

Санкт-Петербургский государственный университет

Математическое обеспечение и администрирование информационных систем

Системное программирование

Мухаматулин Максим Садриевич

Стабилизация видео с помощью 3D-моделирования сцены

Бакалаврская работа

Научный руководитель:
д. ф.-м. н., профессор Терехов А. Н.

Рецензент:
к. т. н., руководитель проектов Федоренко С. И.

Санкт-Петербург
2016

SAINT-PETERSBURG STATE UNIVERSITY

Software and Administration of Information Systems

Area of Specialisation Software Engineering

Maksim Mukhamatulin

Video stabilisation via 3D scene modelling

Bachelor's Thesis

Scientific supervisor:
professor Andrey Terekhov

Reviewer:
project manager Sergei Fedorenko

Saint-Petersburg
2016

Оглавление

Введение	4
1. Существующие решения	7
2. Постановка задачи	11
3. Модификация алгоритма генерации соответствий и их фильтрации	12
3.1. Анализ и модификация алгоритма фильтрации ложных соответствий	13
3.2. Изменение алгоритма генерации соответствий для характерных точек пары изображений	16
4. Создание искусственных примеров и метрики	19
4.1. Создание искусственного примера с помощью аффинного преобразования	19
4.2. Создание искусственной сцены с известными параметрами	21
4.3. Сравнение результатов работы алгоритма на разных этапах	23
Заключение	25
Список литературы	26

Введение

Почти всегда при съемке с движением камеры по сцене наблюдается тряска, обусловленная сложностью плавного перемещения камеры и возникающей из-за этого скачкообразной траектории движения и ориентации камеры.

Тряска затрудняет просмотр видеофрагмента и негативно воспринимается человеком, из-за чего её с момента зарождения кинематографа старались избежать.

В эпоху немого кино камера на протяжении всей сцены почти всегда оставалась неподвижной и меняла положение только со сменой дубля, а вся сцена получалась позже во время монтажа склейкой исходных записей [21]. Причиной этому были громоздкие размеры киноаппаратуры и отсутствие автоматического привода, из-за чего для приведения в движение грейфера, обтюлятора и лентопротяжного механизма оператору приходилось вручную вращать ручку во время съемки сцены.

С дальнейшим развитием киноаппаратуры, уменьшением её размера и становлением кино как нового вида изобразительного искусства появилась возможность перемещения камеры непосредственно во время съемки дубля. Для этого начали использоваться операторскую тележку, перемещаемую по заранее смонтированным рельсам, а позже и операторский кран [16].

В дальнейшем, в конце 1950-ых и начале 1960-ых, с появлением более мобильных камер, стало возможным переносить её без использования специального оборудования. Так как использование операторской тележки или крана было очень трудоемко и требовало большой предварительных работ по прокладке рельсов или установке крана, а также ограничивало подвижность камеры из-за того, что рельсы можно проложить далеко не везде и приходилось избегать их попадания в кадр, а при съёмке с рук или с плеча тряска была неизбежна, что могло сделать запись непригодным для дальнейшего использования, к 1977 году была создана и запатентована система стедикам [19] — носимая система стабилизации съёмочной камеры для видеосъёмки в движении.

С приходом любительской и цифровой кинокамеры появились также встроенные системы стабилизации, такие как SteadyShot [11], Vibration Reduction [14], Image Stabilizer [2] и другие, основанные на компенсации тряски либо с использованием «плавающей» линзы в объективе, либо с применением смещения оптического сенсора, либо комбинацией двух этих технологий. В случае же, если на необходимом видеофрагменте, снятом без использования упомянутых ранее технологий стабилизаций или недостаточно ими стабилизированным, применяют методы цифровой стабилизации на постобработке.

Стандартные методы стабилизации на постобработке, такие как Warp Stabilizer VFX [1], ReelSteady [18], Meralli [22], Deshaker [3] и другие, заключаются в слежении за положением характерных точек [10] на кадрах видеофрагмента и последующей их фиксации или сглаживании, то есть выравнивание скорости и траектории их движения. Таким образом, характерные точки на кадре становятся неподвижными или их траектории станут достаточно гладкими и предсказуемыми.

Эти методы демонстрируют приемлемые результаты для видео с незначительным перемещением по сцене. В случае же съемки в движении возникает эффект параллакса. Из-за этого характерные точки двигаются относительно друг друга, и сглаживание положения конкретной или средней точки приводит к очень сильному искажению геометрии видеофрагмента. Также, из-за движения, точки постепенно выходят за границы кадра, и появляется необходимость в поиске новых характерных точек, из-за чего непрерывный стабилизированный видеофрагмент может прерваться. А появление на сцене движущихся объектов может заставить алгоритм интерпретировать его как часть сцены, и он попытается сгладить его движение относительно остальной сцены. По этим причинам такой метод стабилизации является неприемлемым для определенных видеофрагментов.

В последнее время широкую популярность приобрела техника съемки hyperlapse — метод замедленной киносъемки в движении, названный по аналогии с timelapse — статичной замедленной киносъемкой. Из-за ускорения видеофрагмента проблема стабилизации встает наиболее

остро, так как из-за более быстрого перемещения камеры тряска многократно усиливается, что делает этот видеофрагмент непригодным для использования без постобработки.

