

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ АКАДЕМИЧЕСКИЙ
УНИВЕРСИТЕТ РОССИЙСКОЙ АКАДЕМИИ НАУК»
ЦЕНТР ВЫСШЕГО ОБРАЗОВАНИЯ

КАФЕДРА МАТЕМАТИЧЕСКИХ И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

Серебро Андрей Евгеньевич

**Определение положения и ориентации опор
линий электропередач средствами
компьютерного зрения**

Магистерская диссертация

Допущена к защите.

Зав. кафедрой

д.ф-м.н., профессор Омельченко А. В.

Научный руководитель:

Начальник отдела ПО ООО "Геоскан"

Рыкованов Д. П.

Рецензент:

Ведущий инженер-программист АО "КТ-БС"

Романов А. А.

Санкт-Петербург

2017

Оглавление

1	Список терминов сокращений	5
1.1	Термины	5
1.2	Сокращения	5
2	Введение	6
2.1	Цель и мотивация	6
2.2	Постановка задачи	7
3	Связанные работы	8
3.1	Методы, использующие лидар	9
3.1.1	Классификация облака точек	9
3.1.2	Реконструкция трёхмерной сцены	9
3.2	Методы фотограмметрии	10
3.2.1	Распознавание опор на снимках	11
3.2.2	Оценка трёхмерных координат опор	12
3.3	Оценка положения объекта с использованием моделей	13
4	Предлагаемое решение	16
4.1	Описание подхода	16
4.2	Детектирование опор	16
4.2.1	Оценка координат объекта известной высоты	16
4.2.2	Нахождение опоры по треугольникам	17
4.2.3	Определение начального приближения координат опор	18
4.3	Позиционирование опор	20
4.3.1	Вычисление меры схожести	21
4.3.2	Перебор параметров позиционирования	22
4.3.3	Позиционирование опор по треугольникам	23
4.3.4	Независимое уточнение позиции и ориентации	26
4.4	Интерфейс	28
5	Результаты	29
5.1	Детектирование опор	29
5.2	Определение положения и ориентации опор	30

6	Заключение	35
7	Приложение	36
7.1	Модель стенопа	36
7.1.1	Внешние параметры камеры	37
7.1.2	Внутренние параметры камеры	37
	Литература	39

Реферат

с.41, рис. 16, табл. 2.

Данная работа посвящена задаче определения положения и ориентации в трёхмерном пространстве опор линий электропередач по их фотографиям с использованием алгоритмов компьютерного зрения. Проведено исследование существующих методов трёхмерной реконструкции коридора ЛЭП, а также решения задачи позиционирования модели объекта по фотографиям. Разработан прототип программного инструмента, решающий поставленную задачу, показавший хорошие результаты на тестовых данных.

Ключевые слова: *Позиционирование опор ЛЭП, реконструкция ЛЭП, фотограмметрия, компьютерное зрение, 3D реконструкция*

Список терминов сокращений

Термины

Выравнивание (множества фотографий) - процесс определения координат и ориентаций камер, с помощью которых сняты фотографии, относительно некоторой общей системы координат.

Рендеринг - (англ. rendering — «визуализация») — термин в компьютерной графике, обозначающий процесс получения изображения по модели с помощью компьютерной программы.

Сокращения

БПЛА - беспилотный летательный аппарат

ЛЭП - линия электропередач

CAD - Computer-Aided Design (рус. система автоматизированного проектирования)

LSD - Line Segment Detector - алгоритм компьютерного зрения, использующийся для нахождения отрезков прямых линий на изображениях

PMVS - Patch-Based Multiview Stereo - алгоритм компьютерного зрения, использующийся для построения плотного облака трёхмерных точек по множеству фотографий

RF - Random Forest - алгоритм машинного обучения, решающий задачи классификации, регрессии и кластеризации с использованием ансамбля решающих деревьев

SIFT - Scale Invariant Feature Transform - алгоритм компьютерного зрения, использующийся для нахождения локальных признаков на изображениях

SURF - Speeded Up Robust Features - алгоритм компьютерного зрения, использующийся для нахождения локальных признаков на изображениях

SVM - Support Vector Machine - алгоритм машинного обучения, решающий задачи регрессии и классификации

Введение

Цель и мотивация

Трёхмерное моделирование объектов промышленной инфраструктуры является важным инструментом визуализации и диагностики неполадок. При осуществлении строительных работ также часто существует необходимость учитывать то, где и как расположены те или иные объекты, и трёхмерная модель предоставляет удобный способ представления такой информации.

Моделирование линий электропередач особенно необходимо в связи с большим количеством угроз, связанных с передачей электроэнергии, которые необходимо анализировать. В их число входят обрывы проводов в связи с падениями деревьев, пробой изоляторов, замыкания проводов, вызванные растительностью, структурные изменения опор. Как следствие, естественным образом возникает задача построения трёхмерной модели коридора линии электропередач.

Существует несколько способов построения модели местности, принципиально отличающихся характером имеющихся данных. В приведённой работе в качестве входной информации о сцене полагается множество её фотографий, что связано с относительной простотой его получения.

Известные алгоритмы фотограмметрии, используемые для восстановления трёхмерной информации из фотографий и последующего построения моделей местности (например, SIFT, PMVS), часто возвращают результаты низкого качества для низкотекстурированных объектов (т.е. приблизительно одноцветных), содержащих большое количество ажурных и тонких структур. Примерами таких объектов являются ограды парков, выносные пожарные лестницы, опоры линий электропередач.

Целью данной работы является исследование уже существующих способов автоматизации определения координат и ориентаций опор с использованием аэрофотосъёмки линий электропередач с беспилотного летательного аппарата и разработка на их основе автоматизированного решения поставленной задачи, лишённого описанных недостатков. Методы, предлагаемые в данной диссертации, с небольшими изменениями могут быть применены для позиционирования многих других объектов помимо опор ЛЭП, обладающих сложной ажур-

ной структурой.

Постановка задачи

Необходимо разработать программное решение, автоматически определяющее параметры ориентирования и координаты опор ЛЭП и вставляющее их в модели, генерируемые инструментом Agisoft Photoscan.

Формально, входные данные для требуемого решения содержат следующие объекты:

- Множество фотографий, для которых известны позиции и ориентации камеры на момент съёмки.
- Известные дисторсии используемых камер.
- Карта высот для участка местности, содержащегося на фотографиях.
- Трёхмерная модель опоры, соответствующая опорам, присутствующим на фотографиях.
- Модель местности (с возможно отсутствующими опорами).

Связанные работы

Принципиально решения описанных и других близких задач можно разделить на два различных типа по характеру имеющихся данных. В первом случае, входные данные содержат измерения лазерного дальномера (лидара), во втором же используются исключительно фотографии местности.

Преимуществом решений, использующих данные с лидара совместно с фотографиями является то, что они содержат трёхмерную информацию о снятой сцене в явном виде. В подавляющем большинстве случаев эта информация представлена в виде трёхмерного облака точек, которое можно обрабатывать для получения конкретных результатов.

Существенным недостатком лидаров является их большой вес. Так, используемый в [2] прибор *Leica ALS70 LiDAR* имеет вес, согласно техническому описанию, более 40 килограммов, то есть примерно в 40 раз больше, чем фотоаппарат. Так как вес существенным образом влияет на количество времени, которое летательный аппарат может провести в воздухе, а также из-за того, что многие БПЛА просто не способны поднять такой груз, лидары не всегда целесообразно использовать.

Облако точек отснятой сцены, однако, может быть получено и средствами фотограмметрии, поэтому многие алгоритмы, описанные далее, теоретически могут быть применены в случае отсутствия лидара. Причина, по которой это не всегда возможно, состоит в необходимости наличия большого количества снимков на единицу площади снимаемой сцены, так как в противном случае низкотекстурированные ажурные объекты оказываются не представлены в облаке, либо представлены очень малым и разреженным количеством точек. Увы, и это не гарантирует качественного результата. Кроме того, большая плотность съёмки не всегда достижима и требует существенного увеличения временных затрат на полёты и последующую обработку изображений.

Далее мы кратко рассмотрим результаты, достигнутые к сегодняшнему дню в контексте реконструкции коридора линии электропередач. Также будут проанализированы методы решения задачи позиционирования произвольного объекта с применением САД модели, так как именно такой подход было решено использовать в данной работе.

Для более обширного ознакомления с тем, какие задачи, связанные с ЛЭП, решались за последние десятки лет с помощью различных методов дистанционного зондирования, за-

интересованному читателю рекомендуется ознакомиться со статьёй [1].

Методы, использующие лидар

Классификация облака точек

Вероятно, самой естественной задачей, решаемой с трёхмерным облаком точек, является его разбиение на классы по какому-либо признаку. Так, в случае облака точек для линий электропередач, множеством классов может быть *ландшафт, провода, опоры*. Такое разбиение позволяет оценивать различные характеристики объектов отснятой сцены, а также производить дальнейшую обработку точек в зависимости от того, какая метка им была присвоена.

Задача классификации, связанная с ЛЭП, решается например в [2], где в качестве решающего алгоритма используется Random Forest Classifier (RF). Для вычисления признаков, подаваемых на вход RF, пространство разбивается на воксели, для которых вычисляется набор характеристик (линейность, планарность, средняя высота и др.). В работе [3] используется классификатор JointBoost, после его применения результаты уточняются путём разбиения точек на два класса *фон, объект* с использованием алгоритма поиска разрезов в графе.

Реконструкция трёхмерной сцены

Большое количество исследований посвящено построению трёхмерной модели коридора линии электропередач. По сути, облако точек уже можно считать грубым примером модели, поэтому в данном случае подразумевается, что про некоторые объекты сцены становятся известны их геометрические характеристики: например, построение модели может включать определение характера провисания проводов, определение уравнения цепных линий для них, вычисление позиций вышек. Поэтому, обычно реконструкции сцены предшествует классификация облака точек.

Большинство работ ограничивается реконструкцией проводов, не затрагивая при этом моделирование опор. Эта проблема решается, например, в трудах [5], [4], [6].

