

ФЕДЕРАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ЦЕНТР ИНСТИТУТ  
ПРИКЛАДНОЙ МАТЕМАТИКИ ИМ. М.В. КЕЛДЫША РОССИЙСКОЙ  
АКАДЕМИИ НАУК

На правах рукописи

Новоторцев Леонид Владимирович

**ИССЛЕДОВАНИЕ И РАЗРАБОТКА АЛГОРИТМОВ ПОИСКА И  
ВОССТАНОВЛЕНИЯ НАСЕЛЕННЫХ ПУНКТОВ НА НАБОРЕ  
АЭРОФОТОСНИМКОВ**

Специальность 05.13.11 –

«математическое и программное обеспечение  
вычислительных машин, комплексов и компьютерных сетей»

Диссертация на соискание ученой степени  
кандидата физико-математических наук

Научный руководитель:  
доктор физико-математических наук  
Волобой Алексей Геннадьевич

Москва-2021

## Оглавление

Введение.....	3
1. Описание подхода.....	14
1.1. Обзор существующих технологий.....	14
1.2. Общее описание предложенного подхода.....	21
2. Обнаружение границ и отрезков.....	25
2.1. Существующие методы.....	25
2.2. Предложенный метод.....	29
2.2.1. Объединение по областям.....	32
2.2.2. Общее объединение.....	34
2.3. Результаты.....	36
3. Выделение областей.....	42
3.1. Связывание соседних отрезков в контуры.....	43
3.2. Замыкание областей.....	46
3.3. Результаты.....	47
4. Сопоставление областей.....	50
4.1. Метод сопоставления областей.....	50
4.2. Результаты.....	55
5. Трехмерная реконструкция крыш.....	58
5.1. Метод реконструкции крыш.....	58
5.2. Результаты.....	63
6. Результаты.....	66
6.1. Точность обнаружения.....	72
6.2. Точность трехмерной модели.....	74
Основные результаты.....	77
Приложение 1. Детали программной реализации.....	78
Список литературы.....	87

## **Введение**

### **Актуальность работы.**

В настоящее время индустриальный и технический прогресс достиг такого высокого уровня, когда имеющаяся информация устаревает с огромной скоростью. В связи с этим возникает задача быстрого получения актуальных сведений в той или иной области. Особенно это важно в тех случаях, когда данные перед использованием требуют обработки. Ручная обработка данных занимает крайне много времени, в результате чего полученные результаты могут быть уже недостаточно актуальными на момент завершения данного процесса. Это проявляется особенно явно при работе с огромными объемами исходных данных, которые, помимо прочего, нередко бывают представлены в виде, сложном для понимания человеком. Такая ситуация вызывает потребность в увеличении скорости извлечения необходимой информации из «сырых» данных. Большую роль в этом играет автоматизация обработки информации. Автоматический или даже полуавтоматический режимы обработки информации значительно ускоряют весь процесс, что в свою очередь ведет к увеличению актуальности результирующих данных.

Однако есть и негативная сторона автоматизации. Дело в том, что за значительное увеличение скорости работы во многих случаях мы платим ухудшением результирующего качества и точности. Нередко это не является существенным недостатком, но также часты случаи, когда точность также является важной характеристикой результатов. Именно попыткам достичь высокой точности результатов и скорости обработки исходных данных посвящены работы последних нескольких десятилетий [1].

Одной из таких областей, нередко сочетающей в себе большие объемы данных, необходимость их быстрой обработки и требование высокой точности результатов, является фотограмметрия и картография.

Фотограмметрия (от фото, γράμμα (древнегреческий) — запись, изображение, а метрия — измерение) — научно-техническая дисциплина, занимающаяся определением формы, размеров, положения и иных характеристик объектов по их фотоизображениям [2].

Картография (от греч. χάρτης — бумага из папируса, и γράφειν — рисовать) — наука об исследовании, моделировании и отображении пространственного расположения, сочетания и взаимосвязи объектов, явлений природы и общества [3].

В наши дни человек активно осваивает новые территории и изменяет вид уже освоенных, поэтому важно вести постоянное наблюдение как за прямыми проявлениями этого процесса, такими как, например, застройка и вырубка лесов, так и за вторичными, такими как, например, оседание грунта.

Благодаря научному прогрессу стало возможно достаточно оперативно делать снимки с самолетов, спутников и даже с БПЛА (БесПилотный Летательный Аппарат) и проводить дальнейшие исследования по полученным снимкам. Это значительно облегчает задачу слежения за интересующей территорией. На данный момент наиболее распространенным источником качественных снимков является аэрофотосъемка. В основном это вызвано наиболее оптимальным соотношением цены и качества "сырых" данных, полученных таким образом.

Пример аэрофотосъемки с двумя маршрутами представлен на Рис. 1. Для аэрофотоснимков характерны следующие свойства и характеристики:

- Высота съемки:
  - достаточно большая, чтобы избежать искажений в рамках одного объекта, вызванных его близким расположением к камере;
  - достаточно низкая, чтобы можно было не учитывать кривизну Земли при анализе соседних пар снимков;
- Качество снимков:
  - поскольку нет смысла экономить на камере, то съемка ведется при помощи наиболее дорогих камер, дающих хорошие, четкие снимки;
- Особенности полета:
  - самолет имеет стабильную траекторию без существенных отклонений;
  - все снимки сняты надирно, то есть съемка направлена строго вертикально вниз (или отклоняется настолько незначительно, что это не влияет на результат);
- Съемка по маршрутам:
  - съемка ведется "маршрутами": последовательно сделанными снимками, которые имеют большие пересечения между собой (40-80% от размера снимка);
  - один маршрут – это один заход самолета над интересующей территорией;
  - маршруты между собой тоже имеют пересечения;
- Доступна дополнительная информация о снимках:

- геолокационные данные (GPS, ГЛОНАСС и т.д.);
- направление съемки (ориентация камеры при съемке);
- высота съемки;
- параметры камеры (фокусное расстояние, дисторсия).

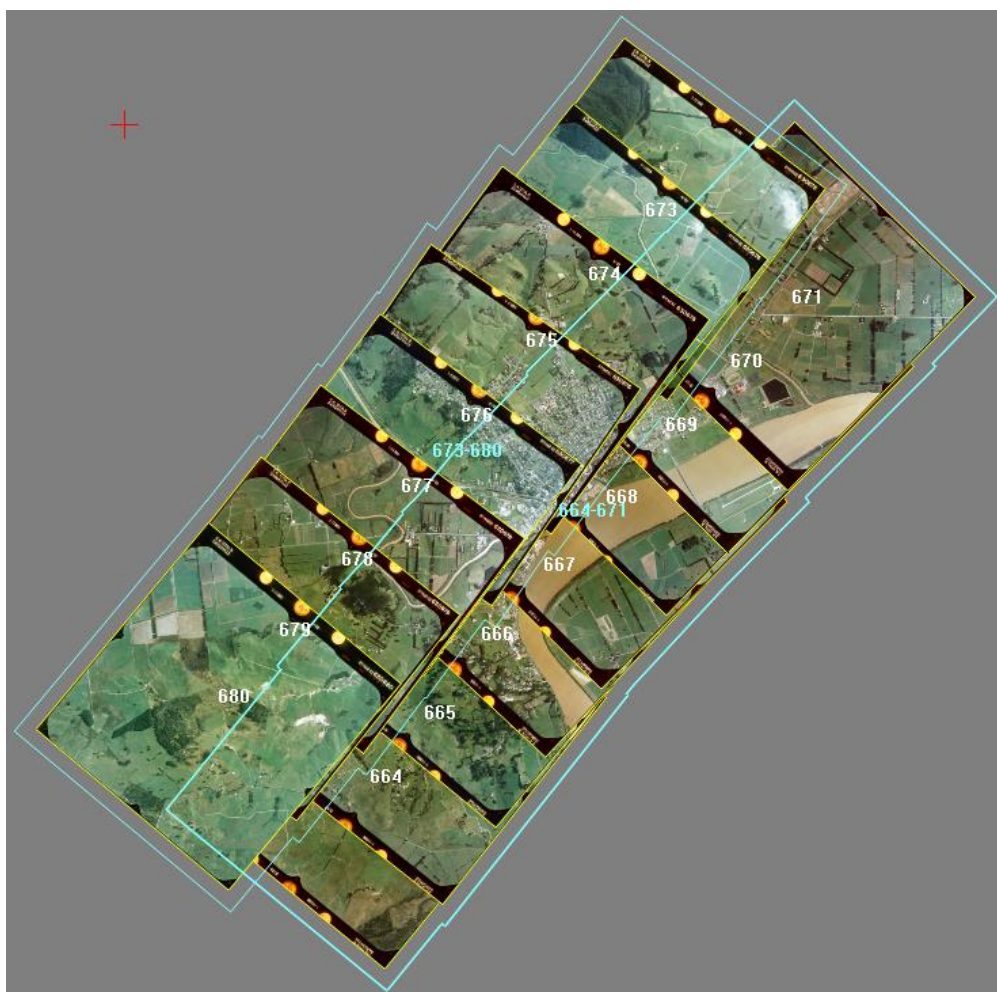


Рис. 1. Пример аэросъемки с двумя маршрутами

Одной из часто встречающихся задач картографии, связанных с анализом аэрофотоснимков, является обнаружение на них строений и их дальнейшая трехмерная реконструкция, то есть создание трехмерной

модели по снимкам. При этом ввиду ортогонального направления съемки аэрофотоснимков обнаружение и трехмерная реконструкция строений сводится в первую очередь к обнаружению и построению крыш зданий. Решению именно этой задачи и посвящена данная работа.

Существуют различные решения данной задачи, основанные на разных подходах и разных исходных данных. Подробный анализ существующих решений представлен в разделе 1.1, который показал, что требуется дальнейшее развитие существующих методов и разработка новых подходов.

### **Цели работы**

Целью работы является создание автоматической технологии, исследование и разработка соответствующих алгоритмов и методов обнаружения строений по снимкам, полученным в результате надирной (вертикальное направление съемки) аэрофотосъемки, с использованием данных ориентирования камер и построения их трехмерных моделей. Реализация разработанной технологии и алгоритмов в прикладных программных комплексах.

### **Задачи**

Представленный подход можно разбить на следующие задачи:

- 1) поиск и выделение отрезков, соответствующих попавшим на изображение границам объектов, на одном изображении;
- 2) выделение областей интереса, получаемых связыванием отрезков в замкнутые контуры, представляющие интерес для дальнейших операций, с целью уменьшения вычислительных затрат;

- 3) сопоставление областей и отрезков, соответствующих одним и тем же объектам, на различных изображениях для последующей трехмерной реконструкции по нескольким кадрам с разных ракурсов;
- 4) построение плоскостей внутри сопоставленных областей и получение трехмерной реконструкции за счет выполнения пересечения и ограничения построенных плоскостей.

### **Научная новизна**

Предложен новый подход обнаружения и трехмерной реконструкции населенных пунктов по аэрофотоснимкам, состоящий в том, что производится анализ каждого из снимков по отдельности с выделением областей интереса, их сопоставление и последующая реконструкция строений посредством перебора полуплоскостей, образованных ранее найденными отрезками.

Разработана модификация метода выделения отрезков LSD [4], позволяющая более точно выделять отрезки на аэрофотоснимках большого размера, что в свою очередь, существенно увеличивает точность остальных методов, используемых в данной работе.

Представлен новый подход выделения областей интереса (областей, в которых проводятся дальнейшие операции), позволяющей сократить в несколько раз (до 5 раз) площадь снимков, подвергающейся дальнейшей обработке.

Предложен новый быстрый метод сопоставления областей, опирающийся на поиск и анализ горизонтальных отрезков, а также на анализ взаимного положения областей.