Техника hyperlapse может использоваться в аэросъемке и в автомобильных регистраторах. Но основная область применения этой техники является интерактивное искусство. Например, видеофрагменты, снятые с применением hyperlapse, используются в телевидении как фон для прогноза погоды [25]. Особенно данная методика популярна при съемке архитектуры, так как hyperlapse позволяет быстро дать представление о форме здания любого размера. Поэтому на видеохостингах можно найти hyperlapse видео почти любого города мира, знакомящее зрителя с основными его достопримечательностями.

Цель настоящей работы заключается в улучшении метода стабилизации видеофрагмента с помощью реконструкции 3D-сцены, позволяющего получать видео в технике hyperlapse на постобработке без использования специальных техник съемки. При этом подразумевается, что исходный видеофрагмент снят со стандартной кадровой частотой (24-60 кадров в секунду), и после обработки будет значительно ускорен (с сохранением кадровой частоты, но уменьшением продолжительности видеоролика) для создания эффекта hyperlapse, для съемки которого, обычно, исходный видеофрагмент изначально обладает небольшим количеством исходных кадров со значительными промежутками между ними. Улучшения будут сконцентрированы в методе построения и фильтрации соответствий характерных точек, с помощью которых производится реконструкция 3D-сцены с вычислением положения камеры в каждом кадре для дальнейшего сглаживания траектории движения камеры и склейки результирующих кадров по информации о геометрии сцены.

1. Существующие решения

Эта работа является продолжением прошлогодней курсовой работы [24], поэтому кратко напомню о достигнутых ранее результатах.

На первом этапе в прошлом году, с целью ознакомления с инструментарием, предоставляемым OpenCV, была предпринята попытка реализации наивного метода стабилизации видео, основанного на применении матрицы гомографии, которая наилучшим образом преобразовывала координаты соответствующих характерных точек на соседних изображениях. Делалось это с целью генерации промежуточных кадров для видео и достижения эффекта, получаемого с помощью сервиса Photosynth [17].

После наложения следующего кадра с применением матрицы гомографии с коэффициентом и полупрозрачностью, с целью улучшения качества получаемых промежуточных кадров, был реализован морфинг [20] соседних кадров, карта смещений для которого вычислялась с помощью алгоритма Farneback [4], дающего в результате грубую карту оптического потока. Результат применения морфинга на соседних изображениях не оправдал ожиданий, и было принято решение от него отказаться. А так как результаты, полученные с использованием только матрицы гомографии, ничем принципиально не отличались от стандартных методов стабилизации видео, был рассмотрен следующий, значительно более интеллектуальный метод стабилизации видеофрагментов.

Microsoft Research был представлен метод стабилизации видео в движении [9], основанный на реконструкции 3D сцены и положений камеры для каждого кадра в течении всего видео с последующим сглаживанием траектории движения и ориентации камеры и склейки выходного изображения из избыточного количества имеющихся кадров (так как видеофрагмент ускоряется, входных кадров значительно больше, чем будет на выходе). Данный метод не зависит от ориентации камеры и скорости появления новых точек в кадре, а также от величины тряски и её частоты (разве что это может привести к смазыванию кадра выходного видеофрагмента). Также для восстановления сцены подразу-

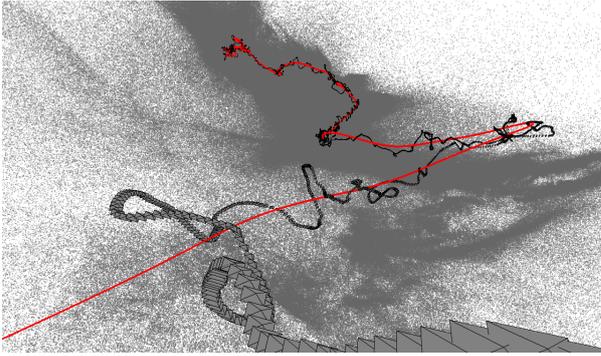


Рис. 1: Реконструкция 3D сцены

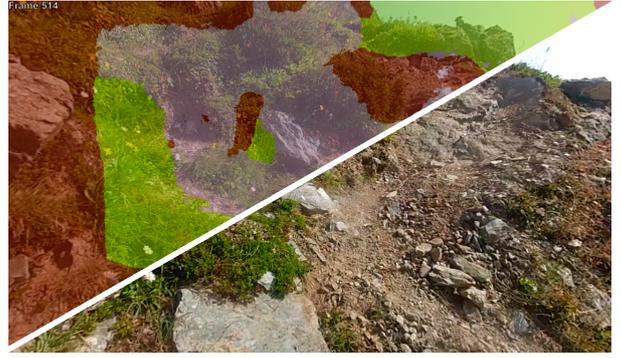


Рис. 2: Склейка результирующего кадра

меваются, что камера не калибрована, то есть её внутренние параметры неизвестны.