Рассмотрим подробнее результаты, связанные с собственно реконструкцией опор ЛЭП. В своей работе [9] Gunho Sohn [и др.] предлагают новый алгоритм реконструкции линий электропередач, а также описывают возможный способ моделирования опор и зданий отснятой сцены.

Используя преобразование Хафа, а затем RANSAC, для каждого вокселя они строят набор планарных и линейных моделей внутри него. После этого определение, опоре или проводу принадлежит каждый линейный участок, осуществляется с помощью марковских цепей. Для уточнения локализации опор также проводится анализ двумерного изображения найденных на предыдущем шаге отрезков, после чего оставшиеся отрезки считаются принадлежащи-

ми модели вышки. Таким образом, результат данной работы, относящийся к моделированию опор, в основном носит визуальный характер, так как отсутствует возможность оценить какие-либо геометрические параметры получившейся модели.

Li, Chen и Hu в [7] предлагают определять ориентации вышек используя их вертикальную часть (т.н. тело, англ. body), которое приближается усечённой пирамидой. Верхнюю часть, к которой собственно крепятся провода, они предлагают выбирать из библиотеки трёхмерных моделей с использованием алгоритма SVM. Заявленная точность их метода позиционирования опор составляет 0.03 метра и 0.99°.

Во Guo [и др.] в [8] представляют метод реконструкции опор с использованием библиотеки параметризованных трёхмерных моделей вышек. После классификации облака с использованием JointBoost, для каждого кластера, содержащего опору, определяется его примерная ориентация по направлению соседних с кластером проводов. Далее, решается задача оптимизации параметров модели с использованием алгоритма MCMC в совокупности с имитацией отжига. При этом, сперва MCMC преимущественно осуществляет перебор моделей различных опор, после определённого числа итераций интенсивнее перебираются параметры лучшей найденной модели.

Погрешность представленного метода на тестовых данных составила 0.32 метра (максимальное зафиксированное расстояние от точек облака до реконструированной модели). Причиной этого, согласно статье, в том числе является пренебрежение возможным отклонением опоры от вертикального положения. Однако, это и позволяет автоматически предлагать гипотезы о ненормальном положении опор. Это предположение будет в дальнейшем использовано и в данной работе, так как оно существенно сужает пространство поиска параметров ориентации моделируемых объектов.

В статье отмечается, что это один из первых результатов, автоматизирующих процесс реконструкции опор.

Методы фотограмметрии

Решения, не использующие лидар, более интересны в контексте данной работы, поэтому при их анализе будут рассмотрены как разработанные ранее методы локализации опор, так и алгоритмы их дальнейшей реконструкции. Следует отметить, что задача распознавания образов на изображениях является в настоящее время хорошо изученной, и существует большое количество не связанных непосредственно с нахождением ЛЭП алгоритмов, которые уже себя хорошо зарекомендовали на самых разных категориях объектов для поиска.

Распознавание опор на снимках

Множество различных методов нахождения опор на изображениях было исследовано к текущему моменту. Несколько работ исследуют применение поиска углов для локализации опор на фотографиях, например, [11], [12].

Статья [13] посвящена нахождению на изображениях столбов ЛЭП с использованием поиска минимального разреза в графе, построенного на основании изображения со столбом. Для уменьшения вычислительной сложности, приблизительное определение региона интереса осуществляется с использованием набора эвристик, связанных с априорным знанием о структуре искомого столба (линейные размеры, форма). За счёт ограничения графа до региона интереса также повышается точность определения столба, в итоге алгоритм показал хорошие результаты на тестовых данных.

К сожалению, для вычисления весов в графе необходимо вручную задавать пороговое значение разницы между интенсивностями вершин (в роли них выступают пиксели), которые считаются близкими. Поэтому, скорее всего возникнут трудности с автоматизацией этого алгоритма. Другой проблемой, с которой увя должен столкнуться этот подход, является невозможность его применения в сложных сценах, когда много частей посторонних предметов (например, деревьев) попадают в выделенные области интереса. Судя по изображениям, представленным в статье, тестовые данные содержали столбы, снятые на фоне неба, что не соответствует обычному фону, на котором будут сняты опоры с БПЛА.

В [14] решается задача определения на видеозаписи областей, содержащих вышки ЛЭП, и удерживания её в поле зрения камеры (автоматическое сопровождение). Для этого на изображениях находятся прямые линии с помощью преобразования Хафа. Затем изображение оно разбивается на непересекающиеся регионы, и в каждом подсчитывается число найденных отрезков, при этом величина нормализуется делением на площадь региона. Предполагается, что чем больше прямых линий в регионе, тем более достоверно можно считать, что в нём содержится опора, при этом пороговое значение выбирается оператором вручную.

Другой работой, использующей детекторы прямых линий на фотографиях для нахождения опор, является [15]. Цель состоит в разработке алгоритма оценки расстояния до опоры по входному видеоряду.

Отрезки прямых на изображениях ищутся с использованием алгоритма LSD, после чего осуществляется фильтрация пикселей, принадлежащих линиям фона, а не столба. Основная гипотеза данной работы состоит в том, что опоры имеют низкую цветность, то есть неформально говоря, они обычно серого оттенка. Поэтому, пиксели в окрестностях найденных линий, не удовлетворяющие условиям низкой цветности, считаются не принадлежащими опоре. Из оставшихся пикселей окрестностей линий принадлежащими вышке считаются те, цветность которых близка к цветности моды в окрестности соответствующей линии.

Также в статье предлагается алгоритм вычисления дескрипторов отрезков для сопостав-

ления их между снимками. В контексте описываемой работы, это необходимо для дальнейшей оценки расстояния между камерой и вышкой, но в принципе могло бы быть использовано и для реконструкции трёхмерных линий опор, если бы выполнялись описанные в статье условия, а именно, предполагается движение камеры вдоль линий электропередач, с направлением камеры приблизительно совпадающим с направлением проводов. Съёмка же с БПЛА обычно осуществляется сверху, и в этом случае скорее всего дескрипторы для одной и той же линейной части опоры будут отличаться от изображения к изображению.

Sampedro [и др.] в своём труде [17] используют обучение с учителем двух многослойных перцептронов: для распознавания опор, а также для классификации найденных вышек по типу конструкции. В качестве признаков, подаваемых на вход нейронным сетям, выступают гистограммы направленных градиентов (HOG).

Интересна работа [18], где автор рассматривает SVM для обнаружения вышек, в качестве входного вектора признаков для которого выступает массив вычисленных в интересующей окрестности ORB/BRIEF дескрипторов. В качестве гипотез, где искать опоры (в каких окрестностях применять SVM), предлагается использовать особые точки с помощью детекторов (например, SIFT, поиск углов Харриса, SURF, и др.), и концы отрезков прямых, найденных на изображениях. Плюсом выбранного ORB дескриптора по сравнению с использованным в предыдущей работе HOG являются его меньшие вычислительные затраты и меньшее число зафиксированных ложных положительных срабатываний.

Оценка трёхмерных координат опор

Определению углов отклонений опоры от вертикального положения посвящена работа [10]. В ней предлагается определять смещения опоры за счёт смещения особых точек, найденных алгоритмом SIFT между изображением несмещённой опоры и изображением, отклонение на котором требуется оценить. Сложностью, с которой сталкиваются авторы, является необходимость фильтрации точек, которые оказываются не принадлежащими фактически изображению опоры (например, особые точки на изображениях объектов окружающего ландшафта).

Автоматическое нахождение опоры в решении отсутствует, но, как ранее упоминалось, это не столь существенно. Более важно то, что, как пишут авторы, пользователю необходимо вручную выбрать часть изображения опоры, находящуюся на относительно простом фоне (в идеале, это должно быть нечто наподобие неба), дабы уменьшить количество лишних особых точек. Это требование ставит под сомнение возможность простого перехода к автоматизации описываемого метода. К тому же, в случае осуществления съёмки с БПЛА, монотонность фона - это фактически невыполнимое условие. Данное решение, согласно статье, разрабатывалось для систем видеонаблюдения за опорами, поэтому вполне понятно, почему такие ограничения были сочтены допустимыми.

В [16] авторы используют БПЛА с установленной на нём парой камер и ставят перед

собой задачу выравнивания получившихся снимков с последующей реконструкцией сцены, включая провода и опоры. Для выравнивания фотографий используется сопоставление особых точек, найденных SIFT, для более точной привязки фотографий вручную размечаются контрольные точки на фотографиях. При этом, модель строится опять же неавтоматизированно по размеченным контрольным точкам.

Различные дескрипторы (SIFT, BRISK, FREAK и ORB) рассмотрены в [18] для сопоставления точек на различных изображениях и дальнейшей реконструкции сцены, содержащей опоры ЛЭП. Наибольшее количество соответствий по результатам исследований находят с использованием SIFT, при этом он является наиболее сложным с точки зрения времени его вычисления. Несмотря на хорошие результаты, описанные в статье, на имеющихся в нашем распоряжении тестовых данных не удалось получить удовлетворительную модель с использованием SIFT.

Стоит сказать о библиотеке Line3D, описанной в статьях [19], [20], [21] и диссертации [22]. В ней используется эпиполярная геометрия для построения множества гипотетических трёхмерных отрезков, после чего гипотезы фильтруются по различным критериям. Так как опоры преимущественно состоят из прямых балок, они теоретически должны быть хорошо представимы отрезками. В приведённых работах в том числе использовались изображения вышек ЛЭП, которые успешно реконструировались алгоритмом. Отмечается существенно более высокая скорость построения модели, нежели в случае PMVS, а также намного более компактное представление модели (отрезки прямых содержат больше семантической информации, поэтому их необходимо меньше для адекватного представления сцены).

Тем не менее, результаты, получаемые с помощью Line3D на входных данных, содержащих малое количество фотографий опоры (13-20 фотографий), на которых опора в основном видна сверху, содержат очень малое число восстановленных линий. Поэтому, применение этой библиотеки также не позволяет решить поставленную задачу.

