Санкт-Петербургский Государственный Университет
Математико-механический факультет

Математическое обеспечение и администрирование
информационных систем
Кафедра системного программирования

Люлина Елена Сергеевна

Комбинирование ключевых точек и прямого подхода для стерео-SLAM

Курсовая работа

Научный руководитель:
Пименов А. А.

Консультант:
Корчемкин Д. А.

Санкт-Петербург
2019

Оглавление

Введение

Компьютерное зрение – достаточно молодая и стремительно развивающаяся область, в которой рассматриваются задачи получения информации из изображений. Такие задачи находят свое применение во многих передовых областях, начиная с дополненной реальности и заканчивая автономными машинами и беспилотными летательными аппаратами. Для последних, например, стоит задача навигации – при помощи установленной на них системы камер из потока изображений требуется получить информацию о местонахождении и карте окружающей обстановки.

Одним из методов решения этой задачи является одновременная локализация и построение карты (Simultaneous Localization And Mapping – SLAM), при котором нахождение траектории и карты окружающей среды происходит одновременно.

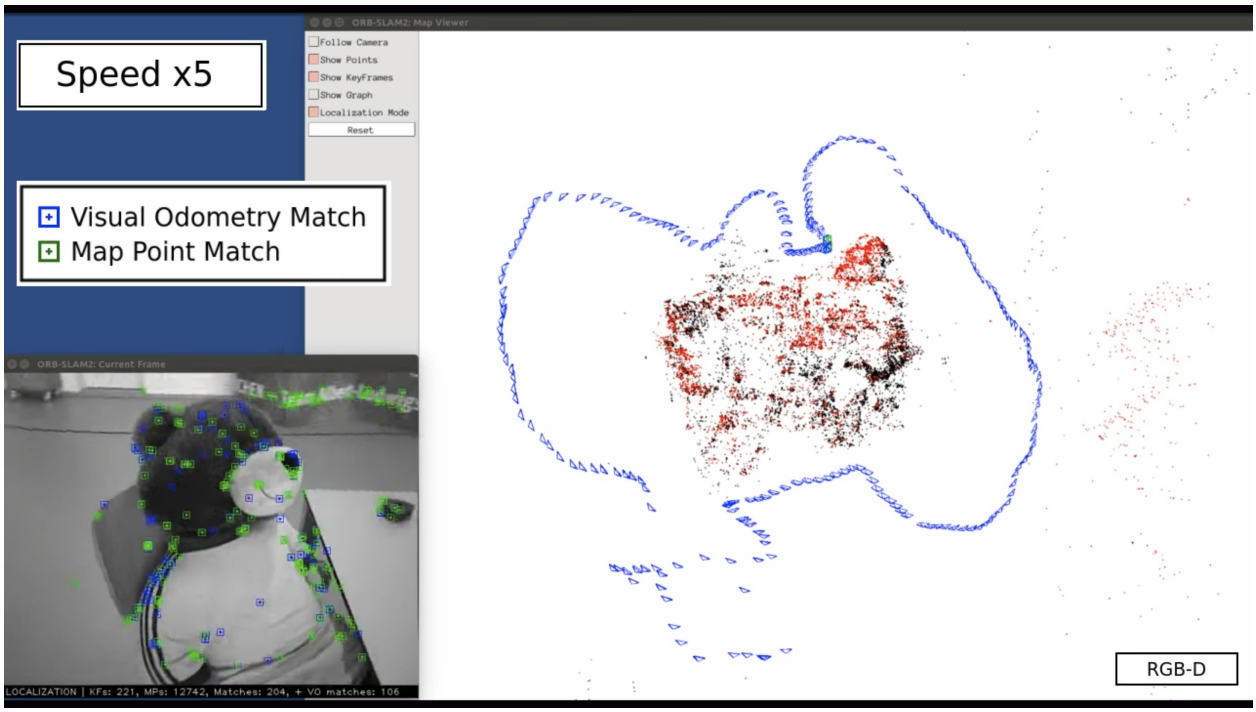


Рис. 1: Пример работы алгоритма ORB-SLAM2 [4], основанном на ключевых точках.