То есть общий план алгоритма состоит из следующих этапов:

1. Реконструкция 3D сцены и положения камеры с помощью кадров видеофрагмента (рисунок 1)
2. Оптимизация траектории перемещения и ориентации камеры, полученных на предыдущем этапе
3. Склейка результирующего кадра из всех имеющихся кадров видеофрагмента (рисунок 2)

В прошлом году был реализован первый этап алгоритма, который на тот момент состоял из

1. Нахождения характерных точек для каждого кадра видеофрагмента. Этот этап был полностью позаимствован из реализации наивного метода стабилизации видео, то есть использовался SIFT [10] стандартной реализации OpenCV
2. Построения соответствий между характерными точками на соседних кадрах, реализация которого тоже была заимствована из наивного метода стабилизации, то есть использовался Flann [12] для поиска наиболее схожей по дескриптору парной характерной точки

3. Фильтрации ложных соответствий, которая состояла из двух последовательных фильтраций

- (a) Первый этап фильтрации заключался в отбрасывании худших по схожести соответствий (всех, что в три раза хуже самого лучшего соответствий) [5]. Коэффициент для этой фильтрации был подобран эмпирическим путем таким образом, чтобы после применения фильтрации оставалось достаточное количество характерных точек для работы с ними на следующих этапах алгоритма. То есть выбор такого коэффициента был малообоснованным, как, впрочем, и вся эта фильтрация, основанная на сравнении дескрипторов найденных характерных точек. Позже от этого этапа фильтрации было решено отказаться
- (b) Следующий этап фильтрации ложных соответствий основывался на применении к найденным соответствиям эпполярного ограничения [23], смысл которого заключается в том, что если два соответствия для двух пар характерных точек на соседних изображениях корректны, то соответствующие точки для характерных точек, лежащих на прямой, проведенной через характерные точки первого изображения, должны лежать на прямой, проходящей через две соответствующие характерные точки второго изображения (рисунок 3). В случае же, если рассматривается множество всех характерных точек и их соответствий на паре изображений, то для них должно быть верно, что все прямые, проведенные через пары точек всех корректных соответствий, должны иметь одну общую точку пересечения (которая не обязательно должна лежать на самом изображении) (рисунок 4). В алгебраических определениях это ограничение описывается как

$$x_1^T F x_2 = 0 \quad (1)$$

где x_1 и x_2 – однородные координаты характерных точек од-

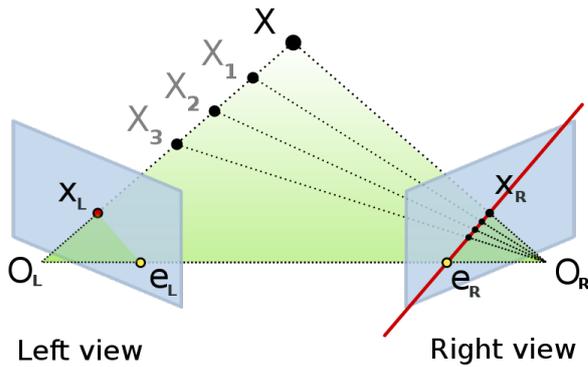


Рис. 3: Геометрический смысл эпиполярного ограничения [15]

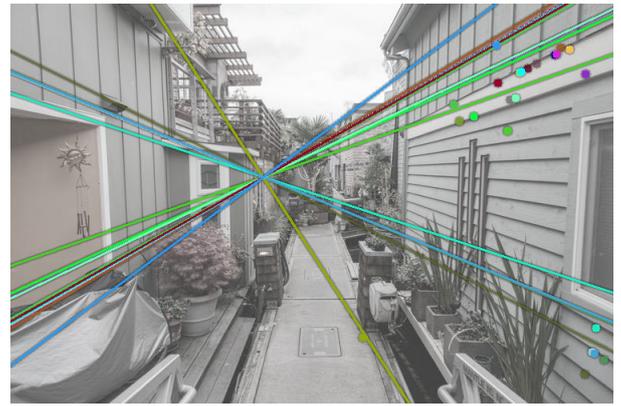


Рис. 4: Эпиполярные линии на изображении

ного корректного соответствия на первом и втором изображении соответственно, а F – матрица размером 3×3 , называемая фундаментальной матрицей. То есть фильтрация заключалась в поиске фундаментальной матрицы, которая удовлетворяла уравнению (1) для как можно большего количества соответствий. Поиск фундаментальной матрицы был реализован с помощью алгоритма RANSAC [6]. После её получения отфильтровываются соответствия, не удовлетворяющие уравнению (1) с некоторой погрешностью

4. Построение разреженного облака точек, перед которым, для получения грубых внешних параметров камеры, был реализован метод их извлечения из полученной на предыдущем этапе фундаментальной матрицы [8] с помощью сингулярного разложения. Далее эти черновые внешние параметры камеры вместе с координатами прошедших фильтрацию характерных точек обрабатывались алгоритмом bundle adjustment [13], результатом которого было разреженное облако характерных точек в пространстве и положения камер для каждого кадра последовательности изображений

2. Постановка задачи

Из-за большого количества ложных соответствий, остающихся после фильтрации, и вычисления некорректной фундаментальной матрицы, вкупе с тем, что после фильтрации соответствий для реконструкции геометрии сцены оставалось очень мало, результат применения алгоритма был довольно нереалистичным и трудно проверяемым.

Цель настоящей работы заключается в улучшении ранее реализованного метода построения соответствий для найденных характерных точек и изменении алгоритма их фильтрации для увеличения количества и повышения качества корректных соответствий, остающихся после фильтрации и используемых в дальнейшем в реконструкции 3d сцены.

Также необходимо реализовать метод проверки корректности построенного облака точек с помощью создания искусственных примеров с известными параметрами и сравнения их с результатами, полученными после вычисления разряженного облака характерных точек в пространстве.

