гистограммы с ошибками перепроекции на 2 и на 50 кадр соответственно. По данным графикам видно, что ошибка перепроекции велика, а для нескольких особенных точек достигает 200. Это связано с тем, что смещение между двумя первыми кадрами очень мало и по проекциям точек на данные кадры сложно получить информацию об их глубине, что привело к неправильному восстановлению 3D координат этих точек на этапе триангуляции. Соответственно перепроекции неточно восстановленных точек будут значительно отличаться от изначальных.

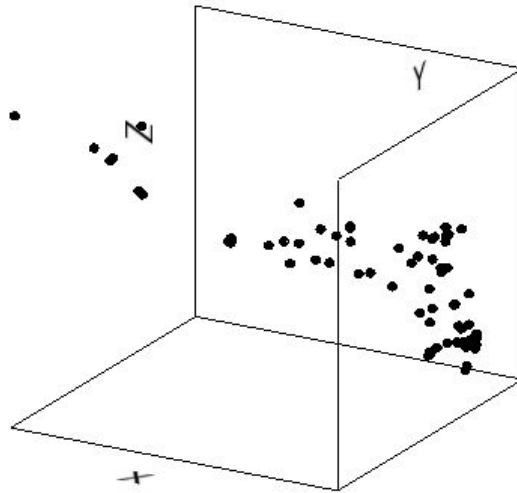


Рис. 10. Восстановленная карта.