В данной работе также разработана модификация метода перебора плоскостей, позволяющая на порядок ускорить перебор при сохранении точности итогового результата.

### **Теоретическая и практическая значимость**

В рамках данной диссертационной работы предложен новый многоэтапный подход, опирающийся на выделение, анализ и сопоставление линейных черт (отрезков) на снимках, а также их взаимное расположение. Это позволяет увеличить как скорость работы всей технологии, так и точность выделения и трехмерной реконструкции населенных пунктов.

На основе проведенного исследования разработана система обнаружения и восстановления населенных пунктов, которая позволяет производить обработку аэрофотоснимков как в полностью автоматическом режиме, так и в полуавтоматическом режиме с незначительным участием оператора.

На основе предложенного подхода была реализована и внедрена программная компонента в программное обеспечение "АО Фирма "Ракурс" ЦФС РНОТОМОД [5]. Этот продукт и программная компонента используется в «Роскартография» и в Федеральном агентстве кадастра объектов недвижимости («Роснедвижимость»).

### **Апробация**

Основные результаты диссертационной работы были представлены и обсуждались на следующих конференциях и семинарах:

- 17-ый научно-практический семинар «Новые информационные технологии в автоматизированных системах», Россия, Москва, 2014;

- 24-ая международная конференция по компьютерной графике и машинному зрению «Графикон-2014», Россия, Ростов-на-Дону, 2014;
- 23th International Conference on Computer Graphics, Visualization and Computer Vision WSCG, Czech Republic, Plzen, 2015;
- 25-ая международная конференция по компьютерной графике и машинному зрению «Графикон-2015», Россия, Протвино, 2015;
- 19-ый научно-практический семинар «Новые информационные технологии в автоматизированных системах», Россия, Москва, 2016;
- 26-ая международная конференция по компьютерной графике и машинному зрению «Графикон-2016», Россия, Нижний Новгород, 2016;
- 16-ая международная конференция CAD/CAM/PDM-2016 "Системы проектирования, технологической подготовки производства и управления этапами жизненного цикла промышленного продукта", Россия, Москва, ИПУ им. В.А. Трапезникова РАН, 2016.
- 27-ая международная конференция по компьютерной графике и машинному зрению «Графикон-2017», Россия, Пермь, 2017;
- 21-ый научно-практический семинар «Новые информационные технологии в автоматизированных системах», Россия, Москва, 2018;
- 28-ая международная конференция по компьютерной графике и машинному зрению «Графикон-2018», Россия, Томск, 2018.

- Семинар им. М.Р. Шура-Бура, Институт прикладной математики им. М.В. Келдыша РАН, Россия, Москва, 2019.

На основе предложенного подхода была реализована и внедрена программная компонента в программное обеспечение PHOTOMOD [5]. Этот продукт, а с ним и программная компонента, используется в «Роскартография» и в Федеральном агентстве кадастра объектов недвижимости «Роснедвижимость». Результаты, полученные в процессе использования разработанной технологии, достигают субпиксельной точности по плану и высокой точности обнаружения (порядка 0.9 по метрике F1) при более высокой скорости работы по сравнению с подходами, дающими аналогичную точность.

### **Публикации**

Материалы диссертации опубликованы в 15 печатных работах [6-20]. 5 статей было опубликовано в рецензируемых журналах, входящих в список изданий, рекомендованных ВАК [7, 10, 17, 18, 20], из них 2 публикации, входящие в системы цитирования Web of Science (WoS) и Scopus [18, 20]. 8 текстов докладов опубликовано в сборниках трудов международных конференций.

### **Личный вклад автора**

Все результаты получены лично автором. Разработка и реализация всех алгоритмов, программной компоненты программного комплекса Photomod, основанной на результатах исследования, выполнены автором лично.

### **Структура и объем диссертации**

Диссертация состоит из введения, шести глав, заключения, приложения (описания аспектов реализации) и списка литературы.

Первая глава посвящена анализу существующих методов, их классификации, а также общему описанию предложенного в данной работе подхода.

Вторая глава посвящена решению вопроса выделения прямых на исходном изображении. Суть этой задачи заключается в том, чтобы найти среди ранее найденных границ участки, которые лежат на одной прямой. Таким образом формируются отрезки. Важным нюансом данного этапа является возможное отсутствие границ в тех местах изображения, где они должны быть. Поэтому необходимо также проводить интеллектуальное объединение разрозненных отрезков.

В третьей главе представлен алгоритм выделения областей, в которых потенциально могут находиться строения. В рамках данной работы этот этап сводится к построению замкнутых контуров на основе ранее найденных отрезков. Здесь важным нюансом является проблема ненайденных отрезков. В этой главе также рассматривается вопрос допустимости выбранной оценки в рамках данной работы.

В четвертой главе рассматривается вопрос сопоставления найденных областей между различными снимками. На этом этапе для ускорения расчетов используются данные о взаимном и абсолютном положении снимков, полученные при съемке.

Пятая глава посвящена заключительному этапу приведенного подхода: сопоставлению конкретных элементов (отрезков) внутри соответствующих областей и построению трехмерной модели на основе параметров камеры и полученных результатах сопоставления. На этом этапе особенно важно учитывать, что на разных снимках один и тот же объект рассматривается под разными углами, и некоторые элементы могут быть

скрыты. Кроме того, на снимках, принадлежащих разным маршрутам, также может отличаться освещение и тени.

В шестой главе рассматриваются результаты работы предложенного подхода, и их сравнение с результатами, полученными при использовании других современных подходов.

В приложении «Детали программной реализации» освещены вопросы и проблемы, связанные с реализацией рассматриваемого подхода, а также вопросы ее оптимизации, рассматриваются инструменты и библиотеки, используемые в системе сборки.

В заключении сформулированы основные результаты работы.

## 1. Описание подхода

### 1.1. Обзор существующих технологий

Исследования в области обнаружения строений ведутся на протяжении последних трех десятков лет [1, 21-24]. Одной из первых работ, посвященных этой тематике, является работа Huertas А. [21], опубликованная в 1988 году. В этом исследовании рассматривается метод нахождения строений, имеющих плоскую прямоугольную или состоящую из прямоугольников крышу. Восстановление высоты строений происходит по теням, ими отбрасываемым. Очевидно, что данный подход имеет сильно ограниченную область применения и низкую точность трехмерной реконструкции.

С тех пор данная предметная область активно развивалась как за счет совершенствования и улучшения алгоритмов [24-27], так и за счет увеличения качества дополнительной информации, получаемой вместе со снимками [23, 28, 29].

Я предлагаю классификацию существующих подходов, представленную на Рис. 2. Она позволит более структурировано и подробно составить их обзор и рассмотреть каждую группу отдельно.

**Методы на основе обработки данных космосъемки.** Космические снимки — это данные, получаемые с помощью космических аппаратов в результате съемки в различных диапазонах электромагнитного спектра. Полученные данные визуализируются для дальнейшей обработки. Одним из принципиальных отличий космической съемки является сканерный принцип получения снимка. Это значит, что снимок получается построчно по мере пролета производящего съемку аппарата. Поэтому элементы

внешнего ориентирования предоставляются не для снимка целиком, а на каждую строку снимка.

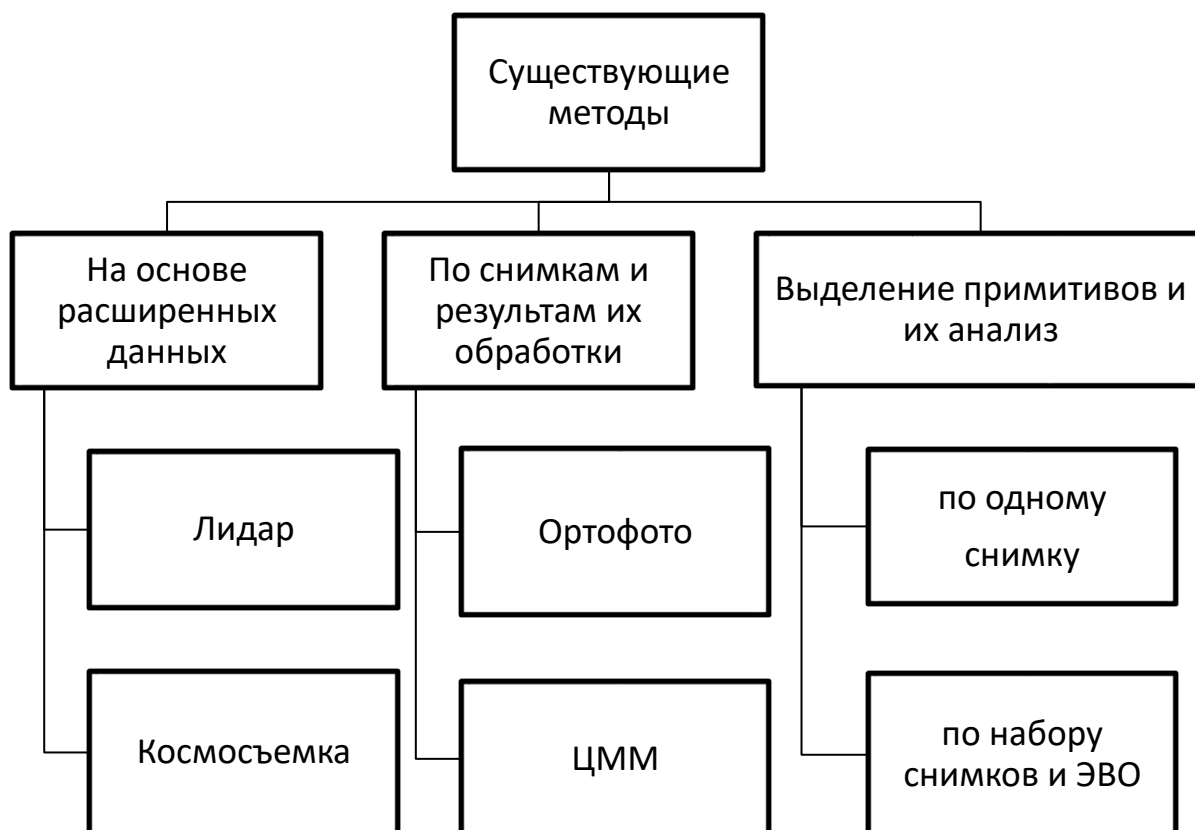


Рис. 2. Классификация основных методов обнаружения и реконструкции строений.

Несмотря на то, что подходы обнаружения и реконструкции строений, ориентированные на работу с "сырыми" растрами, часто применяют и для космоснимков [30, 31], существует отдельная группа подходов, связанных непосредственно с их обработкой [23, 32, 33]. Они используют специфику получения космоснимков (сканерная съемка) и дополнительные данные (элементы внешнего ориентирования, дополнительные каналы

изображений и т.д.). Кроме того, спутниковая съемка предоставляет снимки крайне высокого разрешения, что также является их преимуществом.

Алгоритмы, относящиеся к данной категории, имеют довольно высокое качество обнаружения (точность достигает 84% [23]), но из-за малой доступности качественных космоснимков их применение ограничено, хотя и в меньшей степени, чем алгоритмов, использующих данные лидара.

**Методы на основе обработки данных лидара (LIDAR англ. Light Detection and Ranging** — световое обнаружение и определение дальности) — это оптический локатор для дистанционного зондирования воздушных и водных сред. Также к лидарам относят оптические локаторы, которые позволяют дистанционно получать информацию о твердых объектах. Как правило, лидар содержит источник оптического излучения (лазер), телескоп с приемником света, установленные на общем поворотном основании, а также систему регистрации и обработки результатов зондирования, устройства управления и отображения информации [34].