3. Модификация алгоритма генерации соответствий и их фильтрации

Как уже было сказано ранее, недостатком алгоритма прошлого года, что после этапа фильтрации на обработку алгоритму bundle adjustment передавалось недостаточное количество соответствий, из-за чего, после получения разреженного облака точек, результаты, даже если они и были правдоподобны, было трудно проверить. Количество данных, с которыми работали различные этапы алгоритма, выглядели так:

- Поиск характерных точек ≈ 3000 точек для каждого изображения из последовательности (рисунок 5a и 5b)
- Генерация соответствий ≈ 3000 соответствий для пары изображений (точно такое же количество соответствий, сколько было найдено характерных точек у первого изображения из пары) (рисунок 5c)
- Фильтрация с помощью ограничения по схожести дескриптора ≈ 700 соответствий (рисунок 5d)
- Фильтрация с помощью эпиполярного ограничения ≈ 70 соответствий (рисунок 5e)
- Результат применения bundle adjustment ≈ 70 точек в пространстве (не изменяет предыдущее число)

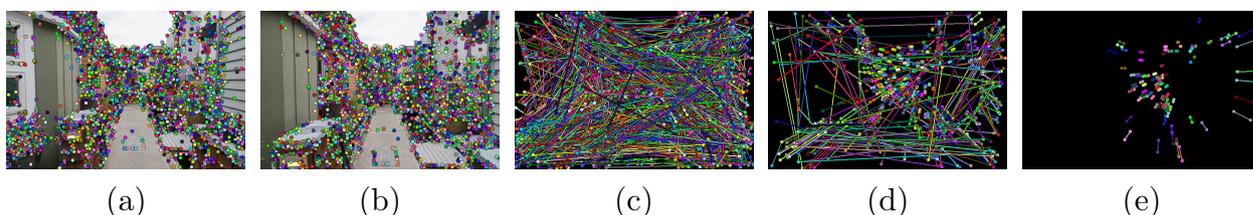


Рис. 5: Характерные точки и соответствия на разных этапах фильтрации

Эти цифры верны для реального тестового примера [7], снятого в технике hyperlapse, на котором в течении всей работы проверялся алгоритм и который соответствует входным данным, которые должны по плану корректно обрабатываться. На других примерах цифры могут значительно отличаться, так как коэффициенты фильтрации не фиксированы, и зависят от специфики входных данных.

То есть при такой реализации фильтрации характерных точек после их обработки получалось облако, содержащее только примерно 2,3% от количества исходных характерных точек. Это значение неоправданно мало, так как, в теории, из-за совсем незначительного изменения положения камеры на сцене количество характерных точек не должно кардинально меняться. То есть если характерных точка присутствовала на первом изображении, она с очень высокой вероятностью (более 80%) будет присутствовать и на втором изображении из пары. 20% характерных точек могли либо изначально ложно определены (например, из-за шума), либо выйти за границы кадра из-за смены положения камеры. Основная масса точек присутствует на каждом изображении из пары, а это значит, что для каждого из них должно быть построено корректное соответствие, которое должно остаться после фильтрации и с помощью которого будет получено положение точки в пространстве относительно камеры.

Так как 2,3% остающихся после фильтрации соответствий от изначального количества характерных точек слишком мало для того, чтобы делать какие-то утверждения о качестве работы алгоритма, стало очевидно, что нужно проанализировать и изменить некоторые этапы работы алгоритма для улучшения этого показателя. Было решено, прежде всего, разобрать и переосмыслить работу фильтрации.

3.1. Анализ и модификация алгоритма фильтрации ложных соответствий

Так как эпиполярная фильтрация теоретически обоснована, у этого этапа была проверена только корректность передачи данных и реали-

стичность получаемых в результате данных. В частности, были перепроверены количественные соотношения соответствий, удовлетворяющих неравенству

$$x_1^T F x_2 < \varepsilon \quad (2)$$

для различных ε для получаемого в результате работы алгоритма фундаментальной матрицы F , где x_1 и x_2 – однородные координаты характерных точек одного соответствия. Также была проверена корректность работы алгоритма, использующего RANSAC, посредством сравнения результатов его работы с результатами работы 7-точечного, 8-точечного и Least Median of Squares алгоритмами. Как и ожидалось, RANSAC дает лучшие результаты по количеству удовлетворяющих уравнению (2) количеству соответствий.

В отличие же от эпиполярной фильтрации, фильтрация с помощью порога по схожести дескриптора не так хорошо обоснована. Она была реализована еще на этапах реализации наивного алгоритма стабилизации с помощью с целью уменьшения количества входных данные для ускорения работы алгоритма, которому, на тот момент, большое количество данных для нахождения матрицы гомографии не было необходимо.

В дальнейшем, при анализе остающихся после применения этой фильтрации соответствий, стало понятно, что очень часто она срабатывает некорректно, оставляя ложные соответствия с идентичными дескрипторами (рисунок 5d). Особенно это было заметно на примере с повторяющейся текстурой, на которой фильтрация оставляла некорректные соответствия, соответствующие схожему участку текстуры, и отбрасывала корректные, на которых дескриптор незначительно изменялся из-за смещения положения камеры.