Входными данными служат последовательно поступающие изображения, полученные при помощи системы камер с пересекающимися областями видимости. Результатом обработки каждого изображения яв-

ляются *поза* системы камер и информация о точках окружающей обстановки, которая может быть представлена *картой глубины*.

Карта глубины хранит в себе информацию о глубинах пикселей на изображении, то есть о расстоянии между камерой и предметом, отображаемым пикселем. С ее помощью строится карта окружающей среды.

Позой объекта называется комбинация его положения и ориентации. Полученные оценки позы системы камер соответствуют траектории движения.

Глобально существует два подхода, применяющихся в алгоритмах SLAM: подход, основанный на ключевых точках (feature-based image alignment), и подход, основанный на прямом сравнении интенсивностей пикселей (direct image alignment). Их упрощенное описание будет приведено ниже. Каждый из этих подходов имеет свои плюсы и минусы, причем недостатки одного подхода покрываются достоинствами другого.

1. Постановка задачи

Уравновесив минусы одного подхода плюсами другого, можно достичь лучших результатов. Целью работы является комбинирование подхода, используещего ключевые точки, и подхода, основанного на прямом сравнении интенсивностей. В качестве алгоритма, основанного на использовании ключевых точек, был выбран SOFT2 [2], для прямого подхода – Stereo-DSO [7], так как в настоящее время они являются одними из лучших в своей области [9].

Задача реализации алгоритма, основанного на комбинации этих двух подходов, достаточно большая и не укладывается по времени в рамки курсовой работы, поэтому было решено сосредоточиться на задачах, которые в дальнейшем будут использованы уже для реализации всего алгоритма, а именно:

1. Ознакомиться с предметной областью, изучить основные подходы и особенности выбранных алгоритмов.
2. Реализовать подзадачи, связаные с ключевыми точками:
   - Детекция ключевых точек на кадрах.
   - Поиск соответствующих ключевых точек на разных кадрах.
3. Оптимизировать время работы реализованных алгоритмов.

2. Обзор

2.1. Упрощённые алгоритмы существующих подходов

1. Подход, при котором на изображении выбираются некие ключевые точки и отслеживаются на протяжении нескольких последующих изображений:

   (a) Получить новое стерео-изображение;

   (b) Выделить ключевые точки. Например, ими могут служить геометрические особенности: углы, пятна;

   (c) С помощью ключевых точек определить местоположение камеры и дополнить карту.

   Пример работы алгоритма с таким подходом представлен на рис. 1, где изображена траектория системы камер и облако точек с известными глубинами. В нижнем левом углу – изображение с детектированными ключевыми точками.

2. Подход, основанный на сравнении интенсивности пикселей соседних изображений:

   (a) Получить новое стерео-изображение;

   (b) Оценить расположение нового изображения, используя оценки глубины значительного числа точек предыдущего ключевого кадра; оптимизация позы происходит по совпадению интенсивностей точек;

   (c) Обновить оценки глубины в старом изображении, или создать новый ключевой кадр из текущего.

2.2. Достоинства и недостатки существующих подходов

Подход, основанный на ключевых точках

Достоинства:

- Не требует инициализации, так как задачи оценки относительной позы двух камер, абсолютной позы камеры относительно трёхмерных точек имеют решения в явном виде;
- Имеет лучшую точность (согласно тестированиям на датасетах KITTI [9]) оценки перемещения и изменения ориентации;
- Определяет замыкание циклов, что позволяет снизить накапливание ошибки, происходящее во время движения.

Недостатки:

- Требуются достаточно текстурированные поверхности и отсуствие размытия вследствие движения;
- Не подходит для задач, требующих проверки пересечения траектории движения с объектами реального мира, так как строится неплотная карта окружающей обстановки.

Подход, основанный на сравнении интенсивностей

Достоинства:

- Оценка глубины известна для значительного количества точек, строится плотная карта окружающей обстановки;
- Работает при однородных текстурах объектов; допускается некоторое размытие вследствие движения.