В [23] предлагается осуществлять кластеризацию линий, найденных алгоритмом, используемым в Line3D, для построения модели опоры. При этом параметры кластеризации определяются пользователем, также существуют проблемы с отсутствующими и некорректно восстановленными частями опор на получающейся модели. Автор отмечает, что это связано с неравномерным распределением найденных на фотографиях отрезков.

Оценка положения объекта с использованием моделей

Задаче позиционирования произвольного объекта по фотографиям с использованием CAD модели также посвящено большое количество работ. Мы рассмотрим лишь некоторые из них, наиболее релевантные в контексте нашей задачи.

Идея применения обобщённого преобразования Хафа отрезков, обнаруженных на фо-

тографии и отрезков отрисованной модели, для решения описанной проблемы, предложена в [27]. Авторы рассматривают задачу определения позиции объекта как задачу поиска оптимальной ориентации камеры на единичной сфере и двумерного смещения модели в экваториальной плоскости сферы. Сам поиск предлагается осуществлять итеративно, изначально разбив сферу на участки, спроецировав на неё грани икосаэдра, и в дальнейшем подразбивая те из граней, выбор которых доставляет пиковые значения аккумулятора преобразования Хафа.

Опора состоит из большого количества прямолинейных балок, вместе формирующих в том числе много треугольников. В [31] описывается способ определения положения объекта по изображению какой-либо из его треугольных частей. Для выбора наилучшей трансформации используется голосование методом обобщённого преобразования Хафа в трёхмерном пространстве смещений (вращения не учитываются).

Авторы [32] предлагают алгоритм POS-IT для итеративного улучшения оценки положения объекта с использованием сопоставления особых точек на модели и изображении. Они используют ортографическую проекцию вместо перспективной, добавляя оценку смещения к координатам точек изображения, и для n сопоставленных точек составляют систему из $2(n - 1)$ линейных уравнений с шестью неизвестными. Для её решения используется метод наименьших квадратов, в результате получается очередная оценка положения объекта и вычисляются новые оценки смещения.

Для выбора правильных сопоставлений предлагается искать минимум меры деформации, определение которой даётся в той же статье. По смыслу, она тем больше, чем менее ортогональными оказываются единичные направляющие вектора глобальной системы координат, преобразованные с помощью найденной матрицы трансформации в модельную систему координат, а также чем больше разница их длин после преобразования. Такой подход, как пишут сами авторы, приемлемо работает в случае малой неопределённости в сопоставлении особых точек.

В [28] вместо исчерпывающего поиска возможных значений параметров ориентации твердого тела предлагается использовать алгоритм, основанный на сопоставлении групп из трёх характерных точек модели и изображения. В статье показано, что имея m характерных точек модели и n точек изображения, можно рандомизированным алгоритмом за $O(mn^3)$ с наперёд заданной вероятностью ошибки определить объект на изображении и его координаты, либо сказать, что объект отсутствует. В [29] описывается обобщение описанного подхода под названием RUDR.

В [33] одновременно предлагается метод распознавания и определения позы объектов на сложных сценах, содержащих до 400 различных предметов одновременно. Для этого сперва выделяются особые точки на моделях и изображениях, при этом в качестве дескрипторов используются SIFT или SURF. Далее между дескрипторами точек объектов и изображений

строится соответствие, после чего применяется алгоритм ISE. Суть его состоит в итеративном выполнении двух шагов, применяемых независимо к каждому изображению:

- Кластеризации особых точек изображения по принципу принадлежности (гипотетической) определённому объекту. В случае первой итерации, это делается, исходя из плотности, с которой распределены особые точки, соответствующие тому или иному объекту. На последующих итерациях кластеры инициализируются, исходя из оцененных ранее позиций объектов.
- Построение множества гипотез о положении объекта для каждого кластера с применением алгоритма RANSAC и LM (метод, предложенный Левенбергом и Маркардом), с последующим объединением близких гипотез.

При этом, алгоритм завершается, когда перестаёт изменяться разбиение особых точек изображения на кластеры.

Большая часть описанных решений взаимодействует с объектами, для которых можно эффективно вычислить качественные текстурные дескрипторы, и поэтому не могут быть непосредственно применены к рассматриваемой в данной работе задаче. Поэтому, мы решили использовать идеи, предложенные в [31], а также подход, основанный на обобщённом преобразовании Хафа, близкий к [27].

Предлагаемое решение

Описание подхода

Описанные в предыдущей главе методы фотограмметрической реконструкции опор имеют недостатки, которые не позволяют их эффективно применять для решения поставленной задачи. В данной работе за основу взяты идеи, описанные в [27] и [31].

Первый этап решения задачи состоит в нахождении снимков, содержащих опоры, и начальной оценке положения опор в пространстве. В будущем планируется решать эту задачу с применением нейронных сетей, но в данной работе, в силу небольшого количества тестовых данных, был реализован прототип детектора на основе поиска треугольников на изображениях.

Мы предполагаем, что опора в контексте поставленной задачи может быть представлена как абсолютно твёрдое тело, таким образом, для её описания в пространстве достаточно задать шесть скалярных величин (например, три декартовых координаты x, y, z и три угла $\varphi_x, \varphi_y, \varphi_z$ - крен, тангаж и курс). Поэтому, второй этап состоит в подборе этих шести параметров для каждой выявленной на предыдущем этапе группы областей. Чтобы оценить, насколько хорошо параметры описывают реальное положение вышки, мы осуществляем рендеринг модели, выбирая для этого положения камеры, соответствующие тем, с которых сняты фотографии и сравниваем результаты с исходными изображениями.

Для удобства дальнейшего расширения и возможности в будущем выбора используемого метода определения опор на снимках, решение реализовано в виде двух независимых модулей. Первый модуль служит для нахождения опор на фотографиях, и приблизительно вычисляет их позиции. Второй модуль решает задачу точного позиционирования вышек.

Детектирование опор

Оценка координат объекта известной высоты

Рассмотрим, как можно оценить положение объекта, зная, где на фотографиях расположена соответствующая ему точка и его примерную высоту.

Обозначим \vec{x}_c позицию камеры, \vec{x}_p искомую оценку позиции объекта, h_p его высоту. Пусть n_p - направление из камеры на точку, соответствующую объекту, а $height(\vec{p})$ - значение высоты, соответствующей точке (p_x, p_y) , записанное в карте высот.

Пусть α - положительная переменная. Тогда координаты любой точки на луче, идущем из камеры в точку изображения, описываются уравнением $\alpha\vec{n}_p + \vec{x}_c$. Нас интересует точка - вершина опоры. Поэтому, для неё мы можем записать уравнение:

$$(\alpha\vec{n}_p + \vec{x}_c)_z = height(\alpha\vec{n}_p + \vec{x}_c) + h_p$$

Это уравнение в общем случае может иметь много решений, при этом по крайней мере одно существует в случае, если объект снят на фоне земли (неформально - если он снят сверху). Количество решений зависит от характера ландшафта и направления \vec{n}_p . Например, если существует несколько пиков и ущелий, чередующихся между собой, и луч $\alpha\vec{n}_p + \vec{x}_c$ проходит над ними, решений может быть несколько. Тем не менее, для относительно равнинных местностей легко убедиться в единственности решения. Докажем это.

Рассмотрим функцию $f(\alpha) = height(\alpha\vec{n}_p + \vec{x}_c) + h_p - \alpha n_{pz} - x_{cz}$. Продифференцировав её, получим $\frac{df}{d\alpha} = \nabla height(\alpha\vec{n}_p + \vec{x}_c) \cdot \vec{n}_p - n_{pz} < 0$ если проекция градиента поверхности земли в любой точке на ось z не превосходит n_{pz} . Это означает, что при описанных условиях $f(\alpha)$ монотонно убывает с ростом аргумента. В совокупности с тем, что решение $f(\alpha) = 0$ существует, это означает, что решение единственно.

Это позволяет искать решение уравнения численно, например методом бинарного поиска значения параметра α .

Нахождение опоры по треугольникам

Используем следующее наблюдение: опоры состоят не просто из большого количества прямолинейных балок, но эти балки обычно, соединяясь между собой, образуют большое количество треугольников. Поэтому, для нахождения опор кажется логичным искать на изображениях именно треугольники.

Отрезки, находимые алгоритмом LSD, обычно короче, нежели их прообразы, поэтому чтобы повысить число находимых на изображении треугольников-частей опоры возникает необходимость продлевать найденные отрезки. Эмпирическим путём было выявлено, что продление отрезков в 1.5 раза даёт приемлемые результаты, при этом не порождая слишком много реально несуществующих треугольников.

Пересечения на изображении можно эффективно находить, например, используя классический алгоритм заметающей прямой, при этом можно динамически поддерживать множество обнаруженных треугольников.

После того, как все треугольники обнаружены, мы фильтруем их, удаляя треугольники с очень большими, очень маленькими сторонами, а также очень маленькими углами.

Сторона считается большой, если более, чем в два раза превосходит максимальную сторону треугольника модели. Аналогично, сторона считается слишком маленькой, если она меньше, чем половина самой маленькой стороны треугольника модели. Для оценки реальной длины стороны (в метрах), мы используем подход, описанный в предыдущем пункте: вычисляем α , считая, что в среднем треугольник располагается на высоте, равной половине высоты опоры, а затем пользуемся отношением подобия.

Затем мы кластеризуем треугольники, используя в качестве метрики евклидово расстояние между их центрами. Если не менее τ_c фотографий содержат кластеры, находящиеся приблизительно в одном месте, и в каждом таком кластере более τ_p треугольников, то позиция, задаваемая этими кластерами, считается содержащей опору, и далее вычисляется область интереса, как и в предыдущем пункте. В нашей работе мы приняли $\tau_c = 3$ и $\tau_p = 2$.



Рис. 4.1: Треугольники, обнаруженные алгоритмом PylonByTriangles

Определение начального приближения координат опор

Будем считать, что центр масс всех найденных треугольников приблизительно соответствует изображению вершины опоры, и воспользуемся алгоритмом позиционирования, описанным ранее. Погрешность в 5-10 метров в данном случае приемлема.