Подходы, основывающиеся на полученных посредством лидара данных, имеют большую точность обнаружения строений и их восстановления: точность составляет около 90% и даже достигает 100% на некоторых наборах данных [28, 29, 35-40]. Это достигается за счет большой точности лазерного сканирования местности и наличия данных о высотах снимаемой местности.

Существенным недостатком методов, основанных на обработке данных лидара, является стоимость и доступность исходных снимков. Наличие данных, свободно распространяемых агентством USGS (United States Geological Survey - Геологическая съемка США), не решают эту



проблему полностью, поскольку для этих данных не гарантируется точность, актуальность и завершенность [41].

**Методы на основе обработки данных Цифровой Модели Местности, ЦММ (digital terrain model, DTM)** – цифровое трехмерное представление высот поверхности земли. Обычно данные ЦММ представляются в виде матрицы высот – данных, подобных одноканальным растрам, в которых значение в каждом пикселе соответствует высоте поверхности Земли в соответствующих этому пикселю геодезических координатах.

**Методы на основе обработки данных Ортофотоплана.** Одним из основных продуктов, получаемых на выходе цифровых фотограмметрических систем (ЦФС) являются ортофотопланы. При их построении на исходных снимках корректируются геометрические искажения, связанные с параметрами камеры, наклоном оптической оси, рельефом местности и т.п. Результат ортотрансформирования в виде единого непрерывного растра или набора листов представляется в какой-либо картографической проекции [42]. Ортофотопланы могут служить не только наглядным, но и геометрически точным источником пространственной информации. Ортотрансформирование позволяет устранить значительную часть искажений и установить единый масштаб по всему полю изображения, что дает возможность проводить измерения и определять координаты с высокой точностью. Однако при классическом способе ортотрансформирования крупномасштабных изображений городских территорий возникает ряд проблем. Одна из них – «мертвые» зоны, т. е. области, закрытые высотными строениями вследствие перспективы изображения. Именно это обстоятельство накладывало ограничения на применение ортофотопланов при решении различных задач муниципального управления, так как часть важной информации (улицы,

проезжая часть и тротуары, канализационные люки и т. п.) могла быть скрыта. Для решения этой проблемы была создана технология TrueOrtho [43] («истинное орто»). TrueOrtho это ортотрансформированные изображения (ортофото), в которых учтены не только смещения, возникшие из-за влияния рельефа местности, но и смещения крыш высотных сооружений вследствие перспективы изображения, а невидимые области восстановлены с использованием перекрывающихся снимков. TrueOrtho базируется не на цифровой модели рельефа, как классическое ортофото, а на цифровой модели местности, содержащей подробную информацию обо всех высотных объектах [44].

Использование таких данных, как ЦММ [45-50] или ортофото [35, 51-57] значительно увеличивает как точность нахождения строений, так и упрощает восстановление трехмерной модели. Точность этих подходов достигает 85%-87% [35,51].

Однако надо понимать, что ЦММ и ортофото изначально получаются в результате крайне ресурсоемкой обработки изображений (аэрофотоснимков в частности). Это влияет либо на доступность исходных данных в случае работы с уже готовым ортофото и ЦММ, либо на время работы алгоритма. Получение ортофото может занимать часы или даже сутки для реальных данных [54-57]. ЦММ более доступны и алгоритмы ее получения менее ресурсоемкие, но все равно занимает часы на современных компьютерах [58-61].

**Методы на основе выделения примитивов на снимках.** Одним из наиболее распространенных подходов решения задачи нахождения строений на аэрофотоснимках является выделение на снимках примитивов (прямых, отрезков) или сегментация этих снимков. Это связано с тем, что строения в своем большинстве имеют линейные черты и легко

сегментируются. Данное направление можно разделить на еще два, каждое из которых имеет особые подходы к решению задачи.

**Методы выделения примитивов на одиночных снимках.** Это самый распространенный и доступный формат данных, используемый для обнаружения строений. Однако из-за отсутствия стерео или другой позволяющей восстановить высоту информации, возникает необходимость в использовании контекстной информации снимков, такой как тени или определение направления освещения [62-65]. Типичными методами, применяемыми в данном классе алгоритмов, являются выделение границ, нахождение прямых и сегментирование.

Данный класс алгоритмов отличается сравнительно малой точностью распознавания (70-80%) [64] и существенными сложностями при решении задачи трехмерной реконструкции строений по снимкам. Ключевую роль в таких алгоритмах играет нахождение и анализ теней, отбрасываемыми различными объектами. Это в свою очередь накладывает ограничение на область использования подобных методов. К примеру, существенной сложностью является обнаружение строений, полностью попавших в тень других строений, являющихся частью крайне плотной застройки. Помимо этого, снимки, сделанные тогда, когда солнце находится в зените, являются непригодными для обработки большинства методов, относящихся к данной категории.

**Методы выделения примитивов на наборе снимков с ЭВО (Элементами Внешнего Ориентирования).** Во многом данный класс алгоритмов похож на алгоритмы выделения примитивов на одиночных снимках. Основным исключением является наличие более подробной информации об объектах на каждом из снимков за счет использования стерео. Это становится возможным благодаря использованию ЭВО: информации о позиции, направлении и других параметров съемки.

Основными методами, используемыми в данном подходе помимо выделения примитивов, являются методы, ориентированные на сопоставление найденных примитивов и анализ полученных сопоставлений [66-69].

Подобный подход позволяет значительно увеличить точность нахождения строений (точность составляет 87-92%) по сравнению с методами, опирающимися на анализе одного изображения. Кроме того, наличие стерео позволяет достаточно точно восстанавливать линейные размеры найденных строений. Основным недостатком такого подхода является медленная скорость вычислений, вызванная необходимостью выполнять сопоставления элементов на снимках между собой, перебирая их.

Наибольший интерес представляют подходы, основанные на анализе аэрофотоснимков. Это связано с тем, что такого рода данные наиболее доступные и дешевые. Кроме того, ЦМР, ЦММ и ортофотоплан являются продуктами обработки наборов снимков. При этом, одиночные снимки без данных ориентирования редко представляют интерес как в задачах геодезии, так и в трехмерной реконструкции объектов, поскольку точность получаемых результатов является низкой из-за недостаточной информации об объектах. Таким образом, наибольший интерес представляют подходы, основанные на анализе набора изображений с ЭВО.

Как указывалось выше, методы выделения примитивов на таком типе данных зачастую работают медленно. По этой причине предлагается разработать подход, который позволил бы увеличить скорость работы за счет применения как существующих, так и новых методов, специфичных для задачи обнаружения и трехмерной реконструкции строений.

Предлагается использовать идею грубой предобработки снимков с целью снизить количество сопоставлений примитивов. Данная работа посвящена разработке именно такого подхода.

## **1.2. Общее описание предложенного подхода**

Предлагается использовать подход, основанный на обработке нескольких аэрофотоснимков одной местности с информацией о внешнем ориентировании за счет высокой доступности исходных данных с высокой потенциальной точностью. Основным недостатком подобных подходов является скорость их работы, вызванная необходимостью обрабатывать и выполнять сопоставление снимков целиком. В случае промышленной аэрофотосъемки, где размер каждого снимка составляет несколько гигабайт, а количество снимков может достигать нескольких десятков, такой недостаток является критическим.

По этой причине значительный упор в данной работе делается на ускорение расчетов и ограничение площади, на которой применяются ресурсоемкие алгоритмы. Для достижения этого предлагается использовать выделение примитивов на снимке, в частности, отрезков, в качестве основы описываемого подхода. Это в свою очередь позволяет использовать выделенные отрезки в качестве базового элемента для последующих операций, тем самым уменьшая ресурсоемкий анализ растровых данных.

Для дальнейшего ускорения предлагается использовать выделение областей, представляющих интерес для дальнейших вычислений, с отсечением остальных частей изображения, не содержащих полезной информации. Эта идея реализуется с помощью следующего четырехступенчатого подхода:

- 1) обнаружение границ и отрезков;
- 2) выделение областей;
- 3) сопоставление областей;
- 4) Трехмерная реконструкция.

Поскольку основной целью данного подхода является достижение высокой скорости, то важно, чтобы выделение областей и их сопоставление производилось быстро, иначе потенциальный выигрыш от этой операции будет полностью аннулирован временем ее выполнения. Поэтому был разработан новый метод выделения областей, основанный на простых операциях, таких как расчет расстояний между отрезками и анализ их взаимного положения, для построения многогранной области.

После того, как получены области интереса, которые ограничивают количество обрабатываемых данных, уже можно приступить к сопоставлению примитивов и построению трехмерной модели. Но можно добиться еще большего ускорения за счет предварительного сопоставления самих областей, тем самым ограничив количество сопоставлений более примитивных элементов между собой, ограничивая их границами сопоставленных областей.

Использование существующих методов сопоставления не обосновано, поскольку приводит к потере выигрыша во времени, даваемого предложенным подходом. По этой причине был разработан оригинальный подход сопоставления для областей, основывающийся на информации о том, что для описываемого подхода наибольший интерес представляют

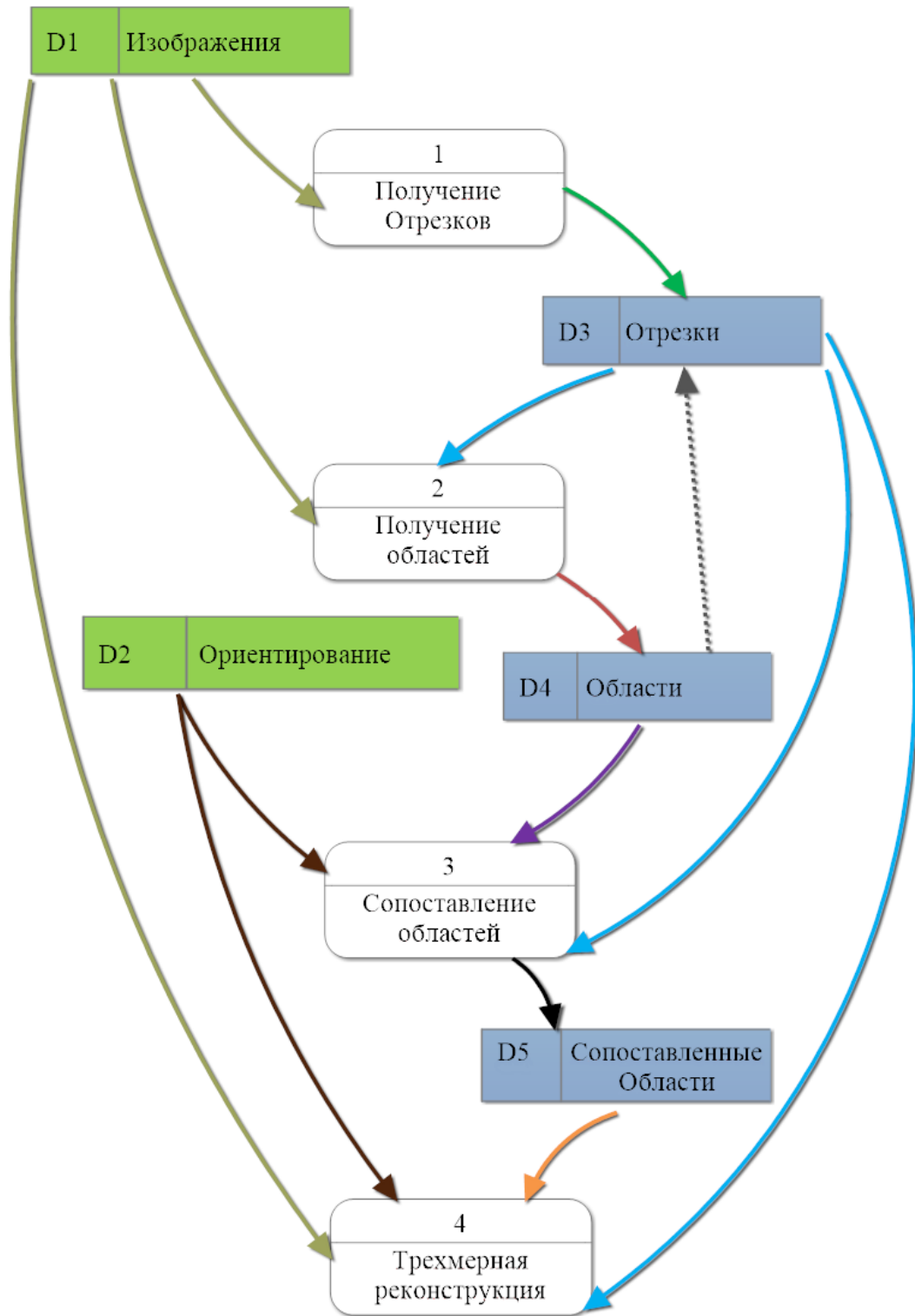


Рис. 3. Диаграмма потоков данных предложенного подхода.