Недостатки:

- Оценки позы очередного кадра выполняются нелинейной оптимизацией и являются локально-оптимальными (в том числе, зависят от инициализации);
- Замыкание циклов “напрямую” требует вычислительных затрат.

2.3. SOFT2

Этот алгоритм описан в статье [2].

2.3.1. Ключевые точки

Ключевые точки делятся на два типа: угловые и пятна. Чтобы найти на изображении точки каждого типа, оно обрабатывается двумя свертками – детекторами каждого из типов (рис. 2). Далее осуществляется поиск локальных экстремумов в результатах свёртки (с помощью [5]). Таким образом, все ключевые точки разделяются на 4 класса: локальные минимумы углов, пятен и локальные максимумы углов, пятен. Дескриптором служат значения градиента интенсивности в нескольких окружающих точках (рис. 2).

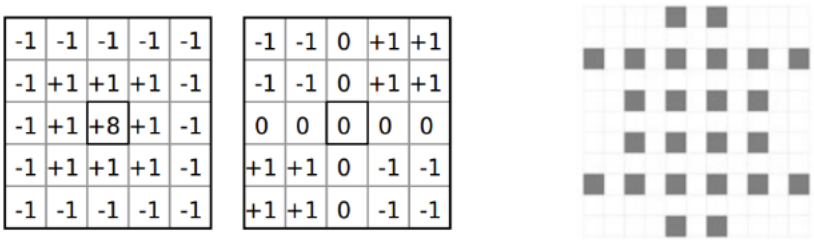


Рис. 2: Ядра сверток для поиска пятен и углов; пиксели, значения в которых используются в качестве дескриптора для центральной точки.

Чтобы найти соответствия детектированных ключевых точек между двумя стерео-кадрами, а так же с точками на предыдущих кадрах, применяется метрика ошибок, основанная на сумме абсолютных разностей. Для увеличения точности, соответствия ищутся по кругу: новый левый кадр – предыдущий левый кадр – предыдущий правый кадр – новый правый кадр – новый левый кадр. Для увеличения скорости, соответствия между правым и левым кадром ищутся только на эпиполярной прямой (с некой ошибкой в несколько пикселей). Если круг замкнулся, то оставшиеся ошибочные точки удаляются после применения нормализованной кросс-корелляции между окрестностями пикселей на сравниваемых изображениях.

Среди оставшихся точек необходимо выбрать лишь небольшое количество равномерно распределенных точек. Это достигается путем деления изображения на области, в каждой из которых выбирается определенное количество точек, основываясь на их возрасте и значению после применения масок-детекторов.

2.3.2. Оптимизация позы

На этом шаге требуется вычислить поворот и смещение камеры относительно предыдущего кадра.

Поворот находится при помощи метода пяти точек [6] для монокамер, в данном случае для него используется только левая камера. Из всего набора ключевых точек выбирается 5, при помощи которых можно найти относительный поворот камеры как один из корней уравнения десятой степени. Для правильного выбора этих пяти точек применяется алгоритм RANSAC (Random Sample Consensus - консенсус случайной выборки), в котором рассматриваются всевозможные сочетания пяти точек. Для каждого полученного поворота определяется количество остальных ключевых точек, ему соответствующих. Поворот с наибольшим количеством признается самым точным и затем итеративно улучшается. Для сглаживания, RANSAC применяется еще несколько раз между предыдущими кадрами.

Смещение находится при помощи стерео-метода одной точки, для которого используется полученная матрица поворота. Для каждой точки смещение находится последовательным уменьшением функции стоимости, а затем выбирается лучшее, как соответствующее максимальному количеству остальных точек. Последним шагом, чтобы уменьшить количество ошибочных ключевых точек и получить итоговый набор ключевых точек, которые будут использоваться в общей оптимизации, применяется алгоритм Гаусса-Ньютона.