В алгоритме SIFT дескриптор характерной точки представляет собой набор коэффициентов, описывающих гистограмму градиентов в окрестности точки (рисунок 6). Таким образом выходило, что порог устанавливался по почти идеальному соответствию этих дескрипторов, которые все же относились к некорректному соответствию. Из-за этого терялось значительное количество потенциально верных соответствий

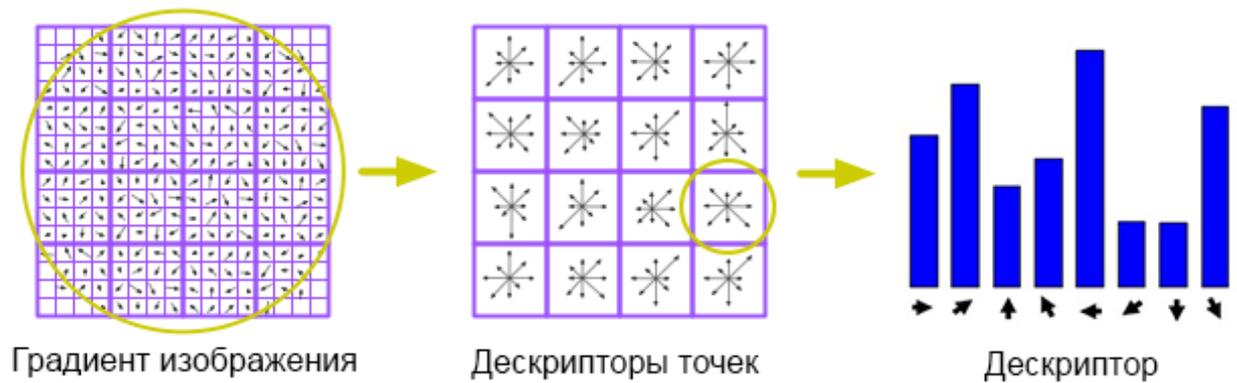


Рис. 6: Дескриптор характерной точки в алгоритме SIFT

и оставалось большое количество неверных, из-за чего после фильтрации эпиполярным ограничением, даже в случае, если фундаментальная матрица находилась корректно, количество остающихся соответствий становилось чересчур малым. Так было установлено, что фильтрация с помощью порога схожести дескриптора не оправдывает себя, и от него было решено отказаться.

Таким образом в алгоритме осталось 4 этапа работы с одним этапом фильтрации. Последовательность и количество данных на каждом из этапов стало выглядеть так:

- Поиск характерных точек ≈ 3000 точек на каждом изображении
- Генерация лучших соответствий ≈ 3000 соответствий для пары изображений
- Фильтрация с помощью эпиполярного ограничения ≈ 500 соответствий
- Результат применения bundle adjustment ≈ 500 точек в пространстве

Но нельзя было говорить о значительном улучшении получаемых результатов. Дело в том, что по результатам работы эпиполярной фильтрации становится очевидно, что находится фундаментальная матрица, не соответствующая действительности (такой вывод можно делать уже хотя бы по внешним параметрам камеры, которые позже извлекаются

из фундаментальной матрицы). Из-за этого после фильтрации не остается почти не одного корректного соответствия, и после применения bundle adjustment геометрия реконструированной сцены совершенно не соответствовала действительности.

Проблема здесь кроется в генерации соответствий по имеющимся характерным точкам двух изображений. А конкретно в том, что для каждой характерной точки первого изображения строится соответствие с наиболее схожей по дескриптору характерной точкой второго изображения. При этом не учитывается ни местоположение точки, ни наличие у этой характерной точки множества других, чуть менее, но все же довольно схожих характерных точек. А приводит это к тому, что в множестве генерируемых соответствий изначально присутствуют большое количество ложных, полученных из-за шума, из-за схожих участков текстуры на различных участках изображения и из-за характерных точек, которые пропадают либо появляются на парном изображении. Конечно, в этом множестве все еще присутствует довольно большое количество и корректных соответствий, вот только фильтрация эпиполярным ограничением не справляется с их отделением от ложных, что было довольно предсказуемо, так как данные получаются настолько зашумленными, что алгоритм RANSAC не в силах отделить выбросы от точек, удовлетворяющих модели. Из-за этого находится неверная фундаментальная матрица и строится некорректное облако точек.

3.2. Изменение алгоритма генерации соответствий для характерных точек пары изображений

Для решения этой проблемы было решено модернизировать алгоритм следующим образом:

- Для каждой характерной точки, вместо одного лучшего соответствия на изображении, искать некоторое количество лучших
- Перед фильтрацией эпиполярным ограничением добавить фильтрацию, которая бы оставляла только бы соответствия, расстоя-

ния между характерными точками которых не превышает определенную величину

Первый пункт решает проблему, из-за которой для характерных точек, принадлежащих повторяющейся текстуре, из-за того, что находились только первые и единственные соответствия с лучшими совпадениями по дескриптору, которые при этом были некорректными, их корректные соответствия не могли участвовать в дальнейшей обработке, несмотря на то, что, помимо первого соответствия, у каждой характерной точки было множество потенциальных других соответствий, одно из которых для каждой точки, вероятно, было корректно. Теперь же для участков с повторяющейся текстурой для каждой точки будет найдено несколько соответствующих характерных точек, а это значит, что, если фильтрация эпполярным ограничением отработает корректно, то из всех найденных для этой характерной точки соответствий останется одно единственное верное соответствие. Из-за этого, даже после фильтрации, количество остающихся соответствий незначительно меньше количества изначально имеющихся характерных точек.

Однако же, если попробовать отфильтровать эти данные эпполярным ограничением без предварительной обработки, то она отработает некорректно. Так как в таком случае во входных данных алгоритма RANSAC менее 10% соответствий будут корректными, что значит, что алгоритм не сможет разделить данные на корректные и ложные. Для решения этой проблемы и был добавлен второй пункт, то есть предварительная фильтрация перед эпполярным ограничением, основанная на ограничении расстоянии, которое может быть у соответствия. Это ограничение теоретически обоснованно, так как известно, что между кадрами внешние параметры камеры, то есть положение в пространстве относительно сцены и ориентация, не могут кардинально измениться, из чего будет следовать, что положение большего количества точек на изображении не будет сильно меняться. Из этого следует, что соответствия с большой длиной скорее ложные, нежели корректные, и их стоит отфильтровать до фильтрации эпполярным ограничением.