строения, и на информацию о внешнем ориентировании снимков. При этом вместо того, чтобы сразу принимать решение о соответствии, выстраивается список «похожих» областей в порядке убывания их корреляции. Этот метод позволяет выполнить быстрое сопоставление областей без потери точности за счет того, что в большинстве случаев достаточно будет первой области из списка для выполнения сопоставления.

После проведения описанных выше операций уже можно проводить непосредственно построение трехмерной модели. Для решения этой задачи в работе предлагается использовать метод перебора плоскостей, заключающийся в переборе всех полуплоскостей (с заданным шагом дискретизации угла поворота), образованных каждым из сопоставленных отрезков и выборе той, что дает наибольшую оценку «похожести» на сопоставленных снимках. Также предлагается далее ускорить данный метод за счет использования смежных отрезков к рассматриваемым (два смежных отрезка потенциально дают 3 точки, однозначно задающих плоскость). Однако из-за погрешности нельзя использовать смежные отрезки для однозначного сопоставления, так как они могут располагаться не на пересекающихся, а на смежных прямых. По этой причине их можно использовать только как «подсказки» для ограничения перебора полуплоскостей. На Рис. 3 представлена диаграмма потока данных, иллюстрирующая общую схему работы представленного подхода. Детали программной реализации этого подхода приведены в Приложении 1.



## 2. Обнаружение границ и отрезков

Первым шагом представленного подхода является поиск отрезков на изображении. Существует несколько подходов решения этой задачи:

### 2.1. Существующие методы

- Метод Хафа
- Метод LSD (Line Segment Detector)
- Метод EdLines

**Метод Хафа** [70] подразумевает предварительное вычисление и нахождение граней на изображении. Для этого строятся карты градиентов яркости изображения в каждой точке и значения вычисленных градиентов используются в качестве показателя границы. На этом основывается фильтр Собеля [71]. Однако он не представляет бинарного решения, которое необходимо для метода Хафа. Решению задачи бинаризации границ изображения посвящено много работ [72-79]. За счет высокого качества результатов, простоты реализации и высокой скорости работы самым распространенным методом является фильтр Кэнни [72] и его модификации [76-79].

Метод Хафа основан на методе голосования. Общий подход заключается в том, что для каждого пикселя, относящегося к грани, строятся все возможные прямые, проходящие через этот пиксель. Если рассматривать параметрическое задание прямых, то получается, что для данного пикселя мы перебираем углы с заданным шагом. Выбранный шаг определяет точность обнаружения (чем меньше, тем точнее) и существенно влияет на скорость работы метода (чем меньше, тем медленнее). Для точки и угла вычисляем величину нормали, опущенной из начала координат.

Полученные параметры (угол и нормаль) округляем в соответствии с наперед заданной точностью и прибавляем единицу в таблицу (строки - углы, столбцы - нормаль). Далее, после заполнения таблицы проводится ее анализ и прямые, соответствующие локальным максимумам в таблице, помечаются как найденные.

В изначальном виде метод Хафа редко используется из-за больших временных затрат, отсутствия нахождения концов отрезков и большой чувствительности к шуму. Основной модификацией этого алгоритма является подход, основанный на построении деревьев из сегментов изображения, применения метода Хафа к листьям этого дерева и последующее объединение результатов [80].

Существует большое количество модификаций, направленных на уменьшение ошибок дискретизации [81, 82], подбору критерия определения выделения прямой по таблице и скорости работы метода в целом.

Особого внимания заслуживает модификация метода Хафа, основанная на вероятностном подходе [83]. Этот подход позволяет не только находить прямые, но и их концы. Такой подход дает большую точность обнаружения, устойчивость к ошибкам и значительно увеличенную скорость работы.

**Метод LSD (Line Segment Detector).** Принципиально другой алгоритм был предложен в работе [4], в которой предлагается метод, состоящий из трех этапов.

- 1) Выделение областей, потенциально содержащих прямые на основе анализа направления градиентов в каждом пикселе
- 2) Приведение найденных сегментов к отрезкам
- 3) Валидация всех найденных прямых

Существенное улучшение этого подхода, которое получило широкое распространение, получило название LSD (Line Segment Detector) [84, 85]. В этом подходе предлагается использовать те же этапы работы алгоритма. При этом этапы 1 и 2 остаются как есть, только с небольшими модификациями. А пункт 3 заменяется на валидацию, опирающуюся на исследования, представленные в [86, 87].

Метод LSD имеет ряд значительных преимуществ перед алгоритмом Хафа. К ним можно отнести следующие:

- значительно более высокая скорость работы (примерно в 80 раз на исследуемых примерах);
- значительно меньшая чувствительность к шуму (к примеру, участки с растительностью не детектируются), что упрощает обработку полученных данных;
- высокая точность обнаружения отрезков.

К недостаткам можно отнести только одно: при использовании данного метода на снимках, имеющих высокий диапазон контрастности и большое количество деталей, состоящих из отрезков, некоторые отрезки не находятся, и необходимо либо масштабировать изображение, либо разбить его на сегменты [85].

**Метод EDLines (Edge Drawing Lines)** [88] является еще одним распространенным алгоритмом нахождения отрезков на изображениях. Идея данного подхода схожа с методом LSD, только гораздо больше внимания уделяется скорости алгоритма (он позиционируется, как алгоритм реального времени) и неразрывности нахождения отрезков.

Преимущества данного алгоритма вытекают из постановки задачи этого метода: высокая скорость обработки данных и высокая точность выделения отрезков.

К существенным недостаткам этого метода относится дублирование при обнаружении некоторых отрезков и образование большого количества близлежащих параллельных прямых. Примером такого недостатка являются прямые, образованные коньком крыши на Рис. 4.



Рис. 4. Отрезки, обнаруженные методом EDLines, реализованным в OpenCV [89].

В рассматриваемой задаче обнаружения строений данный недостаток является критическим: наличие отрезков, относящихся к одной грани (крыше) существенно усложняет сопоставление отрезков между

изображениями и последующую трехмерную реконструкцию строений, и впоследствии может потенциально привести к ошибкам построения.

## 2.2. Предложенный метод

В связи с тем, что пропущенные отрезки могут оказать значительное влияние на итоговый результат, в данной статье предлагается использовать подход, позволяющий эту ошибку минимизировать.

В первую очередь все исходное изображение разбивается на одинаковые квадратные сегменты (возможно за исключением крайних областей). Размер области определяется точностью, которую мы хотим получить при определении угла наклона отрезка или прямой. Чем больше сторона области, тем точнее можно определить угол поворота прямой в каждом сегменте. С другой стороны, при увеличении области увеличивается вероятность пропуска отрезков или погрешность нахождения концов. Мною была выведена зависимость между размером сегмента и точностью нахождения отрезка (его угла наклона), которая представлена ниже.

$\begin{cases} \Delta x = L \cos(\alpha) - \cos(\alpha + \Delta\alpha)  \\ \Delta y = L \sin(\alpha) - \sin(\alpha + \Delta\alpha)  \Leftrightarrow \\ \Delta x + \Delta y \geq \gamma \end{cases}$ $\Leftrightarrow 2L \sin\left(\frac{\Delta\alpha}{2}\right) \geq 2L \left  \sin\left(\frac{\Delta\alpha}{2}\right) \right  \sqrt{1 - \sin(2\alpha + \Delta\alpha)} = \Delta x + \Delta y \geq \gamma \Leftrightarrow$ $\Leftrightarrow 2L \geq \frac{\gamma}{\sin(\Delta\alpha/2)}$	(1)
---	-----

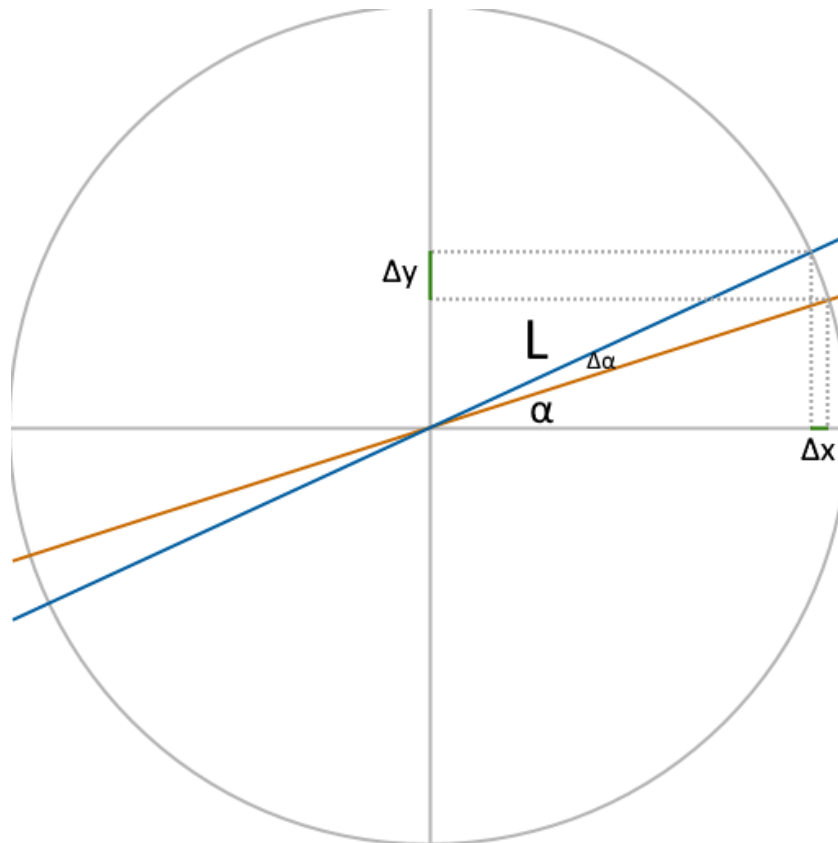


Рис. 5. Вычисление размера сегмента.

Здесь  $L$  – размер сегмента,  $\alpha$  – угол наклона отрезка относительно горизонтали,  $\Delta\alpha$  – погрешность построения отрезка,  $\Delta x$  и  $\Delta y$  – погрешности обнаружения концов отрезка по осям  $OX$  и  $OY$ .  $\gamma$  – «ширина» линии, обуславливаемая качеством снимка (чем эта величина больше, тем более широкая и размытая линия на изображении, и, соответственно, сложнее различить наклон отрезка). На Рис. 5 представлена иллюстрация для этих вычислений. Для облегчения вычислений рассматриваем область, как окружность. Это гарантирует верность расчетов и для прямоугольной области со стороной равной радиусу окружности.

В Рис. 6 приведены соответствие размера стороны области и погрешности в определении угла для  $\gamma=2$  и  $\gamma=4$  пикселей. Далее в работе

подразумевается, что используется значение размера сегмента, соответствующее погрешности в один градус.