2.3.3. Ключевые кадры

Каждый кадр рассматривается как потенциально ключевой и не признается им, если расстояние от него до предыдущего ключевого меньше некоторого предопределённого порога. Если оно больше, то проверяется зацикливание траектории. Если оно не было обнаружено, то кадр признается ключевым, иначе – временно-ключевым и удаяется после оптимизации графа. Для временно-ключевых кадров также ищутся соответствия с кадрами, в которых трактория зациклилась.

2.4. Stereo-DSO

Этот алгоритм описан в статье [7].

2.4.1. Формализация прямого подхода

Для сопоставления двух кадров сравниваются разности их интенсивностей. Чем они меньше, тем точнее сопоставлены кадры, поэтому основной идеей таких подходов служит уменьшение функции энергии, представленной следующей формулой:

$$E_{ij} = \sum_{\mathbf{p} \in \mathcal{P}_i} \omega_p \| I_j[\mathbf{p'}] - I_i[\mathbf{p}] \|_\gamma \tag{1}$$

$\mathbf{p}$ – пиксель на изображении i,
$\mathbf{p'}$ – соответствующий $\mathbf{p}$ пиксель на изображении j,
$\mathcal{P}_i$ – множество пикселей изображения i,
ω_p – взвешивание, уменьшающее вес пикселей с большим градиентом изображения
I_i, I_j – изображения i, j соответственно
$\| \cdot \|_\gamma$ – норма Хьюбера

Несмотря на то, что обычно в прямых подходах эта формула применяется как можно к большему числу пикселей на изображении, это существенно замедляет работу. Поэтому предлагается рассматривать лишь некоторое количество достаточно контрастных точек. В допол-

нение к этому, чтобы противостоять резкому изменению освещения, формула дополняется аффинным изменением яркости.

2.4.2. Оценка положения очередного кадра

При появлении нового кадра на него проецируются точки с предыдущего ключевого кадра и кадр таким образом инициализируется, затем поза нового кадра оптимизируется путем уменьшения функции энергии. Оптимизация происходит при помощи алгоритма Гаусса-Ньютона на пирамиде изображений. При этом требуется, чтобы предыдущий кадр был инициализирован, то есть должны быть известны глубины значительного количества точек.

Для работы всей этой системы необходимо инициализировать первый кадр. Подходы с монокамерой зачастую используют случайные значения для инициализации, но в этом алгоритме используются соответствия между левым и правым стерео-кадрами. Соответствия, как и в прошлом подходе, ищутся вдоль эпиполярной кривой, используя нормализованную кросс-корелляцию.

2.4.3. Ключевые кадры

Если окружающая обстановка или освещение существенно изменились, необходим новый ключевой кадр. Чтобы его создать, из текущего изображения выбирается редкий набор точек, называемых кандидатами. Для их равномерного распределения изображение делится на области и из каждой области выбираются те точки, чьё значение градиента интенсивности превысило некий порог.

Прежде чем активировать кандидатов, их глубина уточняется при помощи предыдущих неключевых кадров, а также при помощи второго стерео-кадра, используя нормализованную кросс-корелляцию.

После этого старые точки на предыдущем ключевом кадре удаляются, заменяясь новыми точками и новым ключевым кадром; так происходит активация кандидатов. После неё каждая точка имеет свой ключевой кадр, в котором она содержится, а так же ряд кадров, из ко-

торых она наблюдаема. Таким образом строится граф, где соединены между собой точки и кадры, из которых они наблюдаются.

Далее происходит общая оптимизация алгоритмом Гаусса-Ньютона – уменьшается функция энергии по всему графу.

2.5. Комбинация SOFT2 и Stereo-DSO

2.5.1. Инициализация Stereo-DSO

Для начальных значений интенсивности на первом кадре в Stereo-DSO предлагается использовать инофрмацию о глубинах ключевых точек из SOFT2. Это достигается путем интерополяции по сетке, построенной с помощью триангуляции Делоне, по точкам, в глубинах которых можно быть уверенными. В качестве этих точек как раз подходят ключевые.