Найдя таким образом приблизительное положение опоры \vec{x}_p , мы можем определить квадратную область на снимке, с большой долей вероятности всецело содержащую изображение опоры. Запишем уравнение подобия:

$$\frac{\Delta}{f} = \frac{d_p}{\alpha}$$

Где Δ - длина стороны искомого квадрата в пикселях, f - фокусное расстояние камеры, выраженное в пикселях, d_p - диаметр опоры в метрах, а параметр α берётся определяет длину луча, соединяющего точку положения камеры и приблизительную точку опоры.



Рис. 4.2: Пример области с опорой, выделенной алгоритмом

Это оценка сверху на размер описанного квадрата, так как в такой квадрат вообще говоря поместится по меньшей мере не пересекаясь два изображения опоры. Это намеренное завышение, так как истинное положение опоры может отличаться от вычисленного на достаточно большие расстояния (на тестовых данных погрешность достигала 5 метров). На рис. 4.2 можно увидеть результирующую область интереса (жёлтый квадрат).

В общем случае описанный алгоритм мог не найти все изображения опоры (на некоторых снимках, например, могли не обнаружиться треугольники). Поэтому, после обработки всех фотографий, мы осуществляем проецирование области интереса на фотографии, снятые

с позиций, близких к той позиции, с которой снята фотография с обнаруженными треугольниками.

Позиционирование опор

Второй, и главной задачей данной работы была разработка метода автоматического позиционирования опор. Для её решения, мы предлагаем подход, основанный на сравнении отрендеренных моделей опоры с имеющимися изображениями.

Введём в рассмотрение две сущности. Во-первых, алгоритм \mathcal{A}_s , оценивающий численно меру схожести изображения фотографии и отрисованного изображения. Во-вторых, алгоритм \mathcal{A}_m , который осуществляет перебор по какому-либо принципу параметров позиционирования опоры и возвращает параметры, порождающие отрендеренную картину, наиболее похожую на фотографию. Формально, \mathcal{A}_m применяется для решения задачи оптимизации - поиска максимума функции схожести, которую вычисляет \mathcal{A}_s .

В компьютерной графике стандартным способом представления положения объекта является модельная матрица - квадратная матрица размером 4×4 , которую можно представить в виде произведения трёх матриц, определяющих соответственно смещение (T), поворот (R) и масштаб (S) модели: $M = T \cdot R \cdot S$. В контексте нашей работы, масштаб модели опоры фиксирован и не подлежит определению, поэтому можно считать S единичной матрицей и далее не рассматривать. Другие две матрицы имеют вид

$$R = R_z \cdot R_y \cdot R_x \quad T = \begin{pmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$R_x = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\varphi_x) & -\sin(\varphi_x) & 0 \\ r_{31} & \sin(\varphi_x) & \cos(\varphi_x) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad R_y = \begin{pmatrix} \cos(\varphi_y) & 0 & \sin(\varphi_y) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\varphi_y) & 0 & \cos(\varphi_y) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$R_z = \begin{pmatrix} \cos(\varphi_z) & -\sin(\varphi_z) & 0 & 0 \\ \sin(\varphi_z) & \cos(\varphi_z) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

При этом положение модели относительно глобальной системы координат определяется вектором $\vec{t} = (t_x, t_y, t_z)$, а углы её поворота - матрицей R .

Преобразование проекции, описывающее отображение точки модели в точку изображения может быть записано с использованием матрицы вида камеры V и матрицы калибровки

C . О них можно прочесть в приложении, здесь лишь отметим, что итоговая матрица трансформации из модельных координат в координаты пикселей имеет вид $T = C \cdot V \cdot M$.

Таким образом, задачей, решаемой \mathcal{A}_m будет нахождение оптимальной матрицы M , а \mathcal{A}_s будет собственно оценивать оптимальность.

Вычисление меры схожести

В качестве информации, используемой для сравнения изображения с отрисованной моделью будем использовать извлечённые с помощью LSD отрезки. Обозначим I_p изображение, снятое камерой, I_r отрендеренное изображение. Размеры I_r и I_p совпадают. Для решения поставленной задачи было выбрано использовать обобщённое преобразование Хафа, применённое к отрезкам изображений, так как оно устойчиво к шумам и предлагается к использованию в ряде описанных работ.

Рассмотрим I_p . Разобьём его регулярной сеткой на клетки размера $k \times k$ пикселей. Выбор размера клетки k определяется точностью, используемой при переборе координат. Зафиксируем число D направлений отрезков на изображении, которые мы хотим различать. В данной реализации было решено взять $D = 180$. Мы теперь определим дескриптор клетки как массив размером D , описывающий отрезки, пересекающие клетку в каждом направлении.

Опишем, как вычислить такой дескриптор. Рассмотрим произвольную клетку изображения. Определим для неё бинарный массив $a[0..D - 1]$, при этом $a[i] = 1$ если на изображении существует отрезок с направлением i , пересекающий эту клетку. Вычислим массив-дескриптор b как свёртку a с одномерным гауссовым ядром, то есть

$$b[i] = \max_{j=i-s..i+s} \text{Round}(\kappa \cdot e^{-\frac{s^2-(j-i)^2}{2\sigma^2}}) \cdot a[j \bmod N]$$

Параметр σ задаёт меру гладкости гауссовского ядра и был выбран равным 1. Параметр s определяет как много соседних направлений дают вклад в вычисление максимума для текущего направления, и в текущей реализации выбрано $s = 2$.

Теперь, определим меру схожести I_p и I_r следующей формулой

$$S(I_p, I_r) = \sum_{i=1}^n \vec{b}_i^p \cdot \vec{a}_i^r$$

где суммирование ведётся по перенумерованным клеткам, на которые разбиты изображения. Использование свёртки с гауссовским ядром позволяет сделать функцию схожести S локально более гладкой.

Для вычисления дескрипторов a и b , а также функции S используется алгоритм Брезенхэма. На вход алгоритму \mathcal{A}_s подаётся массив матриц трансформаций $M[1..n]$, а также массив Seg_p отрезков, обнаруженных на фотографии, и массив Seg_m отрезков трёхмерной модели.

Алгоритм возвращает массив $S[1..n]$ мер схожести между I_p и отрендеренными с данными матрицами изображениями модели.

В приведённом ниже листинге предполагается, что функция *Bresenham* является функцией высшего порядка и принимает на вход отрезок, над которым выполняется обычный алгоритм Брезенхэма, при этом для каждой клетки сетки, полученной алгоритмом Брезенхэма, выполняется переданная вторым аргументом функция.

Algorithm 1: \mathcal{A}_s

Input : $M[1..N]$ - массив матриц трансформаций из модельной системы координат в пиксели; Seg_p - массив найденных на фото I_p отрезков; Seg_m - массив отрезков трёхмерной модели

Output: $S[1..N]$ - массив мер схожести между I_p и отрендеренными с использованием $M[1..N]$ изображениями

```

1 Инициализировать массив дескрипторов  $a$  нулями
2 for  $s$  in  $Seg_p$  do
3   | Bresenham( $s$ , ( $k$ ,  $d$ )  $\rightarrow a[k][d] := 1$ )
4 end
5 for  $k$  in  $[1..n]$  do
6   | for  $i$  in  $[1..D]$  do
7     |  $b[k][i] = \sum_{j=i-s}^{i+s} Round(e^{\frac{s^2-(j-i)^2}{2\sigma^2}}) \cdot a[k][j \bmod N]$ 
8     | end
9 end
10 for  $i$  in  $[1..N]$  do
11   |  $S[i] := 0$ 
12   | for  $s_m$  in  $Seg_m$  do
13     |  $s_{projected} := normalize(M[i] \cdot s_m)$ 
14     | Bresenham( $s_{projected}$ , ( $k$ ,  $d$ )  $\rightarrow S[i] := S[i] + b[k][d]$ )
15   | end
16 end
17 return  $S$ 

```

Практически, вычисления в строках 10-16 производятся параллельно, так как обычно N велико. Для этого мы используем технологию Nvidia CUDA, при этом каждый поток обрабатывает единственный модельный отрезок при единственной матрице трансформации.

Перебор параметров позиционирования

Задачей, которую должен решить \mathcal{A}_m , является поиск параметров положения опоры (x, y, z) в окрестности определённого ранее начального приближения координат опоры, а также опреде-

ление её ориентации $(\varphi_x, \varphi_y, \varphi_z)$. Так как вычисленное приближение позиции является грубым (погрешность может достигать 5-10 метров), неэффективным будет осуществлять полный перебор всех шести неизвестных параметров с необходимым для получения желаемой точности шагом (такой шаг должен составлять менее 0.3 метров и менее 1 градуса в случае курсового угла опоры, менее 0.1 градуса в случае углов крена и тангажа). Поэтому, мы предлагаем алгоритм, состоящий из трёх этапов. Каждый этап выполняется над всеми изображениями опоры независимо, после чего результаты объединяются с применением аппарата математической статистики.

Позиционирование опор по треугольникам

Первый этап использует идею из [31]. Заметим, что искомые опоры конструктивно состоят из балок, образующих треугольники, и потому мы осуществляем поиск соответствий между треугольниками на фотографии и треугольниками модели, после чего вычисляем модельные матрицы, исходя из этих соответствий.

Задача определения трансформации треугольника по его изображению в литературе называется РЗР (англ. Perspective-3-Point) проблемой. Способ её решения можно найти, например, в [30]. Он состоит в нахождении неизвестных OA', OB', OC' в системе трёх уравнений 4.1 второй степени. В [30] доказывается, что РЗР имеет не более 4 физически осмысленных решений, искать которые можно либо численно, либо решив несколько уравнений 4 степени. В большинстве случаев, РЗР имеет не более двух вещественных осмысленных решения.

$$\begin{cases} A'B'^2 = OA'^2 + OB'^2 - 2OA'OB' \cos(\theta_{ab}) \\ A'C'^2 = OA'^2 + OC'^2 - 2OA'OC' \cos(\theta_{ac}) \\ B'C'^2 = OB'^2 + OC'^2 - 2OB'OC' \cos(\theta_{bc}) \end{cases} \quad (4.1)$$

Таким образом, первый этап формализуется алгоритмом 2 (листинг на следующей странице).