После этого в каждой области применяется один из алгоритмов нахождения отрезков. Наибольшую эффективность показал метод LSD, применяемый на квадратных сегментах изображения 250x250 пикселей. Таким образом, точность расчета угла может превышать 1 градус. Детали программной реализации обработки сегментов приведены в Приложении 1.

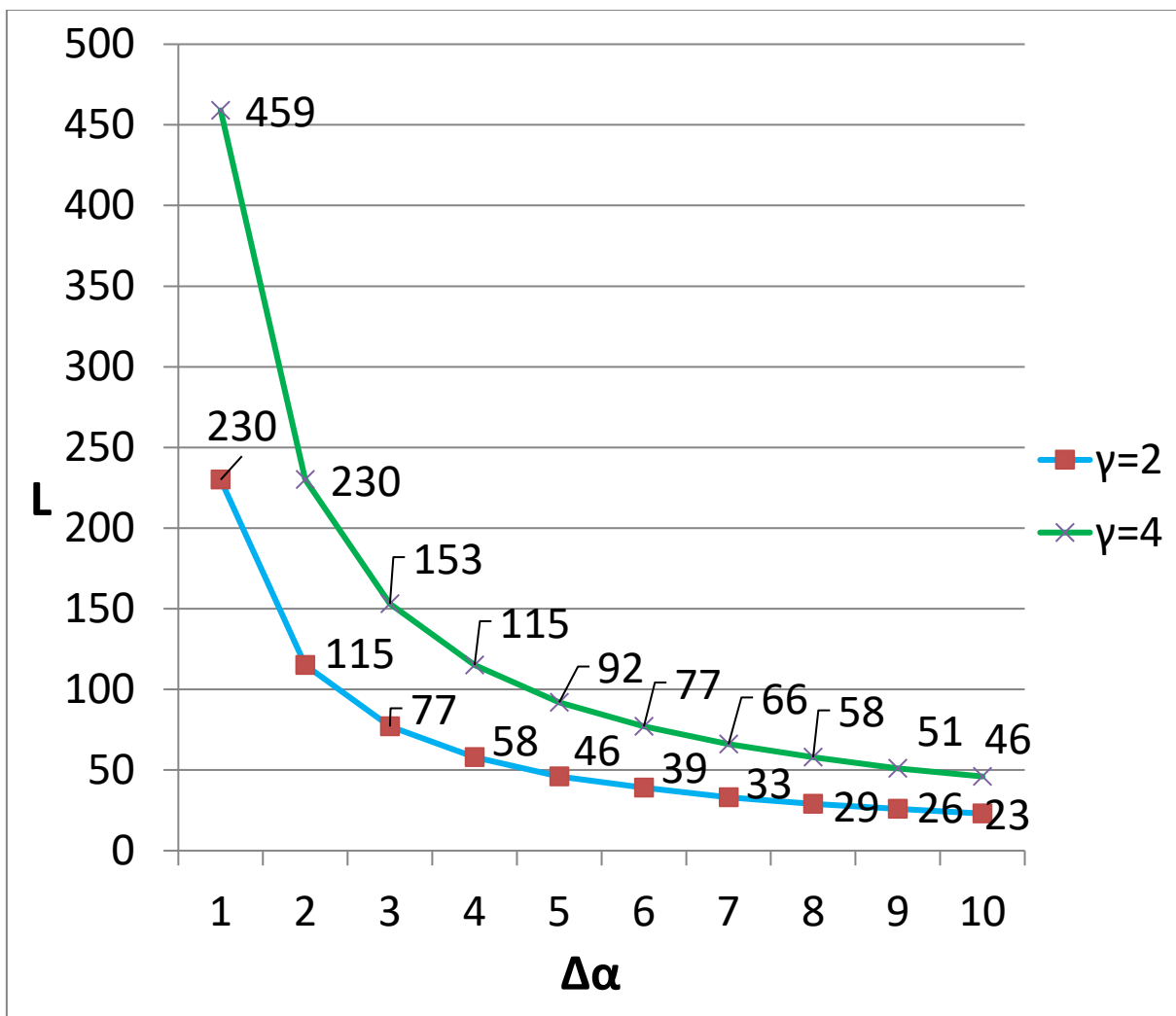


Рис. 6. Зависимость размера сегмента от желаемой погрешности обнаружения отрезка.  $L$  – размер сегмента,  $\Delta\alpha$ - погрешность построения отрезка (угол наклона),  $\gamma$  – «ширина» линии

Поскольку в разных сегментах изображения могут оказаться части одного и того же отрезка, то необходимо произвести объединение его частей. При этом необходимо учитывать, что отрезки могут быть не обнаружены в заданной области или разбиты на сегменты внутри области из-за низкого контраста участков изображения или особенностей алгоритма обнаружения отрезков. Поэтому необходимо выполнять более детальную проверку на дальнейшее объединение отрезков.

### **2.2.1. Объединение по областям**

На данном этапе выполняется объединение отрезков, находящихся в соседних областях. При этом необходимо учитывать, что отрезки, принадлежащие к одной линии, могут иметь разные углы из-за ошибок дискретизации на предыдущем этапе.

Был разработан следующий алгоритм объединения отрезков:

1. В первую очередь все найденные отрезки помечаем как необработанные.
2. Последовательно проходим по всем областям, на которые было разбито изображение.
3. В каждой области обрабатываем полученные в нем отрезки, которые все еще помечены как необработанные.

Рассмотрим алгоритм обработки отрезков. В целях ускорения расчетов для рассматриваемого отрезка сначала вычисляем участок границы области (далее будем называть это образом отрезка), на котором может находиться конец отрезка с учетом погрешности (на Рис. 7 выделен красным). В случае если отрезок не пересекает границ области, то рассматривается пересечение прямой, на которой он лежит с границами



области. Продолжение отрезка ищется в тех соседних областях, на границу которых попал вычисленный нами участок.

Для рассматриваемых отрезков при выполнении проверки необходимо учитывать, что концы могут не совпадать, как и углы. Поэтому объединение выполняется при выполнении следующих условий.

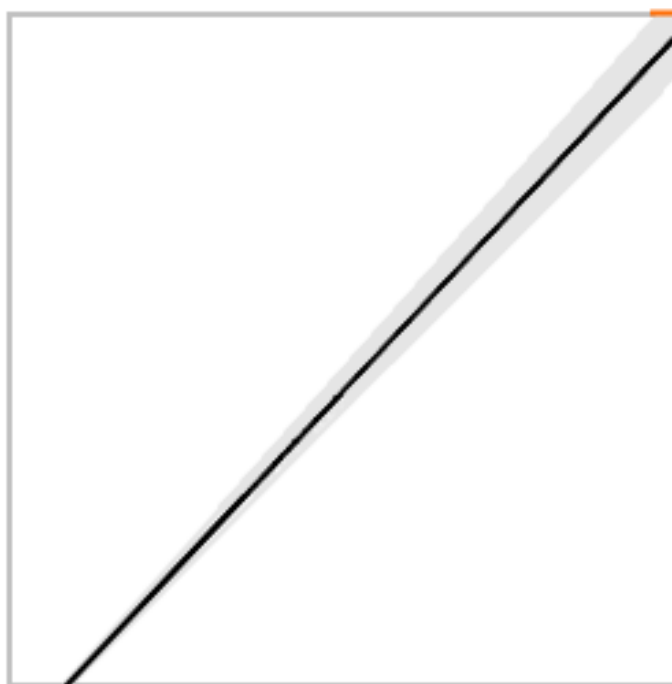


Рис. 7. Образ отрезка на границе сегмента.

- 1) Ближайшие концы отрезков находятся на расстоянии, не превышающем 10 пикселей (значение выбрано исходя из особенностей алгоритма LSD).
- 2) На основе формулы (1) и графика зависимости ошибки от размера сегмента (Рис. 6) выбирается допустимая ошибка угла наклона отрезков по длине наименьшего из отрезков. Разница

между углами наклона этих отрезков не должна превышать полученное значение допустимой ошибки.

Если вышеприведенные условия соблюдены, то выполняется слияние отрезков. Оно происходит в несколько этапов.

- 1) Усредняются параметры прямых, на которых лежат отрезки. При усреднении в качестве весов используются длины объединяемых отрезков. Это делается, чтобы избежать существенного искажения, вызванного отрезками малой длины, которые имеют большую погрешность определения угла.
- 2) Концы отрезков определяются как проекция дальних концов объединяемых отрезков на прямую, полученную в результате усреднения параметров.

В случае если на каком-либо этапе в соседних сегментах не найдено продолжения для отрезка, то отрезок экстраполируется на этот сегмент изображения. Если длина продолжения, определяемая границами сегмента, не превышает  $1/3$  стороны этого сегмента, то далее проверка ведется, как если бы этот сегмент реально существовал. Если и на следующей итерации не обнаружилось сегмента или его мнимая длина превышает  $1/3$  стороны сегмента, то считается, что найдено окончание отрезка с рассматриваемой стороны.

### **2.2.2. Общее объединение**

После полного выполнения операций, описанных в предыдущем разделе, мы получаем набор отрезков. Однако для того, чтобы максимально устранить разрывы, необходимо продолжить слияние отрезков, только уже используя больше информации об изображении.

Несмотря на уже выполненное объединение отрезков между областями, результат может быть неудовлетворительным, поскольку отрезки в некоторых областях могут быть не найдены из-за низкого контраста или особенностей алгоритма LSD.

Для устранения подобного рода ошибок необходимо произвести дополнительные проверки на возможные объединения прямых.

Для каждого необработанного отрезка проверяется наличие отрезков на заданном расстоянии. Это значение является входным параметром моего подхода и соответствует среднему размеру строений в пикселах. В представленной работе используется значение по умолчанию, соответствующее 15-и метрам для рассматриваемого проекта, что соответствует среднему размеру домов в США (максимальный средний размер домов) [90], выраженное в пикселах. Расстояние между отрезками определяется минимальным расстоянием между их концами. Для объединения отрезков необходимо, чтобы было выполнены следующие условия.

- 1) Углы прямых, определяемых отрезками, должны отличаться не более чем на допустимую погрешность. Допустимая погрешность определяется исходя из графика зависимости размера сегмента (Рис. 6) по длине наименьшего отрезка.
- 2) Выполняется проверка, насколько изменится угол объединенной прямой по сравнению с прямой, образованной большим из отрезков. Если этот угол превышает значение из графика на Рис. 6 для соответствующей длины, то объединение не выполняется.
- 3) Между ближайшими концами рассматриваемых отрезков выполняется проверка поля градиентов. Слияние выполняется,

если в рассматриваемой области находится градиент, имеющий направление, совпадающее с нормалью рассматриваемой прямой.

Отрезки, для которых выполнены эти три условия, объединяются по принципу, описанному в предыдущем разделе.

### 2.3. Результаты



Рис. 8. Исходное изображение.

Рассмотрим применение нескольких методов к изображению аэрофотосъемки и сравним их. На Рис. 8 приведено исходное изображение. На Рис. 9 показано изображение с линиями, найденными алгоритмом LSD,



на Рис. 10 – результат метода EdLines, а на Рис. 11 – результат работы предлагаемого подхода.



Рис. 9. Отрезки, обнаруженные методом LSD.

На Рис. 11 видно, что предложенный метод, по сравнению с алгоритмом LSD (Рис. 9) дает результат с гораздо большим шумом, но при этом успешно выделяет отрезки, которые были пропущены методом LSD. Ввиду особенностей представленного подхода, шум будет отбракован на

следующих его этапах, что компенсирует основной недостаток рассмотренного метода.



Рис. 10. Отрезки, обнаруженные методом EDLines.