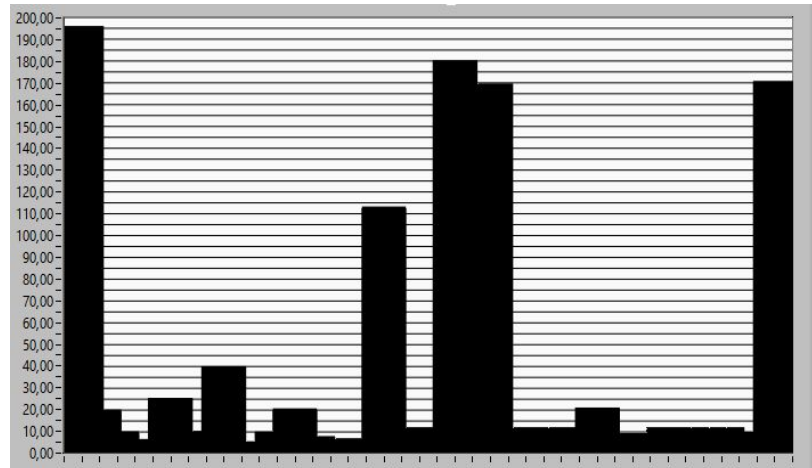


Рис. 11. Ошибка перепроектирования на 2 кадр

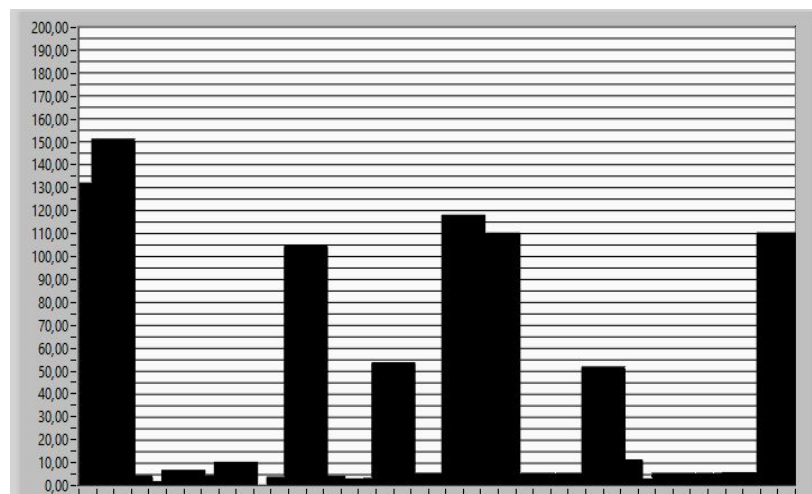


Рис. 12. Ошибка перепроектирования на 50 кадр

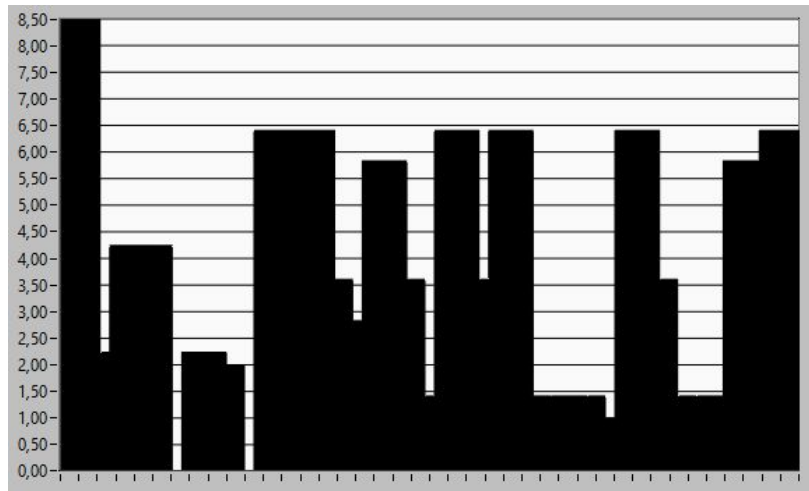
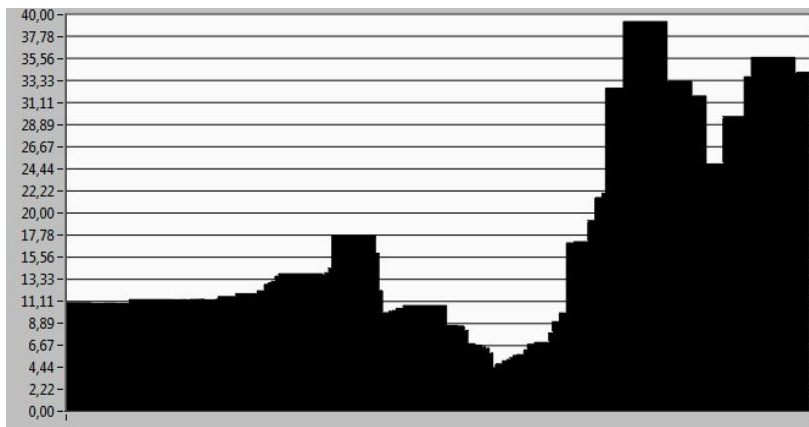


Рис. 13. Ошибка перепроектирования на 2 кадр

Далее был проведен эксперимент, карта восстанавливалась не по первым двум кадрам видео последовательности. На рис. 13 изображена гистограмма с ошибкой перепроектирования.

На рис. 14 изображена гистограмма со средней ошибкой перепроектирования по всем кадрам для случая, когда карта инициализировалась не по первым двум кадрам.



5.2.3. Восстановление траектории.

На рис. 15 изображена восстановленная траектория движения для случая, когда карта инициализируется не по первым двум кадрам.

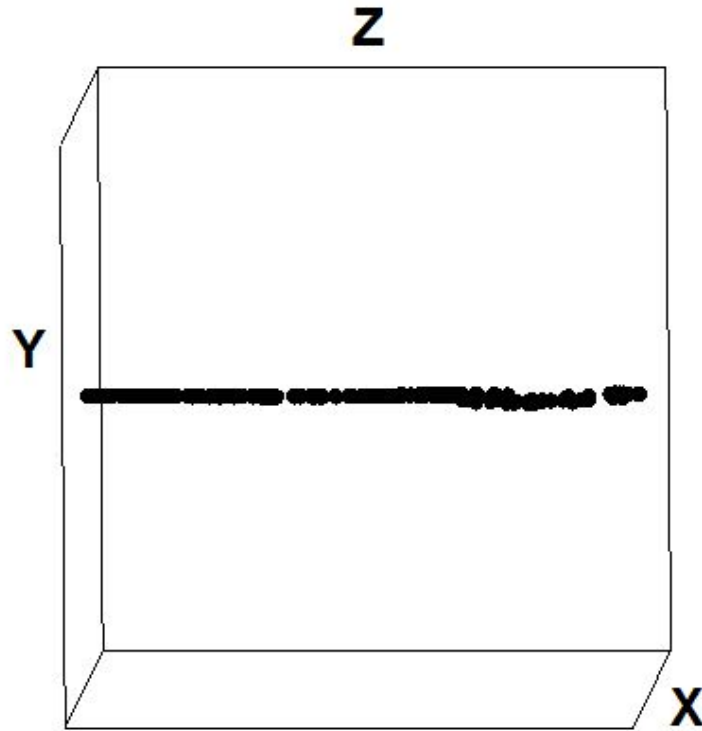


Рис. 15. Восстановленная траектория.