В итоге после этих изменений алгоритм состоит из пяти этапов.

План и приблизительное количество данных на каждом этапе стали выглядеть так:

- Поиск характерных точек ≈ 3000 точек на каждом изображении
- Генерация соответствий ≈ 30000 соответствий для пары изображений
- Фильтрация с помощью ограничения на длину соответствия ≈ 2600 соответствий
- Фильтрация с помощью экиполярного ограничения ≈ 1000 соответствий
- Результат применения bundle adjustment ≈ 1000 точек в пространстве

Также заметно повысилось качество данных передаваемых алгоритму bundle adjustment (что заметно по оставшимся после фильтрации соответствиям и полученной фундаментальной матрице, которая также довольно реалистично раскладывается на внушающие доверие внешние параметры камеры) и результат, получаемый после его применения.

4. Создание искусственных примеров и метрики

До текущего момента качество работы каждого этапа алгоритма можно было проверить лишь субъективно. Дело в том, что, например, по количеству соответствий, остающихся после фильтрации, нельзя говорить о качестве этой фильтрации, так как оставшиеся соответствия, несмотря на их большое количество, все могут оказаться ложными, из-за чего все последующие этапы алгоритма тоже будут давать не соответствующий действительности результат. Поэтому есть необходимость в создании какой-то метрики, позволяющей оценивать качество одного или всех этапов работы алгоритма, для фиксации улучшений в работе алгоритма в процессе его модернизации.

4.1. Создание искусственного примера с помощью аффинного преобразования

Первым делом было решено создать искусственный пример на основе натурального примера за счет отказа от пространственной координаты. Реальный пример служит основой для того, чтобы на изображении присутствовало достаточное количество уникальных характерных точек, так как создать искусственную текстуру с уникальными дескрипторами на различных участках изображения достаточно проблематично. Основой послужило изображение из привычного стандартного примера [7].

Для создания примера с этим изображением было необходимо произвести некоторые манипуляции. Так как идея заключалась в создании примера без пространственной координаты, то есть создании плоского примера, нужно было, чтобы координаты характерных точек второго изображения могли быть получены с помощью аффинного преобразования координат характерных точек первого изображения. Это значит, что само изображение должно быть получено с помощью применения аффинного преобразования к другому изображению. После чего SIFT

должен будет выделить примерно одни и те же характерные точки (из-за изменения масштаба алгоритм может также начать находить новые точки и не находить старые), `flann` должен будет построить лучшие соответствия для каждой характерной точки (считается, что соответствие с корректной соответствующей характерной точкой будет в этом множестве присутствовать, так как они почти идентичны за исключением масштаба и поворота), фильтрация по длине соответствий оставит только соответствия, находящиеся рядом с неподвижной точкой (которая, предсудотрительно, находится в границах изображений), фильтрация с помощью эпиполярного ограничения выделит множество соответствий, указывающих в направлении неподвижной точки, и в конце множество этих соответствий будут отправлены на обработку `bundle adjustment`, который, из-за того, что координаты характерных точек соответствия можно выразить с помощью аффинного преобразования, должен расположить их в одной плоскости. То есть в данном примере имитируется, что с помощью камеры была снята какая-то плоская картина, висящая на стене, с двух позиций. Невзирая на сюжет картины и её глубинный смысл, алгоритм должен распознать, что это плоское изображение, так как несмотря на изменение положения камеры не возникает эффекта параллакса.

Если какой-то из этапов алгоритма работает неправильно, то в результате не получится плоской стены из пространственных координат характерных точек. Таким образом результат очень просто поддается проверке.

Из-за того, что одно изображение является обрезанной и смещенной копией другого, то в результате фильтрации соответствий их остается значительно больше, чем на реальном примере. По результату, получившемуся после обработки этого искусственного примера, можно говорить, что алгоритм отработал корректно.

4.2. Создание искусственной сцены с известными параметрами

После успешной проверки работы алгоритма на сцене, в которой все характерные точки расположены в одной плоскости, было необходимо произвести проверку на более реалистичных и сложных примерах, где предметы располагались на произвольных расстояниях и имели бы сложную и разнообразную форму.

Проверку корректности расположения характерных точек в пространстве было решено провести с помощью генерации карты глубины сцены. То есть после получения координат характерных точек и камеры в пространстве, нужно для каждой характерной точки получить его дистанцию от камеры и двумерные координаты проекции этой точки на изображение, совпадающего по размеру с картой глубины, для камеры с вычисленным положением. После этого необходимо для всех точек сравнить дистанции, вычисленные с помощью положения характерной точки в пространстве и полученных из карты глубины по вычисленным координатам проекции этой характерной точки. Для этих двух дистанций должна наблюдаться некоторая закономерность, зависящая от метода представления расстояний на карте глубины.

Существуют различные методы получения карты глубины для реальной сцены, но, чтобы не усложнять себе жизнь, было решено создать искусственный пример с помощью инструмента трехмерного моделирования, который предоставляет большие возможности для визуализации карты глубины сцены.