По Рис. 10 и Табл. 2 можно сделать вывод, что на представленных данных метод EdLines нередко дублирует выделенные линии, несмотря на то, что общее количество отрезков, обнаруженных этим методом наименьшее (Табл. 1).





Рис. 11. Отрезки, найденные предложенным методом.

	LSD	EdLines	Предложенный метод
Время (сек)	2.68	1.56	3.58
Количество отрезков	2511	1467	4289
Ложные срабатывания (шт.)	471	213	1167

Табл. 1. Сравнение методов на скорость работы и на количество найденных отрезков.










LSD	EDLines	Предложенный метод
		
		
		

Табл. 2. Сравнение методов выделения отрезков на увеличенных участках изображения.



В Табл. 2 и Табл. 1 представлены сравнения результатов выделения отрезков предложенным методом, EdLines и LSD на одном изображении размером 1655 x 1698 (исходное изображение). Из этой таблицы видно, что предложенный метод медленнее, но находит большее количество отрезков (больше ложных срабатываний, но не настолько много, чтобы существенно отразиться на производительности). Так как дальнейшая обработка легко справляется с шумом и ложными срабатываниями, то предложенный метод является наиболее подходящим для предложенного подхода реконструкции моделей зданий.

### **3. Выделение областей**

Обычно большую часть аэрофотоснимков занимают поля, леса, огороды, дороги, которые не представляют особого интереса в рамках поставленной задачи. Использование методов реконструкции строений на подобных участках приводит к существенному понижению скорости работы всего рассматриваемого подхода. По этой причине в рамках разработанной технологии мной был предложен новый метод выделения областей интереса, который позволяет существенно сократить площадь, к которой применяются методы трехмерной реконструкции.

Важной характеристикой предлагаемого метода является высокая скорость работы и низкая доля пропущенных строений, не попадающих ни в одну из областей, возможно даже ценой высокой избыточности выделения. Использование существующих методов обнаружения строений по одному снимку [62-65] для решения данной задачи является необоснованным в связи с высокими затратами по времени. Поэтому в данной работе предложен новый подход, позволяющий достичь этих целей за счет анализа уже полученных отрезков и их взаимного положения. После этого на основе анализа производить такие операции как продление отрезка до пересечения, разбиение пересекающихся отрезков или объединение отрезков. Данный метод имеет низкую вычислительную сложность, но при этом позволит выделить области, представляющие интерес для дальнейшего анализа, и отсеять такие участки, как поля, леса и водные пространства.

Предложенный метод состоит из двух этапов. Первый этап представляет собой поиск пар отрезков, которые можно было бы скомпоновать тем или иным образом в зависимости от их взаимного

положения. В результате этого этапа мы получим ломаные, которые в общем случае не являются замкнутыми. Поскольку цель метода, описываемого в данной главе, – это получить области интереса, то возникает необходимость произвести замыкание ломаных контуров в замкнутые области, что и выполняется на втором этапе метода.

Рассмотрим описанные выше этапы предложенного метода подробнее.

### **3.1. Связывание соседних отрезков в контуры**

На данном шаге описываемого метода выделения областей производится попарное соединение отрезков друг с другом. Для формализации алгоритма связывания соседних отрезков в ломаные контуры, необходимо произвести классификацию возможного взаимного расположения пар отрезков. На Рис. 12 проиллюстрированы эти варианты:

- 1) отрезки лежат на одной прямой;
- 2) отрезки лежат на пересекающихся прямых;
- 3) отрезки лежат на параллельных прямых.

При этом важно, чтобы обработка проводилась именно в таком порядке для всего изображения, иначе возможно потерять некоторые элементы или выполнить лишние операции.

Случай, когда отрезки лежат на одной прямой, был рассмотрен в предыдущей главе при описании метода обнаружения границ, в частности, на этапе общего объединения, где производилось общее объединение без учета сегментов изображения.

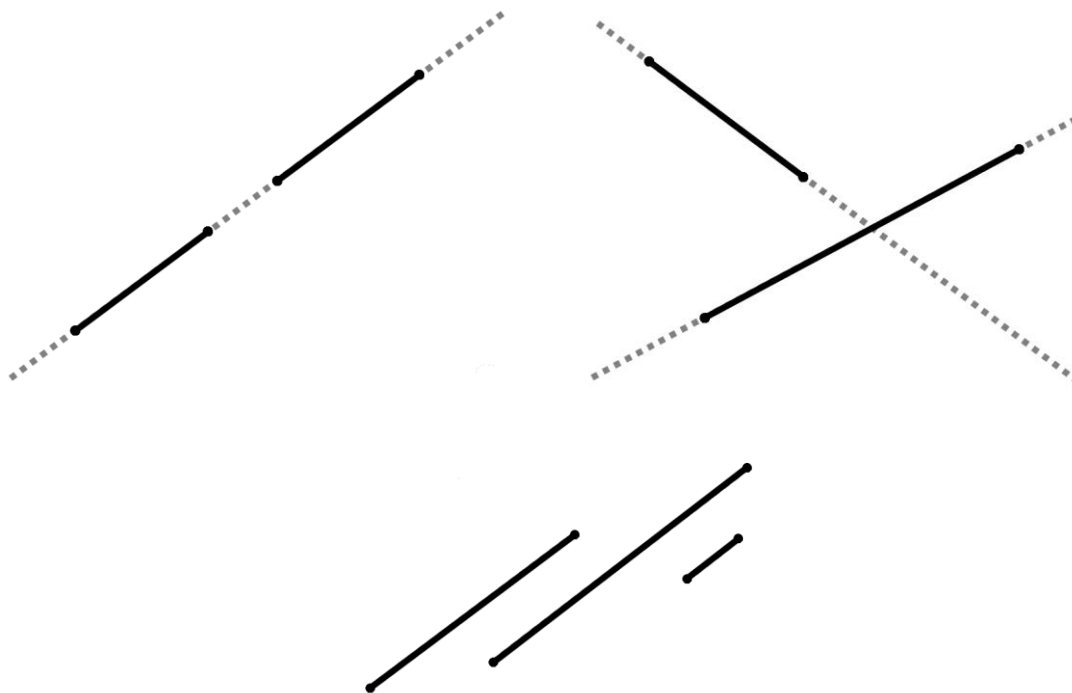


Рис. 12. Варианты взаимного положения отрезков: отрезки на одной прямой (сверху-слева), отрезки на пересекающихся прямых (сверху-справа), отрезки на параллельных прямых (снизу).

Если два отрезка находятся на пересекающихся прямых, то выполняется их продление до пересечения друг с другом. В этом случае возможны два варианта для каждого из отрезков. Первый вариант – точка пересечения находится на самом отрезке. Тогда отрезок разделяется на два отрезка. Второй вариант – точка пересечения находится на продолжении отрезка. Рассмотрение этого варианта идентично рассмотрению случая с отрезками на одной прямой за тем исключением, что не производится проверка угла наклона. То есть рассматривается расстояние продолжения отрезка до пересечения и происходит анализ градиента.

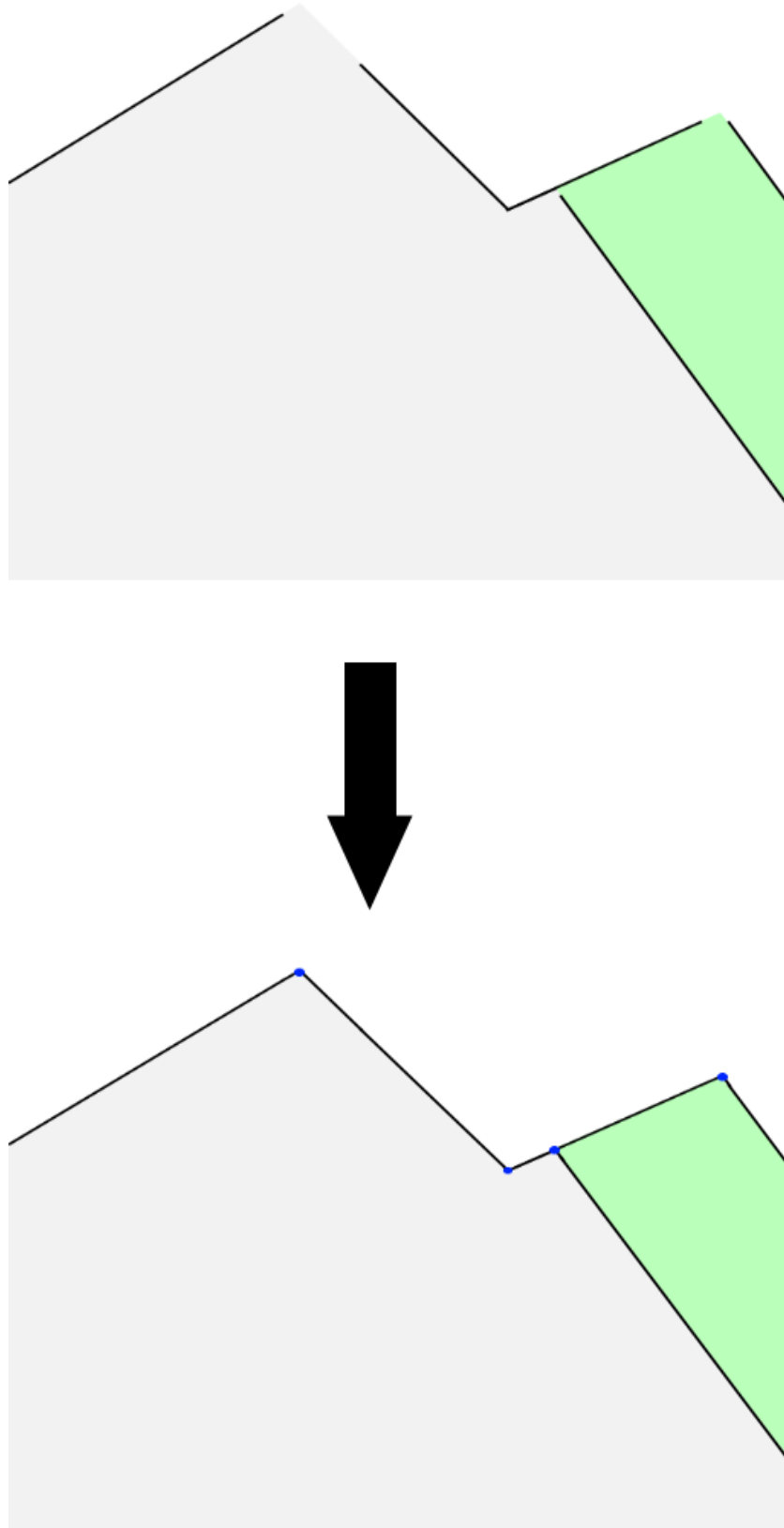


Рис. 13. Продление отрезков до соединения.

Если отрезки лежат на параллельных прямых (с учетом погрешности угла наклона), то рассмотрению подлежат следующие варианты: отрезки сдвинуты относительно друг друга, отрезки мало отличаются по длине и их концы на небольшом расстоянии друг от друга. В остальных случаях отрезки остаются как есть.

В первом варианте предлагается построить отрезок, соединяющий ближайшие концы параллельных отрезков. Далее по аналогии с предыдущими этапами проверяется градиент в области построенного отрезка и на этой основе принимается решение о создании нового соединительного отрезка.

Во втором варианте рассматривается расстояние между отрезками (перпендикуляр). Если оно незначительно (сильно зависит от качества снимков и порога, определяющего незначительность расстояния и задаваемого как входной параметр метода), и если один из отрезков является частью контура, то тогда контур расширяется до второго отрезка. В противном случае все остается как есть.

На Рис. 13 проиллюстрирован описанный подход. При применении данного подхода критичным является эффективный поиск отрезков, удовлетворяющих требованиям. Детали программной реализации этого алгоритма приведены в Приложении 1.