Для уменьшения количества проверяемых гипотез в строке 27 используется знание о приблизительном положении опоры. Это позволяет не оценивать гипотезы, заведомо неверные. В текущей реализации таковыми считаются гипотезы, соответствующие смещению опоры относительно предполагаемого положения более, чем на 15 метров, либо определяющие её углы тангажа и крена (отклонения от вертикальной оси) большими, чем 1° . Треугольники ищутся на изображении, как и в случае детектирования опоры.

Если в результате m_i оказалось пустым множеством, то мы осуществляем исчерпывающий поиск в окрестности положения, предсказанного на этапе детектирования, со смещениями не более 3 метров вдоль каждой из осей.

После осуществления этой процедуры над всеми областями интереса, содержащими од-

Algorithm 2: MatchByTriangles

Input : $I[1..r]$ - массив изображений, каждое $I[j]$ представлено выделенными на нём отрезками; T_m - массив треугольников модели; $V[1..r]$, $C[1..r]$ - матрицы вида и калибровки камер; Seg_m - массив отрезков модели; \vec{t} - приблизительное положение опоры

Output: m - модельная матрица, найденная алгоритмом; MAD_{pos} , MAD_{angle} - трехмерные вектора медианных абсолютных отклонений положения и углов

```
18  $M := \emptyset$ 
19 for  $i$  in  $[1..r]$  do
20    $T_p := \text{FindTriangles2D}(I[i])$ 
21    $m_i := \emptyset$ 
22   for  $t_p$  in  $T_p$  do
23     for  $t_m$  in  $T_m$  do
24        $m_i := m_i \cup \text{SolveP3P}(t_p, t_m)$ 
25     end
26   end
27    $\text{FilterUnprobableHypotheses}(m_i, \vec{t})$ 
28    $s_i := \mathcal{A}_s(m_i, I[i], Seg_m)$ 
29   if  $\max(s_i) > \tau$  then
30      $j := \arg \max_j s_i[j]$ 
31      $\text{append}(M, (m_i[j], s_i[j]))$ 
32   end
33   else
34     Выполнить грубый исчерпывающий поиск в окрестности изначальной гипотезы
35   end
36 end
37  $m, MAD_{pos}, MAD_{angle} := \text{MergeResults}(M, V)$ 
38 return  $m, MAD_{pos}, MAD_{angle}$ 
```

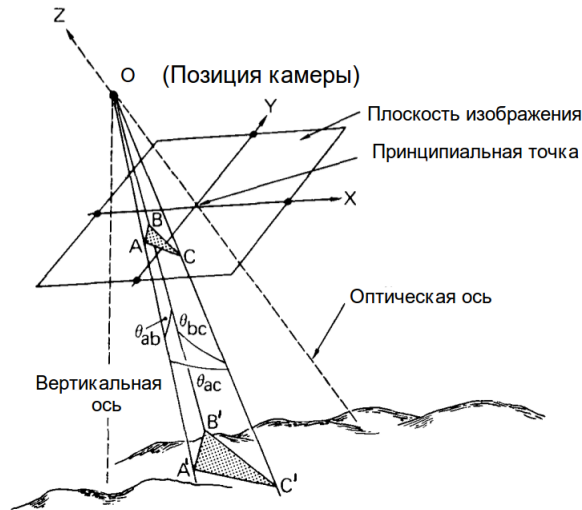


Рис. 4.3: Пояснение к системе 4.1, изображение взято из [30].

ну и ту же опору, мы объединяем результаты. Для этого мы вычисляем взвешенные медианы параметров x, y, z независимо друг от друга, отбрасываем значения параметров, абсолютные отклонения которых от медиан превышают соответствующие абсолютные медианные отклонения, после чего вычисляем взвешенное среднее каждого параметра как итоговую оценку параметра.

В качестве весов используются значения меры похожести, умноженные на синус угла ψ между направлением от камеры на опору и направлением соответствующей оси. Необходимость использования такого масштабирующего множителя для весов состоит в зависимости точности определения конкретной координаты от угла между её направляющим вектором и лучом, соединяющим камеру с опорой. Когда этот угол мал, смещения объекта вдоль оси практически не изменяют его изображения, что связано с малостью рассматриваемых смещений по сравнению с расстоянием до камеры (типичное расстояние до камеры составляет 100-150 метров, тогда как рассматриваемые смещения имеют порядки метров).

Процедура для усреднения углов несколько отлична, так как взятие взвешенного среднего по выборке едва ли даст хороший результат. Связано это прежде всего с тем фактом, что опора обладает сильно выраженной симметрией. Если говорить об основании опоры, то обычно поворот на 90° переводит его в само себя. Верхняя часть опоры часто обладает более слабой симметрией поворота на 180° , хотя и не всегда (пример не обладающей такой симметрией опоры является опора У-110). Поэтому, большая дисперсия углов курса может наблюдаться после применения алгоритма 2, и, как следствие, вместо выборочного среднего имеет смысл брать взвешенную медиану курсового угла. При этом, в силу описанной симметрии опоры теоретически возможен выбор существенно неверного направления, и эта проблема решается на следующем этапе.

Независимое уточнение позиции и ориентации

Второй этап состоит в том, чтобы уточнить решение, найденное на первом этапе, выполнив несколько итераций последовательного уточнения углов и положения.

Используемые далее *MatchRotation* и *MatchTranslation* для каждого входного изображения осуществляют перебор параметров ориентации и позиции соответственно с шагами, определяемыми медианными абсолютными величинами соответствующих величин.

Конкретно, *MatchRotation* совершает полный перебор параметров $(\varphi_x, \varphi_y, \varphi_z)$ в границах $[[\varphi_x - MAD_{\varphi_x}, \varphi_x + MAD_{\varphi_x}], [\varphi_y - MAD_{\varphi_y}, \varphi_y + MAD_{\varphi_y}], [\varphi_z - MAD_{\varphi_z}, \varphi_z + MAD_{\varphi_z}]]$ с шагом $1^\circ, 0.1^\circ, 0.1^\circ$ соответственно. Для каждого входного изображения определяется набор параметров, сообщающий максимум мере похожести S .

MatchTranslation совершает полный перебор параметров (x, y, z) в границах $[[x - MAD_x, x + MAD_x], [y - MAD_y, y + MAD_y], [z - MAD_z, z + MAD_z]]$ с шагом, определяемым желаемым максимальным числом итераций, границами поиска координат и минимальной ожидаемой погрешностью (в текущем решении используется не более 30 000 итераций, а границей погрешности считается 0.1 метра). $MAD_x..MAD_z$ - медианные абсолютные отклонения по осям, x, y, z - текущие оценки координат опоры. Для каждого входного изображения определяется набор параметров, сообщающий максимум мере похожести S .

Algorithm 3: MatchSeparate

Input : m - модельная матрица, найденная алгоритмом; MAD_{pos}, MAD_{angle} - трехмерные вектора медианных абсолютных отклонений положения и углов; $I[1..r]$ - массив изображений, каждое $I[j]$ представлено выделенными на нём отрезками; $V[1..r], C[1..r]$ - матрицы вида и калибровки камер; Seg_m - массив отрезков модели

Output: модифицированные $m, MAD_{pos}, MAD_{angle}$

```

39 Инициализация
40 while iterations < maxIterations AND (|MAD_pos| > 0.3 OR |MAD_angle| > 1) do
41     M := MatchRotation(m, MAD_angle, I, V, C, Seg_m)
42     m, MAD_angle := MergeAngleResults(M)
43     M := MatchTranslation(m, MAD_pos, I, V, C, Seg_m)
44     m, MAD_pos := MergePositionResults(M, V)
45     if iterations = 0 then
46         | Выбрать лучший из поворотов вокруг z среди {0°, 90°, 180°, 270°}
47     end
48     iterations := iterations + 1
49 end
50 return m, MAD_pos, MAD_angle

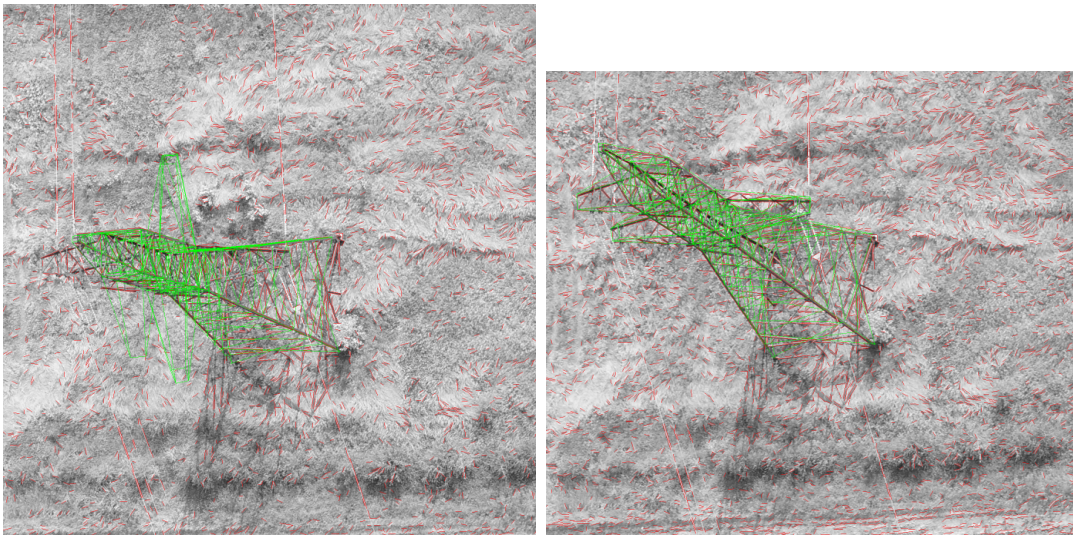
```

Разделение перебора параметров позволяет существенно снизить количество рассматри-

ваемых вариантов. При этом, на втором этапе погрешность позиций уже существенно ниже, а потому такое решение имеет смысл - углы спроецированных отрезков практически не изменяются при смещении опоры на несколько метров. Использование сглаженного дескриптора b ведёт к тому, что небольшие погрешности в ориентации не вызывают скачков меры схожести.