Сначала была попытка создания простого примера с небольшим количеством объектов. Сцена состояла из диагонально развернутого массива размером $4 \times 4 \times 4$ примитивов различного цвета (рисунок 7). Для первого кадра камера оставалась в нулевой позиции, для второго камера смещалась вправо, вверх и вперед на 2, 4 и 7 метров соответственно, а также меняла ориентацию на тангаж 3° и рыскание -8° . Так как объектами являлись простые примитивы без определенной текстуры, возникли проблемы на этапе поиска характерных точек, их было относительно

мало (50 штук) и они находились на границах объектов или их границах (рисунок 8). Также из-за малого разнообразия объектов возникли проблемы на этапе генерации соответствий – границы каждого объекта были довольно схожи, из-за чего алгоритм flann для каждой характерной точки находил слишком большое количество соответствий, из-за чего на далее фильтрация не справлялась с выделением корректных соответствий (рисунок 9), в результате чего в результатах сравнения координат характерных точек в пространстве с картой глубине не обнаруживалось какой-либо закономерности.

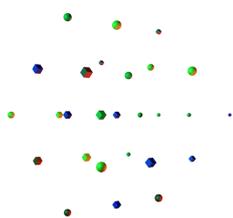


Рис. 7: Кадр тестового искусственного примера



Рис. 8: Характерные точки искусственного примера

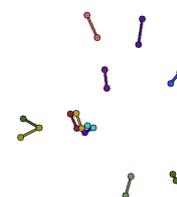


Рис. 9: Соответствия искусственного примера после фильтрации

В данном случае проблема заключалась не в некорректной работе алгоритма, а в неправильно построенной тестовой сцене, из-за чего алгоритм уже на первых этапах работы давал неудовлетворительные результаты. Было решено переделать сцену с использованием большого количества разнообразных контрастных текстур, которые увеличили бы количество находимых SIFT-ом характерных точек на изображении с уникальными дескрипторами. Также было решено ограничиться одним типом примитивов на сцене, увеличить их количество, изменить их расположение и добавить случайное смещение и ориентацию для уменьшения количества находимых характерных точек с идентичными дескрипторами. Все эти меры предоставляют лучшие возможности для сравнения результатов по карте глубины. В результате для этого примера на графике сравнения глубин характерных точек наблюдается явная зависимость.

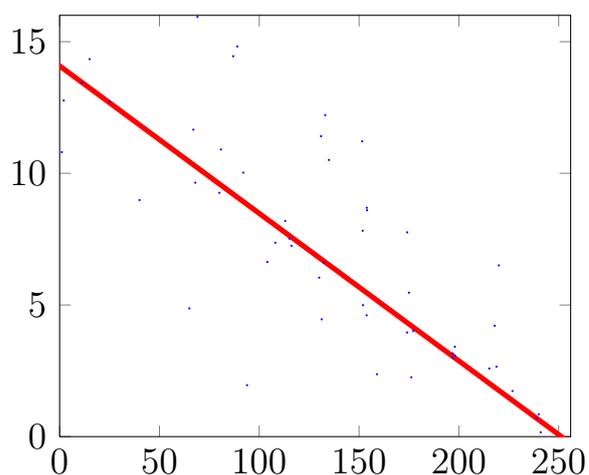


Рис. 10: График сравнения глубин старого алгоритма

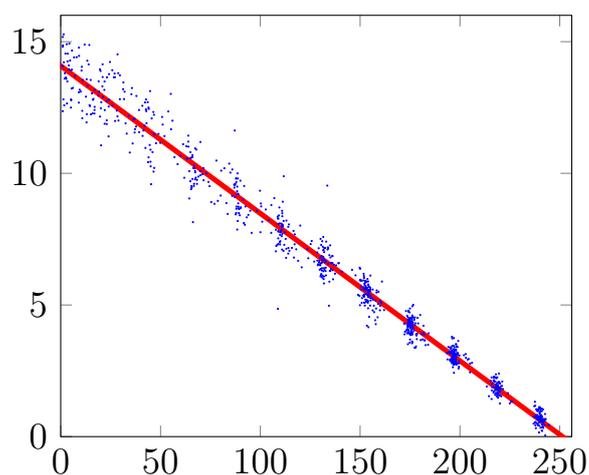


Рис. 11: График сравнения глубин нового алгоритма

4.3. Сравнение результатов работы алгоритма на разных этапах

После реализации этого метода сравнения, позволяющего получать объективную оценку работы алгоритма, его можно использовать как метрику. Ниже приведены результаты сравнения работы алгоритма на разных этапах его реализации с количественными и качественными показателями:

Этап модификации алгоритма	Количество точек	Суммарная ошибка	Среднеквадратичная ошибка
Прошлогодняя модификация	53	430,456	2,889
Без первого этапа фильтрации	482	1798,969	3,732
Генерация множества соответствий	3795	7050,329	3,451
Финальная модификация	929	289,452	0,558

Рис. 12: Сравнение результатов

Суммарная ошибка получена как сумма квадратов дистанции от точек до прямой линейной регрессии вида

$$f(x) = -0,0560053x + 14,0791 \quad (3)$$

которая в свою очередь была получена с помощью метода наименьших квадратов по данным, являющимся результатом сравнения глубин координат характерных точек в пространстве финальной модификации алгоритма с картой глубины сцены.