### **3.2. Замыкание областей**

На предыдущем этапе мы получили набор ломаных линий, которые могут быть и незамкнутыми. Так как здание может быть представлено только замкнутым контуром, то их необходимо замкнуть, чтобы выделить области, где здания находятся. Отрезок, построенный для замыкания ломаной, будем называть «замыкающим». На данном этапе ломаные линии могут получиться разветвленными и иметь больше двух концов. Поскольку

на данном этапе происходит фактически предобработка данных, то гораздо важнее не допустить пропуск объектов, чем выделение лишнего. Поэтому соединяться будут все концы, соединение которых не приведет к пересечению замыкающего отрезка с остальной ломаной. После этого проводится повторный анализ замыкающих отрезков.

Если некоторые замыкающие отрезки имеют пересечения друг с другом (согласно условию построения, они не могут пересекаться с другими частями ломаной), то они разбиваются точкой пересечения на разные отрезки, каждый из них тоже считается замыкающим. После этого удаляются те замыкающие отрезки, что оказались внутри контуров.

В результате данной операции мы получаем набор контуров, которые ограничивают зоны, в которых могут находиться здания. В дальнейшем вычислительно более сложный алгоритм будет обрабатывать только эти зоны.

### **3.3. Результаты**

На Рис. 14 приведены результаты выделения областей. Большая часть элементов, попадающих на границы снимка, не распознаются вследствие особенностей реализации. Это не влияет на результат, поскольку строения, не полностью попадающие на изображение гораздо сложнее реконструировать и точность реконструкции будет ниже, чем у полностью видимых строениях. Так как аэрофотосъемка обычно делается с перекрытием областей, то реконструкция строений на границах, скорее всего, будет выполнена по другому снимку.

Данная операция производится исключительно с целью уменьшения вычислительных затрат посредством уменьшения площади изображения, к которой в последующем применяется алгоритм реконструкции крыш по снимкам. Поэтому неточность выделения не влияет на качество конечного

результата. Стоит отметить, что внутри областей попали не только сами строения, но и тени от них. Это связано с тем, что представленный алгоритм работает только с геометрией, а тени имеют ту же геометрию, что и здания.



Рис. 14. Выделенные области.

Если не учитывать строения, не попавшие полностью на изображения, то ошибки типа false negative (ошибочно не найденные строения) составляет



порядка 5%-10%. Но поскольку этот метод используется как предобработка снимков для алгоритма, обрабатывающего снимки снятых с разных ракурсов, то это не так важно. Дело в том, что если на одном снимке строение не обнаружено, то оно будет обнаружено на другом, снятым с более удобного ракурса. Таким образом, точность обнаружения будет примерно такой же, как и в более затратном методе, таком, как метод перебора плоскостей [91]. При этом количество данных, обрабатываемых этим алгоритмом, будет значительно меньше.

На аэрофотоснимках данный алгоритм выделял для будущей обработки 20%-30% от общей площади для сельской местности и 50-70% для местности городского типа. Это позволяет рассчитывать на ускорение всего подхода порядка 30-60%, поскольку данные, полученные на всех этапах, кроме анализа полученных отрезков и соединения их в контуры, используются в алгоритме реконструкции зданий по набору аэрофотоснимков. Дополнительные вычислительные затраты малы, поскольку обработка векторных данных вычислительно проще, чем анализ растров.

## **4. Сопоставление областей**

Выделение областей, представляющих интерес для дальнейших операций, значительно уменьшают количество проводимых вычислений. Но возможно его сократить еще больше, если произвести сопоставление выделенных областей на разных снимках, чтобы ограничить количество сопоставлений примитивов, выделенных на изображении на следующих шагах подхода. Здесь также, как и при выделении областей важно, чтобы выигрыш от применения этой операции был больше, чем затраты на нее. Наиболее распространенные существующие методы такие, как SIFT [92] или ORB [93] не подходят как раз из-за большой вычислительной сложности и большого времени работы. Точность этих методов не оправдывает временных затрат на них. По этой причине возникла необходимость разработать новый метод, использующий особенности рассматриваемых данных и работающий быстрее существующих аналогов.

Для реализации метода, который удовлетворял бы поставленным требованиям, предлагается реализовать новый подход, опирающийся на упрощенное сопоставление крупных примитивов (отрезков) внутри областей с использованием априорной информации о том, что в рассматриваемых областях находятся строения.

### **4.1. Метод сопоставления областей**

Для поиска соответствующей области используется преобразование из координат одного снимка в геодезические, после чего из геодезических координат в координаты второго снимка [94]. Однако, поскольку нельзя восстановить высоту точки по одному рассматриваемому снимку, то представленные преобразования параметризуются. В результате, в общем случае точка на одном снимке переходит в луч (отрезок, если ограничить

высоту) на другом. Исходная точка может находиться где угодно на полученном луче или отрезке (Рис. 15) в зависимости от ее положения в трехмерном пространстве.

Таким образом, отрезки при переносе на другой снимок будут переходить в отрезки с концами на противоположных границах четырехугольника, образованного образами точек-концов исходного отрезка. На Рис. 16 представлена иллюстрация нахождения отображения отрезка с одного снимка на другой. Зеленым раскрашены элементы рассматриваемого снимка (камера, отрезок, изображение, линии проекции концов отрезков), красным – снимка на которой ведется отображение.

В данной работе предлагается использовать следующую особенность аэросъемки: все снимки сделаны примерно на одной высоте. Это позволяет сделать предположение, что горизонтальные линии (в трехмерном пространстве) существенно не изменяются от снимка к снимку, и единственное неизвестное при построении линии на другом снимке – ее высота в трехмерном пространстве (Рис. 17). Поэтому, если построить линию на втором снимке, предполагая, что она имеет нулевую высоту, то искомый отрезок будет параллелен построенному.

При идеальном совпадении отрезок должен заканчиваться у границ зоны сопоставления. По этой причине в качестве веса используется отношение длин сопоставляемого отрезка к рассматриваемому. Чем ближе значение веса к единице, тем более вероятно, что сопоставление верно.

Наконец, вычисляются веса для оценки соответствия областей на сопоставляемом снимке. Для каждой сопоставляемой области вес высчитывается как усредненная сумма весов отрезков, вычисленная на предыдущем этапе. Чем ближе вес к единице, тем более вероятно, что области коррелируют.

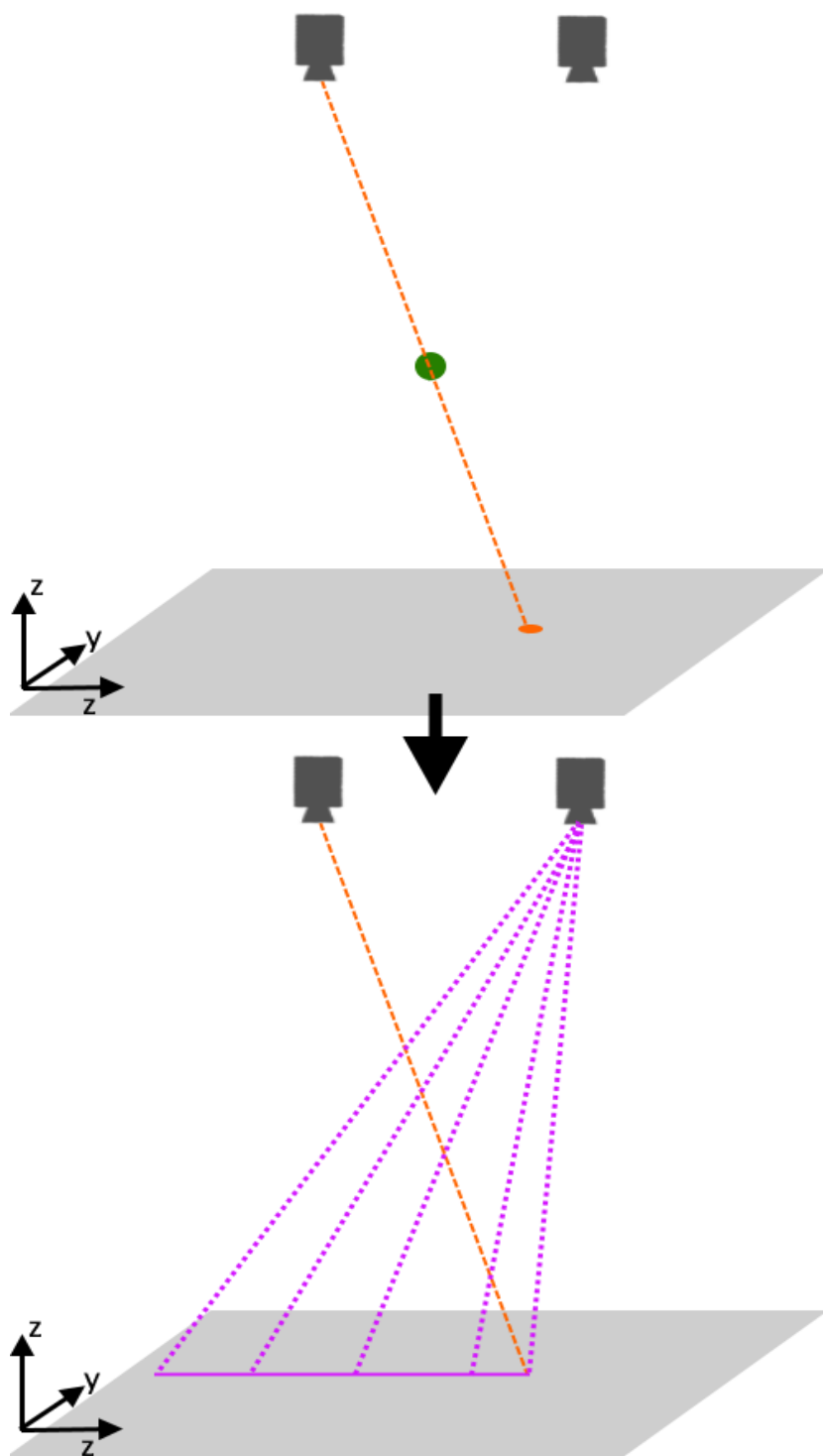


Рис. 15. Возможные положения точки на втором снимке в зависимости от ее высоты.