2.5.2. Оценка положения очередного кадра

Для оценки положения нового кадра необходимо иметь какое-то начальное предположение для нового кдара, а затем улучшать его. Вместо инициализации путем оптимизации функции энергии (1), что дает локально-оптимальное решение, предлагается использовать инициализацию по ключевым точкам, так как для них существуют глобально-оптимальные оценки для расположения кадра относительно трехмерных точек.

2.5.3. Общая оптимизация

В оптимизации всех поз и глубин точек предлагается совместить остатки для ключевых точек (ошибки проецирования) и ошибки совпадения яркостей (для контрастных точек в прямом подходе);

2.5.4. Релокализация и замыкание циклов

Для релокализации и замыкания циклов предлагается использовать подход алгоритма ORB-SLAM2 [4], основанный на *наборе слов* (bag of

words) [1]. *Визуальными словами* будет являтся дискретизация пространства дескрпиторов, составляя так называемый *визуальный словарь*. *Словарь* создается в автономном режиме с дескрипторами ORB, извлеченными из большого набора изображений. Если изображения достаточно общие, один и тот же словарь можно использовать для разных сред, получая хорошую производительность. Система постепенно создает базу данных, в которой хранится каждое *визуальное слово* в *словаре* и то, в каких ключевых кадрах оно было просмотрено, так что запросы к базе данных могут выполняться очень эффективно.

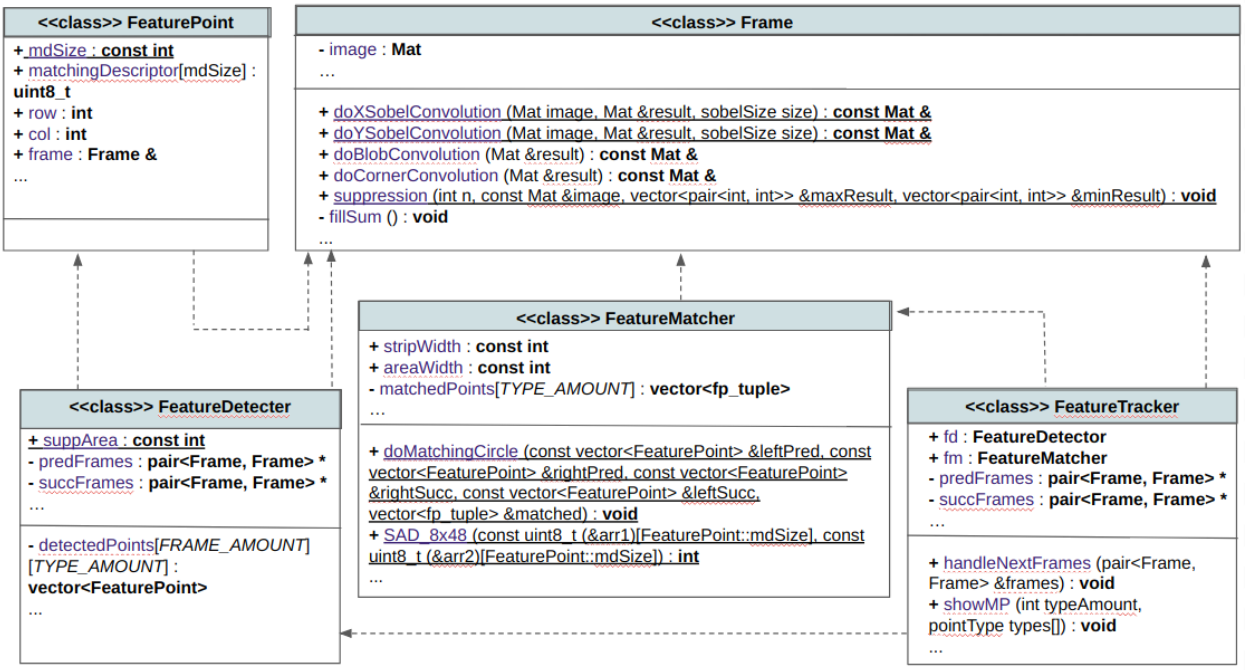
Таким образом, дополнительно с дескрипторами из SOFT2, предлагается использовать дескрипторы ORB.