В строке 46 алгоритма выбирается поворот, максимизирующий суммарную меру схожести для всех фотографий.

Выбор границ поиска в случае углов продиктован тем, что нормальные отклонения вышки от вертикального положения согласно [34], составляет менее 0.5° относительно вертикального положения.



(a) Пример ситуации, когда после первого этапа выбрана неверная ориентация (b) Результат после выполнения строки 46 алгоритма 3

Третий этап похож на второй, и уточняет решение, выполняя одновременный перебор всех шести параметров в пределах их текущих медианных абсолютных отклонений.

Интерфейс

На данный момент, реализованный инструмент предполагается использовать совместно с ПО *Agisoft Photoscan*. В качестве формата входных данных о сцене (матрицы трансформаций выровненных фотографий, карта высот, модель местности) используется представление, применяемое в проектах *Photoscan*. Таким образом, от пользователя требуется передать на вход программе директорию, содержащую проект, а также файл, содержащий модель опоры.

Для удобства отладки имеется возможность включить визуализацию отдельных шагов, а также сохранить получившиеся изображения.

Результаты

В качестве тестовых данных были рассмотрены снимки девяти опор: семи ВЛ-220 кВ и двух У-110. ВЛ-220 обладает симметрией поворота на 180° вокруг вертикальной оси, У-110 имеет основу, переходящую в себя при повороте на 90° градусов, но асимметричную верхнюю часть. Разрешения всех изображений составляли 6000×4000 , снимки производились с высоты около 120 метров, среднее число снимков на одну опору типа ВЛ-220 составляет 12 фотографий, с опорами У-110 имелось 35 фотографий. Опоры сняты вдоль нескольких линий электропередач, проходящих по загородной местности.

Эксперименты проводились на ноутбуке с процессором core i5 с пиковой тактовой частотой 3200 МГц, 16 Gb оперативной памяти и видеокартой Nvidia GeForce 840M.



(a) Опора ВЛ-220



(b) Опора У-110

Рис. 5.1: Пример тестовых данных.

Детектирование опор

Разработанный в рамках данной работы детектор опор приемлемо работает со снимками сельской местности. В табл. 5.1 представлена информация о результатах нахождения опор на тестовых данных. Ложноположительные результаты в основном связаны со слабыми ограничениями на количество треугольников, которое тем не менее не получается усилить при условии малого количества фотографий на одну опору. Скорее всего, в случае городской среды, будет наблюдаться большее количество ложноположительных результатов.

Ложноположительные срабатывания	Ложноотрицательные срабатывания	Положительные срабатывания
4	0	9

Таблица 5.1: Результаты использования детектора

Среднее время обработки одной фотографии 6000 x 4000 составляет 2.4 секунды, большую часть которого занимает применение алгоритма LSD. Тем не менее, так как алгоритм не составляет труда параллелизовать, можно добиться более быстрой обработки (например, в текущей версии применение OpenMP на 4-х ядерном процессоре позволило обработать 567 фотографий за 9 минут, то есть в среднем за 0.95 секунды на фотографию).



Рис. 5.2: Пример ложноположительного детектирования.

Определение положения и ориентации опор

На рис. 5.3 можно увидеть пример зависимости меры похожести от того, после какого этапа она измерена ($p3p$ - после первого этапа, $A1$, $A2$ - подбор $(\varphi_x, \varphi_y, \varphi_z)$, $T1$, $T2$ - подбор x, y, z , AT

- одновременный перебор углов и положения). Как видно, наибольший скачок меры происходит после первого подбора положения, при этом подбор углов в данном случае практически не осуществляется. Это связано с тем, что даже при неточной позиции удаётся достаточно точно определить углы из-за наличия большого количества практически параллельных балок в конструкции опоры. Существенный разброс значений меры между фотографиями связан с различным количеством найденных на снимках отрезков опоры (например, какие-то снимки содержали лишь её часть).

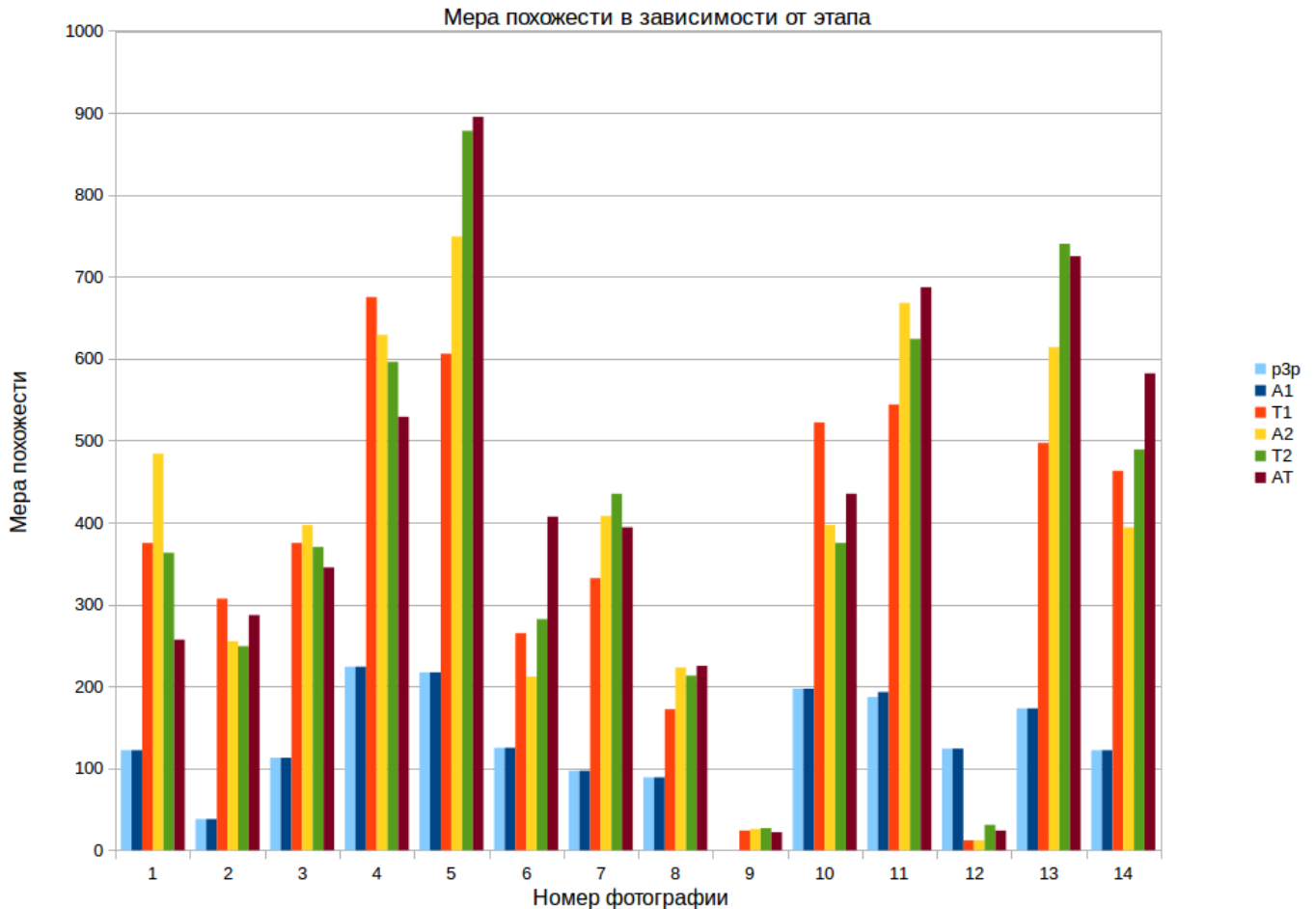
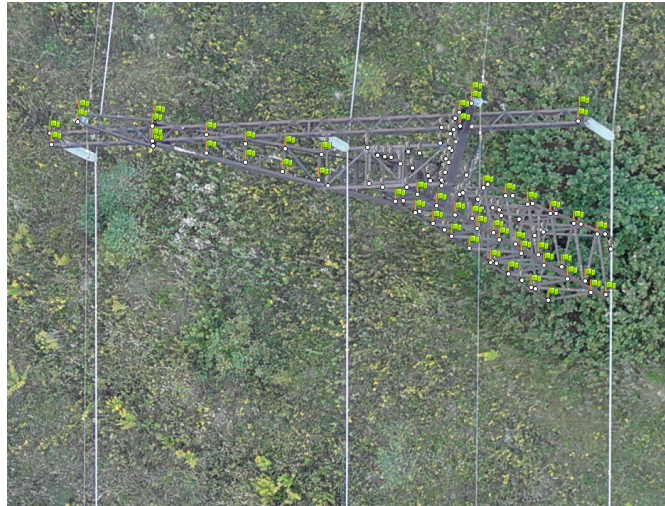


Рис. 5.3: Гистограмма меры схожести.

Отсутствие на данный момент открытых библиотек, решающих задачу, рассматриваемую в данной работе, усложняет оценку результатов. Было решено использовать функцию расстановки маркеров продукта Agisoft Photoscan, которая позволяет реконструировать отдельные точки трёхмерного пространства, отмечая их положения на нескольких выровненных фотографиях. Пример вставленных инструментом опор в построенную предварительно модель местности можно увидеть на рис. 5.4 и рис. 5.5. Жёлтыми точками показаны маркеры, поставленные по фотографиям. Некоторые маркеры для наглядности соединены белыми

линиями.