6. Результаты.

В ходе работы над дипломом были выполнены следующие задачи:

1. Создана синтетическая модель данных для тестирования алгоритма.
2. Реализован геометрический модуль реконструкции на основе алгоритма РТАМ
3. Проведены испытания на синтетических и реальных данных.

В ходе работы над дипломом выявились некоторые сложности с инициализацией карты. Если реконструкция выполняется по кадрам, между которыми сдвиг небольшой, воспроизведение траектории происходит со значительными ошибками. Поэтому в дальнейшем хотелось бы реализовать инициализацию карты по кадрам, соответствующим описанному критерию. Так же сделать выбор ключевых кадров не по времени, а по отношению количества соответствий в карте к количеству новых соответствий.

7. Список литературы.

1. Richard Szeliski. Computer Vision: Algorithms and Applications. // Springer, 2010
2. Simultaneous localization and mapping, [URL:https://en.wikipedia.org/wiki/Simultaneous_localization_and_mapping](https://en.wikipedia.org/wiki/Simultaneous_localization_and_mapping), дата последнего обращения: 09.06.2015
3. Edmundo Guerra, Rodrigo Munguia, Yplanda Bolea, Antoni Grau. A Highest Order Hypothesis Compatibility Test for Monocular SLAM. // International Journal of Advanced Robotic Systems, 2013
4. Rodrigo Munguia, Bernardino Castillo-Toledo, Antoni Grau. A Robust Approach for a Filter-Based Monocular Simultaneous Localization and Mapping (SLAM) System. //Sensors no. 7, 2013
5. Stefan Leutenegger, Paul Furgale, Vincent Rabaud, Margarita Chli, Kurt Konolige, Roland Siegwart. Keyframe-Based Visual-Inertial SLAM Using Nonlinear Optimizaion. // in Proceedings of Robotics: Science and Systems (RSS), 2013
6. E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, P. Sayd. Monocular Vision Based SLAM for Mobile Robots. // in Proc. Int. Conf. on Pattern Recognition. Washington, DC, USA: IEEE Computer Society, 2006, pp. 1027–1031
7. Andrew J. Davidson, Ian D. Reid, Nicholas D. Molton, Olivier Stasse. MonoSLAM: Real-Time Single Camera SLAM. //IEEE Trans. Pattern Anal. Mach. Intell. 29(6): 1052-1067 (2007)

8. Sara Elgayar, Mohammed A.-M. Salem, Mohamed I. Roushdy. Nested Graph Representation for Visual SLAM based on Local and Global Feature Processing. // International Journal of Computer Applications. Vol. 78, 2013
9. Matia Pizzoli, Christian Forster, Davide Scaramuzza. REMODE: Probabilistic, Monocular DenseReconstruction in Real Time. // IEEE International Conference on Robotics and Automation (ICRA), 2014.
10. Javier Civera, Dorian Galvez-Lopez, L. Riazuelo, Juan D. Tardos, J. M. M. Montiel. Towards Semantic SLAM using a Monocular Camera. // in Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS), 2011
11. Thomas Lemaire, Cyrille Berger, IL.-Kyun Jung, Simon Lacroix. Vision-Based SLAM: Stereo and Monocular Approaches. // International Journal of Computer Vision, 2007
12. Алгоритмы SLAM, URL: <https://www.openslam.org/>, дата последнего обращения: 09.06.2015
13. G. Klein, D. Murray. Parallel Tracking and Mapping for small AR workspaces. // In IEEE and ACM Intl. Sym. on Mixed and Augmented Reality (ISMAR), p. 225-234, 2009
14. Hordur Johannsson, Michael Kaess, Maurice Fallon, John J. Leonard. Temporally Scalable Visual SLAM using a Reduced Pose Graph. // in Robotics and Automation (ICRA), 2013 IEEE International Conference on, pp. 54-61
15. Gang Xu, Zhengyou Zhang. Epipolar Geometry in Stereo, Motion and Object Recognition - A Unified Approach. // Computational Imaging and Vision 6, Springer, ISBN, 1996
16. Richard I. Hartley, Peter Sturm. Triangulation. // In Computer Vision and Image Understanding. Vol. 68, No. 2, November, pp. 146-157, 1997

17. OPENCV, URL: <http://opencv.org/about.html>, дата последнего обращения:
09.05.2015
18. R. Hartley, A. Zisserman. Multiple View Geometry in Computer Vision 2nd
Ed. // Cambridge University Press, 2004