3. Описание реализации

В качестве языка реализации был выбран язык C++. Он хорошо подходит для целей работы, так как для него есть несколько полезных библиотек алгоритмов компьютерного зрения, в частности, OpenCV. Более того, хотя у выбранного алгоритма SOFT2 и нет открытой реализации, она есть у Stereo-DSO [8] на языке C++, что может быть частично применимо в этой работе.

Сама реализация поставленных задач доступна в репозитории [3].

Упрощённая UML-диаграмма реализации представлена ниже:



При этом за обработку кадров отвечает класс *Frame*, за задачу детекции точек отвечает класс *FeatureDetector*, за поиск соответствий - *FeatureMatcher*. Класс *FeatureTracker* предоставляет функционал обработки последовательных стерео-кадров, которая состоит из детекции точек и поиска соответствий между кадрами. Сами ключевые точки представляются классом *FeaturePoint*.

4. Особенности реализации

4.1. Детекция ключевых точек

В качестве ключевых точек используются локальные минимумы и максимумы интенсивности изображений, полученных из оригинального при помощи двух сверток: для поиска углов и для поиска пятен, выделяя таким образоч 4 типа точек.

Рис. 3: Область исходного изображения.

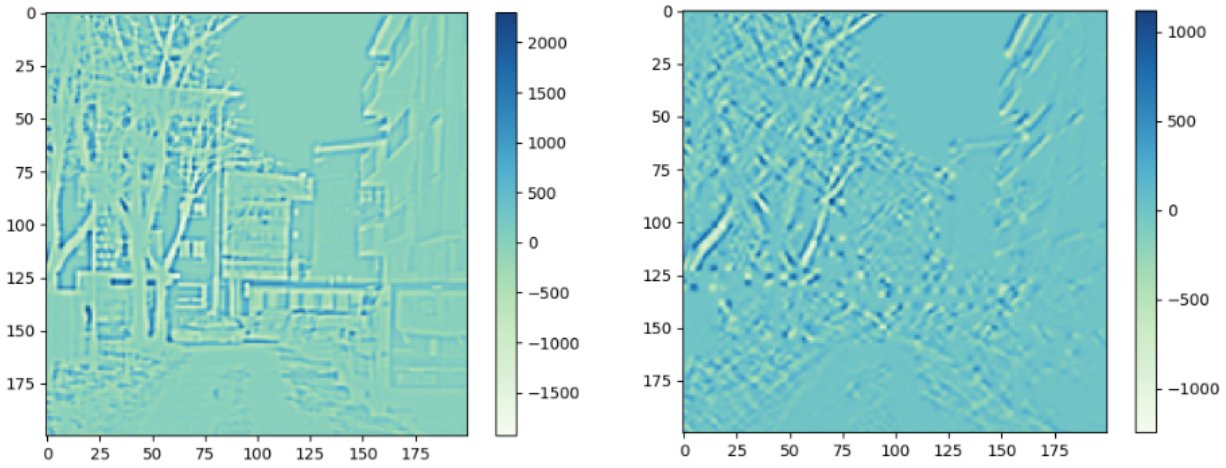


Рис. 4: Область изображения после обработки свертками для поиска углов и пятен.

Для оптимизации времени работы строилось интегральное изображение, а так же использовался тот факт, что свертка для поиска углов – сепарабельна.

Для нахождения локальных минимумов и максимумов был реализован алгоритм [5], позволяющий находить максимум и минимум в области рамером $(2n+1) \times (2n+1)$ за $4 - \frac{4}{n+1}$ операций на один пиксель, что при достаточно больших n является значением, независимым от размера области.