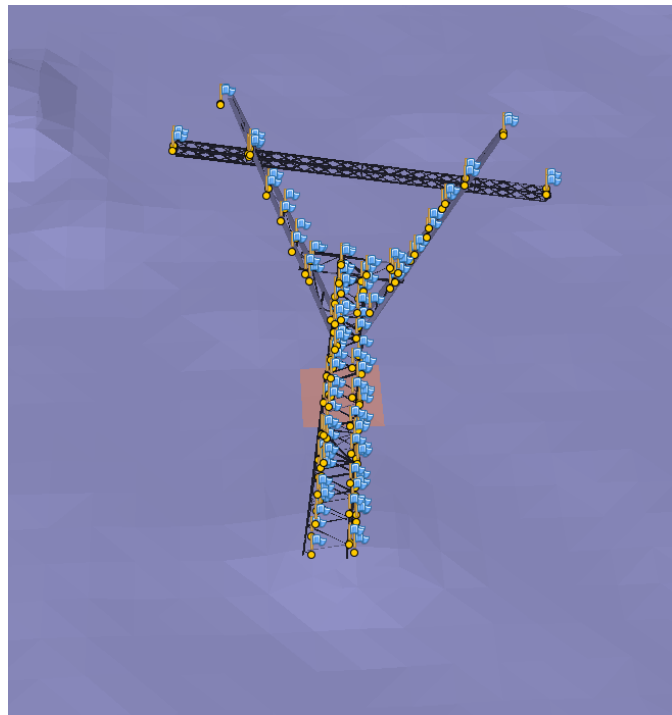
Видно, что результат в целом соответствует маркерам, самое большое смещение наблюдается обычно вдоль оси z (вертикальная ось) и составляет величину порядка 0.4 м. Это связано с тем, что на большинстве фотографий угол между осью z и направлением на опору мал.



(а) Опора, размеченная маркерами



(b) Вставленная опора ВЛ-220 (вид сбоку)

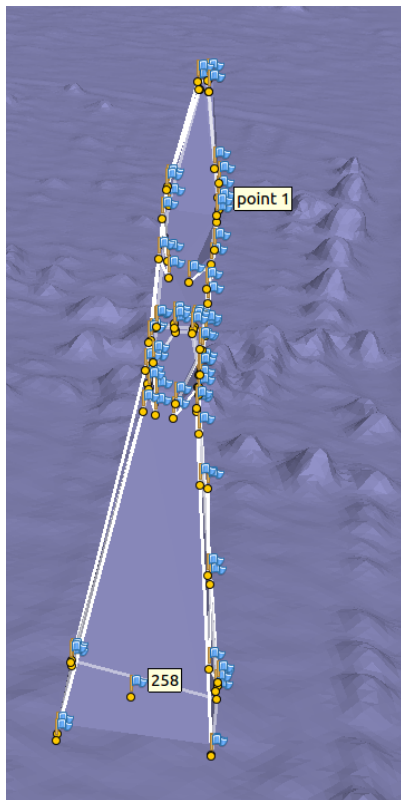


(c) Вставленная опора ВЛ-220 (вид спереди)

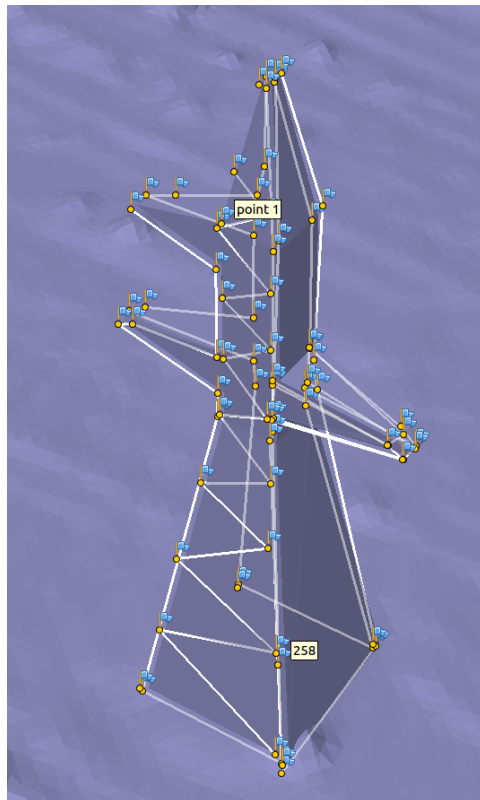
Рис. 5.4: Результаты вставки опоры в сцену.



(a) Опора, размеченная маркерами



(b) Вставленная опора У-110



(c) Вставленная опора У-110

Рис. 5.5: Результаты вставки опоры в сцену.

Среднее смещение	0.35	0.3	0.39	0.31
Максимальное смещение	0.44	0.51	0.63	0.55

Таблица 5.2: Оценки смещений контрольных точек

Для нескольких вставленных моделей были проведены измерения смещений ключевых точек опоры от соответствующих им маркеров, получившиеся средние смещения представлены в таблице 5.2.

Время работы алгоритма линейно зависит от числа фотографий опоры, и в среднем составило 27 секунд на фотографию на указанном оборудовании.

Заключение

В ходе выполнения работы исследованы методы, которые могут быть применимы для решения поставленной задачи. Реализован прототип решения, позиционирующий опоры с точностью порядка 1 метра и погрешностью курсового угла менее трех градусов. Показана возможность решения поставленной задачи описанным методом. Результаты позиционирования достаточно точны для визуализации коридора линий электропередач.

Дальнейшие шаги, которые можно предпринять в контексте работы, включают в себя:

- Замена алгоритма детектора на более надёжный. На данный момент планируется применить метод, описанный в [17].
- Улучшение точности определения координат. Можно попытаться использовать в качестве особых точек вершины разных найденных треугольников и применить подход, описанный в [28].
- Добавление автоматического выбора опоры из заданной библиотеки моделей. Для определения типа опоры в принципе можно использовать меру похожести, но такое решение приведет к линейному росту времени работы алгоритма в зависимости от размера библиотеки. Поэтому, предпочтительным будет определение типа опоры на стадии её детектирования.
- Проверка возможности детектирования существенных отклонений опоры от вертикальной оси. В текущем решении предполагается, что опора стоит практически вертикально, но естественно особый интерес представляют случаи, когда это не так.

Приложение

Модель стенопа

Классическая модель камеры-стенопа имеет вид, показанный на рис. 7.1.

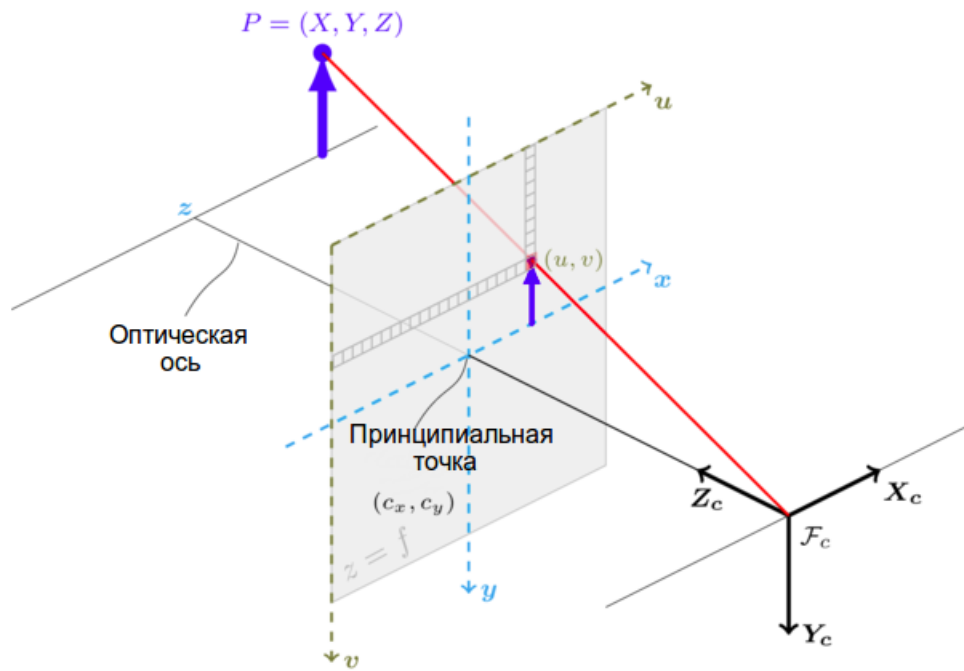


Рис. 7.1: Модель камеры-стенопа, изображение взято из [35].

Оптическая ось является лучом, исходящим из точки \mathcal{F} , в которой находится камера, в направлении, в которое камера направлена. Картинная плоскость - это плоскость, перпендикулярная оптической оси и пересекающая её на фокусном расстоянии в принципиальной точке. Картинную плоскость можно рассматривать в качестве поверхности, на которую с помощью центральной проекции через точку \mathcal{F} отображается фотографируемая сцена. Физически камера имеет ограниченный угол обзора/размер матрицы сенсора, поэтому кадр изображает часть картинной плоскости со спроецированной на неё сценой.

Чтобы описать проецирование камерой сцены, необходимо также ввести понятия внут-

ренных и внешних параметров камеры.

Внешние параметры камеры

Введём в рассмотрение две системы координат. Во-первых, будем считать, что в пространстве введена глобальная прямоугольная декартова система координат $OXYZ$.

Во-вторых, свяжем с камерой локальную систему координат $O_cX_cY_cZ_c$, имеющую центр в точке, где находится камера, ось Z_c , сонаправленную с оптической осью, O_cX_c и O_cY_c - горизонтальная и вертикальная оси камеры, так, что на получающемся снимке оси O_cX_c параллельна горизонтальная ось, проходящая слева направо через принципиальную точку, а оси O_cY_c параллельна вертикальная ось, проходящая сверху вниз через принципиальную точку.

Внешними параметрами камеры являются её позиция и ориентация относительно $OXYZ$, которые могут быть определены с использованием однородных координат в виде четырёхмерной квадратной матрицы

$$V = (T \cdot R)^{-1}, \text{ где}$$

$$R = \begin{pmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, T = \begin{pmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Будем считать, что оптическая ось неповёрнутой камеры направлена вдоль положительной полуоси OZ , а горизонтальная ось камеры сонаправлена с OX . По смыслу, $\vec{t} = (t_x, t_y, t_z)$ - это координаты камеры в глобальной СК, $\vec{r}_1 = (r_{11}, r_{21}, r_{31})$, $\vec{r}_3 = (r_{13}, r_{23}, r_{33})$ это координаты направляющих векторов осей локальной системы координат камеры. Чтобы преобразовать однородные координаты точки $P_{global} = (X, Y, Z, 1)$ из глобальной системы координат в камеральную, необходимо умножить их на матрицу V : $P_{camera} = VP_{global}$.