Заключение

В рамках данной работы были получены следующие результаты:

- Улучшен метод построения и фильтрации соответствий характерных точек
- Созданы искусственные тестовые примеры с известными параметрами
- Реализован дополнительный функционал для проверки корректности работы алгоритма
- Проведено сравнение качества работы различных модификаций алгоритма

Таким образом были достигнуты все поставленные в данной работе цели. Полученное на выходе разряженное облако точек значительно превосходит по плотности результаты прошлого года и, как выяснилось при сравнении, лучше соответствует реальной геометрии снятой сцены. В дальнейшем эти данные могут использоваться для ректификации изображений с целью получения плотного облака точек в каждом кадре, что позволило бы точнее установить геометрию сцены для использования её в склейке результирующих кадров из избыточных кадров исходного видеофрагмента.

Список литературы

- [1] Adobe. Stabilize shaky video footage // Adobe After Effects CC tutorials. — URL: <https://helpx.adobe.com/after-effects/how-to/stabilize-footage.html> (online; accessed: 2.05.2016).
- [2] Canon. Image Stabilization // Canon Professional Network. — URL: http://cpn.canon-europe.com/content/education/infobank/lenses/image_stabilisation.do (online; accessed: 2.05.2016).
- [3] Deshaker - video stabilizer. — URL: <http://www.guthspot.se/video/deshaker.htm> (online; accessed: 2.05.2016).
- [4] Farneback Gunnar. Two-frame motion estimation based on polynomial expansion // Image analysis. — Springer, 2003. — P. 363–370.
- [5] Feature Matching with FLANN. — URL: http://docs.opencv.org/2.4/doc/tutorials/features2d/feature_flann_matcher/feature_flann_matcher.html (online; accessed: 2.05.2016).
- [6] Fischler Martin A, Bolles Robert C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography // Communications of the ACM. — 1981. — Vol. 24, no. 6. — P. 381–395.
- [7] Fremont Houseboats hyperlapse sample sequence. — URL: <https://github.com/MaxWellHays/VideoStabilisationVia3DSceneModelling/tree/master/Sample> (online; accessed: 2.05.2016).
- [8] Hartley R. I., Zisserman A. Multiple View Geometry in Computer Vision. — Second edition. — Cambridge University Press, 2004.
- [9] Kopf Johannes, Cohen Michael F, Szeliski Richard. First-person hyperlapse videos // ACM Transactions on Graphics (TOG). — 2014. — Vol. 33, no. 4. — P. 78.

- [10] Lowe David G. Distinctive image features from scale-invariant keypoints // International journal of computer vision.— 2004.— Vol. 60, no. 2.— P. 91–110.
- [11] Macmanus Christopher. The technology behind sony alpha dslr's steadyshot inside.— 2009.— URL: <http://www.sonyinsider.com/2009/12/17/the-technology-behind-sony-alpha-dslrs-steadyshot-inside/> (online; accessed: 2.05.2016).
- [12] Muja Marius, Lowe David G. Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration. // VISAPP (1).— 2009.— Vol. 2.— P. 331–340.
- [13] Multicore bundle adjustment / Changchang Wu, Sameer Agarwal, Brian Curless, Steven M Seitz // Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on / IEEE.— 2011.— P. 3057–3064.
- [14] Nikon. VR (Vibration-Reduction).— 2008.— URL: http://www.nikon.com/about/technology/rd/core/software/vr_e/ (online; accessed: 2.05.2016).
- [15] Nordmann Arne. Epipolar geometry.svg // Wikipedia, the free encyclopedia.— 2007.— URL: https://commons.wikimedia.org/wiki/File:Epipolar_geometry.svg (online; accessed: 2.05.2016).
- [16] Orfano Finn. History of Camera Movement, Ed. by Rhonda Callow.— 2010.— URL: <http://www.brighthub.com/multimedia/video/articles/61492.aspx> (online; accessed: 2.05.2016).
- [17] Photosynth - Capture your world in 3D.— URL: <https://photosynth.net/> (online; accessed: 2.05.2016).
- [18] ReelSteady.— URL: <http://www.reelsteady.com/> (online; accessed: 2.05.2016).

- [19] Tustin Kevin. History of steadicam. — URL: <http://tiffen.com/steadicam/history-of-steadicam/> (online; accessed: 2.05.2016).
- [20] Wikipedia. Morphing // Wikipedia, the free encyclopedia. — 2008. — URL: <https://en.wikipedia.org/wiki/Morphing> (online; accessed: 2.05.2016).
- [21] Winokur M., Holsinger B.W. The Complete Idiot's Guide to Movies, Flicks, and Film. — The Complete Idiot's Guide Lifestyle Series. Alpha Books, 2000. — ISBN: 9780028639888. — URL: <https://books.google.ru/books?id=o7eg-0SrN4gC>.
- [22] proDAD Mercalli V2 - 3D image stabilizer, rolling shutter removal, zoom optimisation, pan shot optimisation. — URL: http://www.prodad.com/home/products/video_optimization/300391667,1-us.html (online; accessed: 2.05.2016).
- [23] A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry / Zhengyou Zhang, Rachid Deriche, Olivier Faugeras, Quang-Tuan Luong // Artificial intelligence. — 1995. — Vol. 78, no. 1. — P. 87–119.
- [24] Мухаматулин М.С. Hyperlapse. Стабилизация видео с помощью 3D-моделирования сцены. — 2015. — URL: <http://se.math.spbu.ru/SE/YearlyProjects/spring-2015/YearlyProjects/2015/344/344-Muhamatulin-report.pdf> (online; accessed: 2.05.2016).
- [25] Погода «Вести Москва». — 2015. — URL: <http://www.vizart.tv/project/promaxbda> (online; accessed: 2.05.2016).