4.2. Дескрипторы ключевых точек

В качестве дескриптора ключевой точки использовались значения интенсивности в 24 точках около ключевой после обработки вертикальным фильтром Собеля и еще 24 – после обработки горизонтальным фильром Собеля, что в итоге дает вектор из 48 элементов. Дескрипторы двух ключевых точек сравнивались при помощи суммы абсолютных разностей дескрипторов (SAD). Большая часть времени работы всего алгоритма приходится как раз на исполнения этой функции, так как она применяется для поиска соответствий ключевых точек, которых детектируется порядка 11000 на один кадр. Она была оптимизирована при помощи intel intrinsics, что уменьшило время поиска соответствий на 40 % по данным профайлера perf.

При фильтрации Собеля также использовалась сепарабельность ядра этого фильтра. Для улучшения точности результатов выполнялась свертка ядром 5×5, а не 3×3, как в исходной статье, потому что это несильно увеличивает время исполнения. Однако в таком случае элементы десриптора имеют 32-битный размер, на котором SAD работает в ~ 4 раза медленнее, чем на 8-битных значениях.

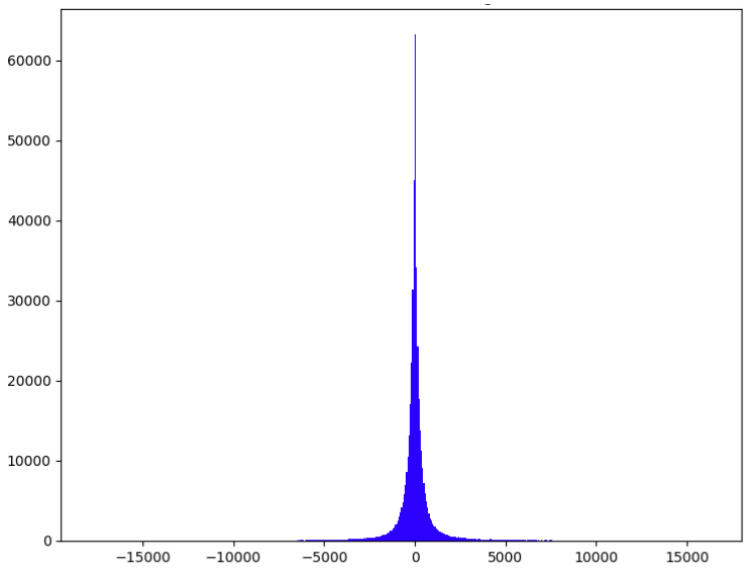


Рис. 5: Пример распределения градиента интенсивности на тестируемых изображениях, построенный при помощи matplotlib.

Так как фильтр Собеля дает приближенное значение градиента интенсивности изображения, то известно распределение полученных значений (рис. 5). Поэтому можно без особых потерь точности конвертировать значения в 8-битный формат, не учитывая данные с достаточно маленьким значением функции вероятности.

4.3. Поиск соответствий

Как было рассмотрено выше, соответствия между двумя стереокадрами ищутся по кругу: для каждой точки из нижнего левого изображения ищется наиболее подходящая точка из нижнего правого, для нее - из верхнего правого, далее - из верхнего левого и заключающая - из левого нижнего. Если круг замкнулся, то соответствия найдены.