Внутренние параметры камеры

Определим двумерную прямоугольную декартову систему координат изображения $O_I X_I Y_I$, с центром в левом верхнем пикселе изображения, осью $O_I X_I$ направленной вправо, а осью $O_I Y_I$ вниз. В этой системе, координаты принципиальной точки будут равны $(c_x, c_y) \approx (\frac{W}{2}, \frac{H}{2})$, где W и H - ширина и высота кадра соответственно. Точное равенство достигается в случае несмещённой принципиальной точкой, что редко наблюдается на практике.

Внутренние параметры камеры состоят из матрицы калибровки и коэффициентов дисторсии. Матрица калибровки C для случая однородных координат выражается следующим образом

$$C = \begin{pmatrix} f_x & s & c_x & 0 \\ 0 & f_y & c_y & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Матрица калибровки является вырожденной, семантически это соответствует потере информации о глубине при проецировании сцены с помощью камеры. Величины f_x и f_y - это фокусные расстояния камеры по осям X_c и Y_c , выраженные в пикселях, а $s = f_y \tan \alpha$ - скошенность, которая не равна 0 в случае, когда пиксели камеры не являются квадратными, т.е. их углы отличаются от прямых на $\pm\alpha$. Обычно α крайне мал, но его следует учитывать для увеличения точности вычислений.

Так как нас обычно будет интересовать не вся фотография, а лишь вырезанная из неё область интереса, ограниченная прямоугольником с координатами левого верхнего угла (в пикселях) $p_{min} = (x_{min}, y_{min})$ и нижнего правого угла p_{max} , мы будем использовать несколько изменённую матрицу калибровки камеры $C(p_{min})$:

$$C(p_{min}) = \begin{pmatrix} f_x & s & c_x - x_{min} & 0 \\ 0 & f_y & c_y - y_{min} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

По сути, такое изменение матрицы калибровки задаёт перенос начала СК $O_I X_I Y_I$ в точку p_{min} . Пусть $I(P)$ - однородные координаты пикселя области интереса, в который отображается точка P , имеющая в глобальной СК координаты (X, Y, Z) . Тогда $I(P) = C(p_{min}) \cdot V \cdot (X, Y, Z, 1)$.

Кроме матрицы калибровки, идеальная камера имеет ненулевые коэффициенты дисторсии, за счёт которых пиксель изображения, в который проектируется произвольная точка P , не совпадает с вычисленным по предыдущей формуле пикселем. Различают радиальные коэффициенты дисторсии k_1, k_2, k_3 и тангенциальные p_1, p_2 . Чтобы устранить дисторсию изображения I , необходимо ко всем его пикселям (u, v) применить преобразование:

$$(x, y, z, 1) = C_{3 \times 3}^{-1} \cdot (u, v, 1), r = \sqrt{x^2 + y^2}$$

$$x' = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + p_1(r^2 + 2x^2) + 2p_2 xy$$

$$y' = y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + p_2(r^2 + 2y^2) + 2p_1 xy$$

$$(u', v', 1) = C \cdot (x', y', z), \quad I_{undistorted}(u, v) := I(u', v')$$

Литература

- [1] Matikainen, L., Lehtomäki, M., Ahokasa, E., Hyypää, J., Karjalainen, M., Jaakkola, A., Kukkoa, A., Heinonen, T., 2016. *"Remote sensing methods for power line corridor surveys"*, ISPRS J. of Photogram. Remote Sensing, том 119, стр. 10–31.
- [2] Guo, B., et al., 2014. *"Classification of airborne laser scanning data using JointBoost"*, ISPRS J. Photogram. Remote Sensing.
- [3] Kim, H.B., Sohn, G., 2010. *"3D Classification of power-line scene from airborne scanning data using random forests"*, Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci. 38 (часть 3A), стр. 126–132.
- [4] Jwa, Y., Sohn, G., Kim, H. B., 2009. *"Automatic 3D powerline reconstruction using airborne LiDAR data"*, IAPRS, том XXXVIII, часть 3/W8, 2009, стр. 105-110.
- [5] McLaughlin, R. A., 2006. *"Extracting Transmission Lines From Airborne LIDAR Data"*, IEEE Geoscience and Remote Sensing Letters, том 3, №2, стр. 222-226.
- [6] Xiang, Q., 2014. *"3D Reconstruction of 138 KV Power-lines from Airborne LiDAR Data"*, магистерская диссертация.
- [7] Li, Q., Chen, Z., Hu, Q., 2015. *"A Model-Driven Approach for 3D Modeling of Pylon from Airborne LiDAR Data"*.
- [8] Guo, B., Huang, X., Li, Q., Zhang, F., Zhu, J., Wang, C., 2016. *"A Stochastic Geometry Method for Pylon Reconstruction from Airborne LiDAR Data"*.
- [9] Sohn, G., Jwa, Y., Kim, H. B., 2012. *"Automatic Powerline Scene Classification And Reconstruction Using Airborne Lidar Data"*, ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, том I-3, 2012, стр. 167-172.
- [10] Yu, P., Dong, B., Xue, Y., 2012. *"Electric Power Tower Inclination Angle Detection Method Based on SIFT Feature Matching"*, Applied Mechanics and Materials Vols 236-237 (2012) стр 759-764.

- [11] Golightly, I., Jones, D., 2003. "*Corner detection and matching for visual tracking during power line inspection*", Image and Vision Computing 21 (2003) стр. 827–840.
- [12] Ting, F., Zhao, Y., Hu, X., Bing, X., 2012. "*A fast power tower extraction based on an improved algorithm for Harris corner detection*", Applied Mechanics and Materials тома 303-306 (2013) стр. 1072-1076.
- [13] Cheng, W., Song, Z., 2008. "*Power Pole Detection Based on Graph Cut*"
- [14] Tilawat, J., Theera-Umpon, N., Auephanwiriyaikul, S., 2010. "*Automatic Detection of Electricity Pylons in Aerial Video Sequences*", ICEIE 2010, том 1, стр. 342 - 346.
- [15] Araar, O., Aouf, N., Luis, J., Dietz, V., 2015. "*Power pylon detection and monocular depth estimation from inspection UAVs*", Industrial Robot: An International Journal, том 42 вып. 3 стр. 200 - 213.
- [16] Jiang, S., Jiang, W., Huang, W., Yang, L., 2017. "*UAV-Based Oblique Photogrammetry for Outdoor Data Acquisition and Offsite Visual Inspection of Transmission Line*".
- [17] Sampedro, C., Martinez, C., Chauhan, A., Campoy, P., 2014. "*A Supervised Approach to Electric Tower Detection and Classification for Power Line Inspection*", IJCNN 2014 стр. 1970-1977.
- [18] Correa, A. C., 2017. "*UAV vision system: Application in electric line following and 3D reconstruction of associated terrain*", докторская диссертация.
- [19] Hofer, M., Wendel, A., Bischof, H., 2013. "*Line-based 3D Reconstruction of Wiry Objects*", 18th Computer Vision Winter Workshop.
- [20] Hofer, M., Donoser, M., Bischof, H., 2014. "*Semi-Global 3D Line Modeling for Incremental Structure-from-Motion*", 18th Computer Vision Winter Workshop.
- [21] M. Hofer et al., 2016. "*Efficient 3D scene abstraction using line segments*", Computer Vision and Image Understanding.
- [22] Hofer, M., 2016. "*Building with Lines: Efficient 3D Scene Abstraction for the Built Environment*", Graz University of Technology, докторская диссертация.
- [23] Morarjee, B., 2016. "*Using Multiple View Geometry for Transmission Tower Reconstruction*", master thesis, University of Cape Town.
- [24] Duda, R. O., Hart, P. E., 1971. "*Use of the hough trasformtion to detect lines and curves in pictures*", Comm. ACM, том 15, стр. 11-15.

- [25] Grompone von Gioi, R., Jakubowicz, J., Morel, J.-M., Randall, G., 2010. *"LSD: A Fast Line Segment Detector with a False Detection Control"*, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, том 32, № 4, стр. 722-732.
- [26] Akinlar, C., Topal, C., 2011. *"EDLines: A real-time line segment detector with a false detection control"*, Pattern Recognition Letters 32 (2011) стр. 1633–1642.
- [27] Silerberg, T. M., Davist, L., Harwood, D., 1984. *"An iterative hough procedure for three-dimensional object recognition"*, Pattern Recognition, том. 17, № 6, стр. 621-629.
- [28] Olson, C. F., 1997. *"Efficient Pose Clustering Using a Randomized Algorithm"*, International Journal of Computer Vision 23(2), стр. 131–147.
- [29] Olson, C. F., 2001. *"A General Method for Geometric Feature Matching and Model Extraction"*, International Journal of Computer Vision 45(1), стр. 39–54.
- [30] Fischler, M. A., Bolles, R. C., 1981. *"Random Sample Consensus: A Paradigm for Model Fitting with Apphcatlons to Image Analysis and Automated Cartography"*, Communications of the ACM, том 24, стр. 381 - 395.
- [31] Linnainmaa, S., Harwood, D., Davis, L. S., 1987. *"Pose Determination of a Three-Dimensional Object Using Triangle Pairs"*, IEEE Transactions On Pattern Analysis And Machine Intelligence. том 10. №5.
- [32] DeMenthon, D. F., Davis, L. S., 1992. *"Model-Based Object Pose in 25 Lines of Code"*, Sandini G. (eds) Computer Vision — ECCV'92. ECCV 1992. Lecture Notes in Computer Science, том 588.
- [33] Collet, A., Martinez, M., Srinivasa, S. S., 2011. *"The MOPED framework: Object recognition and pose estimation for manipulation"*, The International Journal of Robotics Research 30(10), стр. 1284-1306.
- [34] *"Строительные нормы и правила Российской Федерации"*, 2017.
- [35] *OpenCV documentation*, URL: <http://docs.opencv.org/2.4/>