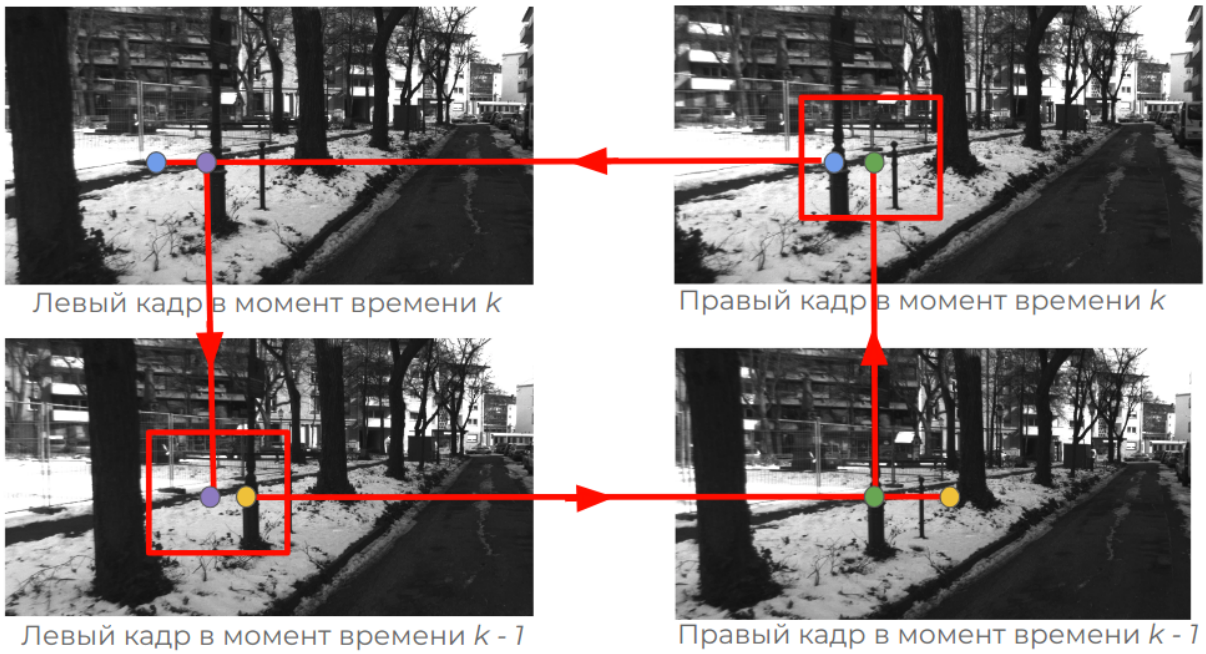


Рис. 6: Оптимизация алгоритма поиска соответствий, уменьшающая количество сравниваемых ключевых точек.

Чтобы не перебирать абсолютно все точки на кадрах, можно использовать ректифицированные изображения – стерео-изображения, эпиполярные линии которых находятся параллельно линии изображения. Тогда при поиске соответствий между изображениями из одной стереопары можно рассматривать точки, лежащие только на части эпиполярной прямой. А при поиске соответствий между верхними и нижними кадрами можно рассматривать лишь небольшую область вокруг точки, что существенно снижает количество точек, необходимых для рассмотрения, а, следовательно, и время работы алгоритма.

Список литературы

[1] Gálvez-López Dorian, Tardós Juan D. Bags of Binary Words for Fast Place Recognition in Image Sequences // IEEE Transactions on Robotics, vol. 28, no. 5, pp. 1188-1197. — 2012.

[2] Igor Cvišić, Josip Ćesić, Ivan Marković, Ivan Petrović. SOFT-SLAM: Computationally efficient stereo visual simultaneous localization and mapping for autonomous unmanned aerial vehicles // Journal of field robotics (1556-4959) 35 (2018), 4; 578-595. — 2017.

[3] Lyulina Elena. StereoSLAM: fusing of two approaches. — URL: `https://github.com/elena-lyulina/StereoSLAM`.

[4] Mur-Artal Raul, Tardós Juan D. ORB-SLAM2: an Open-Source SLAM System for Monocular, Stereo and RGB-D Cameras // IEEE Transactions on Robotics. — 2017.

[5] Neubeck A., Gool L. V. Efficient non-maximum suppression // ICPR 2006. — 2016.

[6] Nistér David. An Efficient Solution to the Five-Point Relative Pose Problem // IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 26, NO. 6, JUNE 2004. — 2004.

[7] Rui Wang, Martin Schwörer, Daniel Cremers. Stereo DSO: Large-Scale Direct Sparse Visual Odometry with Stereo Cameras. — 2017.

[8] Tu Jinge. Implementation of Stereo-DSO. — URL: `https://github.com/JingeTu/StereoDSO`.

[9] of Technology Karlsruhe Institute. Results of odometry algorithms testing on KITTI benchmark. — URL: `http://www.cvlibs.net/datasets/kitti/eval_odometry.php`.