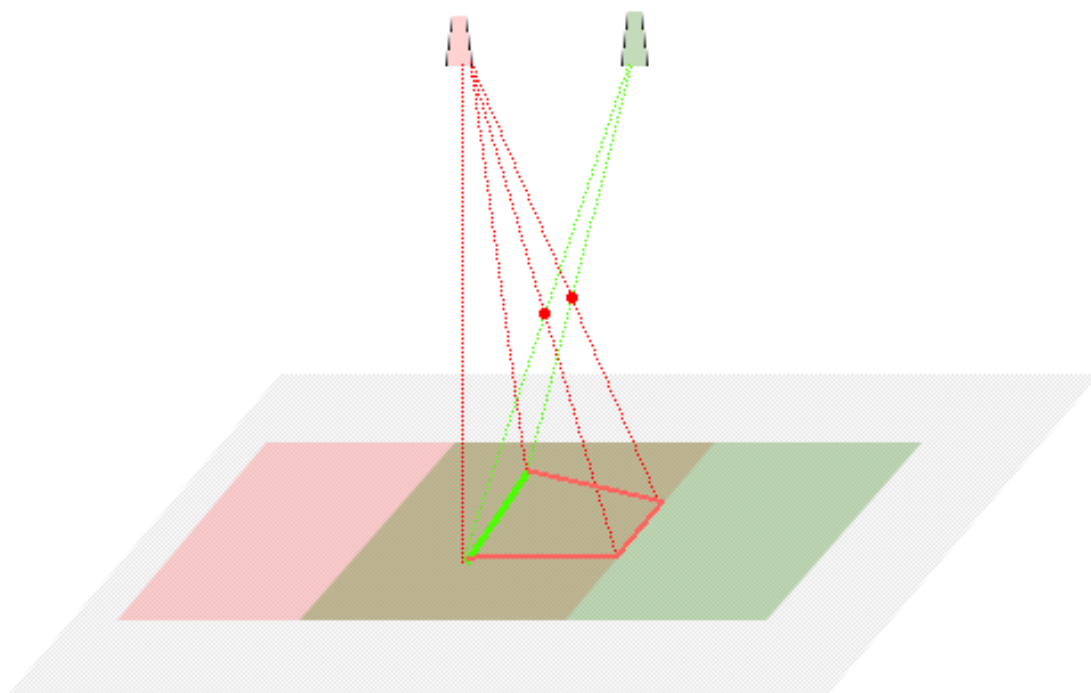


Рис. 16. Отображение отрезка на другой снимок

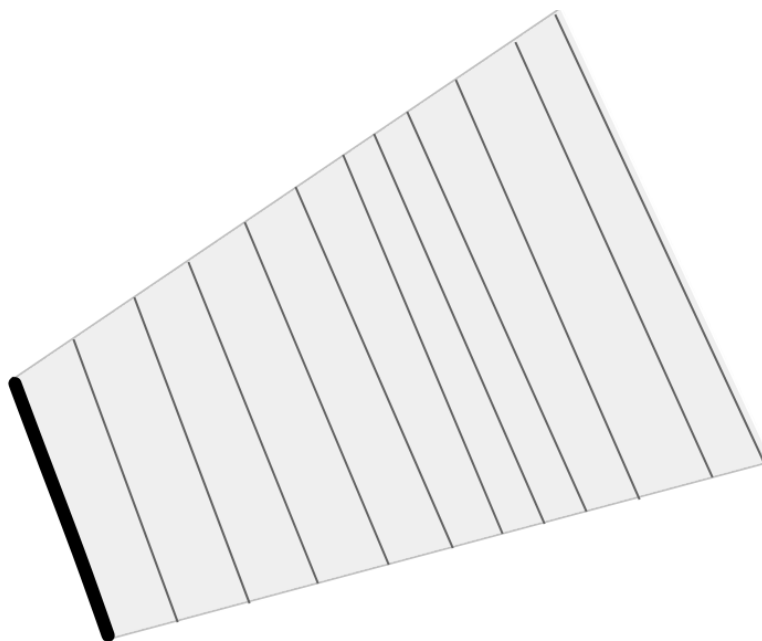


Рис. 17. Проекция отрезка на второе изображение и область поиска, соответствующая ему.

Однако, из-за того, что единственный метод при сопоставлении областей между снимками, – это геометрические операции, то есть вероятность того, что описанный выше метод даст некорректное сопоставление. По этой причине вместо того, чтобы выбрать одну область, сопоставляемые области сортируются по значениям веса. Далее используется верификационный алгоритм, описанный ниже.

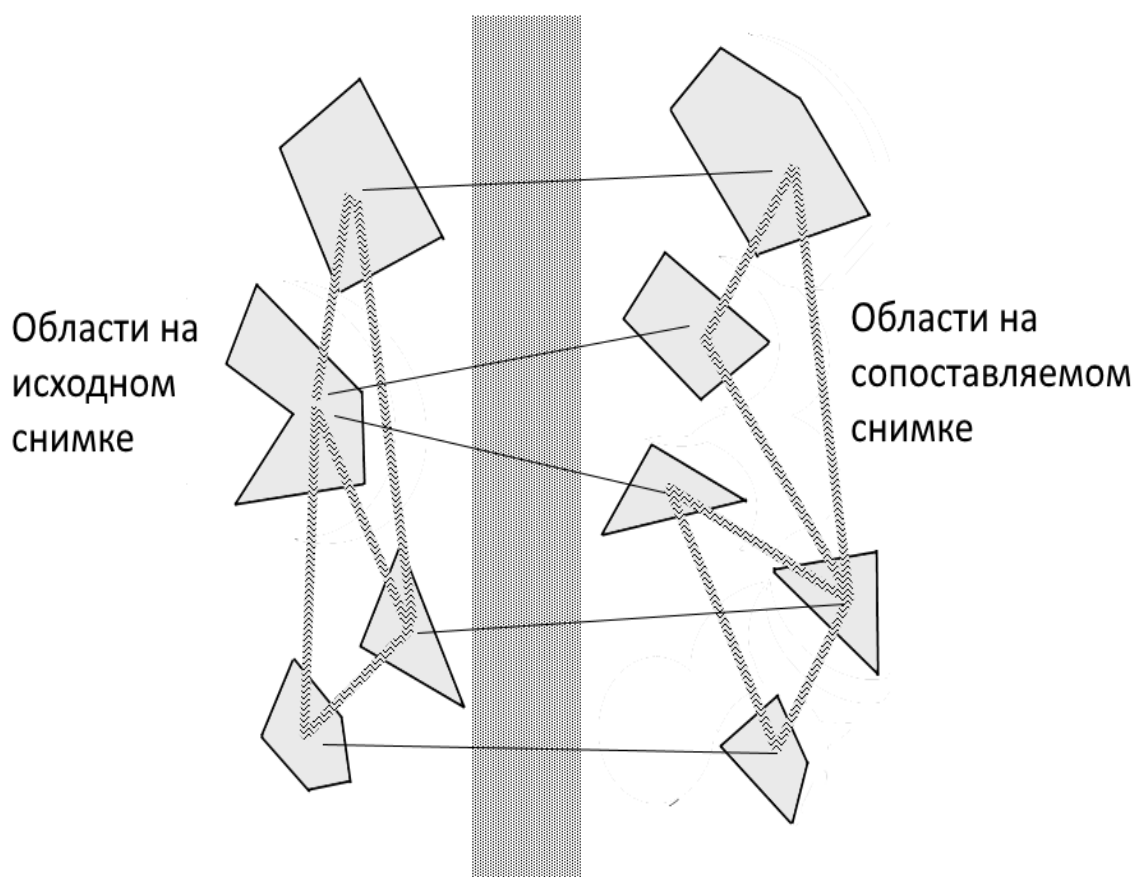


Рис. 18. Области и их соответствие на двух снимках.

Сначала рассчитываются расстояния между областями на исходном и сопоставляемом снимках. В качестве соответствующих областей выбираются те, что имеют наибольший вес. После этого расстояния между соответствующими областями сравниваются (Рис. 18). Далее на

сопоставляемом снимке находятся те области, расстояния между которыми больше остальных отличаются от расстояний связанных областей на исходном снимке. Тогда для исходной области ставится в соответствие следующая область в списке, полученном в разделе на предыдущем этапе.

Данная операция продолжается для всех связей до тех пор, пока не будет установлен минимум разницы расстояний.

## 4.2. Результаты

На Рис. 19 показаны результаты выделения областей и их сопоставления. Из данного примера легко заметить, что на нижнем снимке одна из областей не была ни с чем сопоставлена из-за того, что на верхнем изображении она включена в одну область со зданием.

Результаты, полученные на этапе сопоставления областей, представлены в Табл. 3. Сравнение производится с результатами метода SIFT [92], который является одним из наиболее распространенных подходов сопоставления изображений.

Из таблицы следует, что предложенный метод дает существенное ускорение за счет некоторой потери точности сопоставления областей между двумя снимками. Потеря точности больше для снимков плотной городской высокоэтажной застройки с множеством однотипных строений. В этом случае и ускорение не такое большое, как для сельской местности.

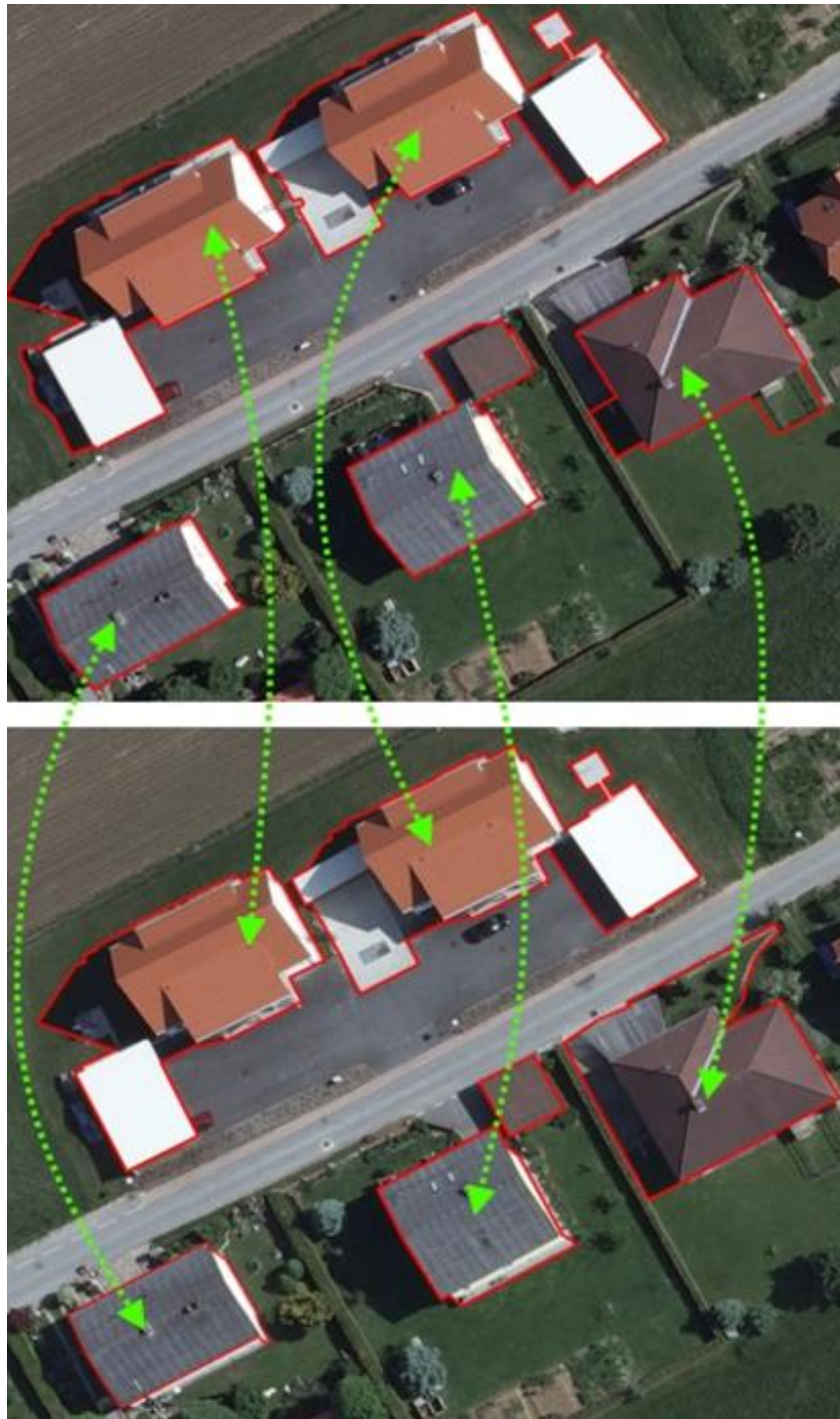


Рис. 19. Выделенные области и их сопоставление



		SIFT	Предложенный метод
Сельская местность (размер данных ~1GB )	Точность	93%	87%
	Скорость	247 сек.	43 сек.
Городская застройка (размер данных ~1GB)	Точность	85%	78%
	Скорость	20 мин.	9 мин.

Табл. 3. Сравнение методов сопоставления областей между двумя снимками.

Потеря точности частично компенсируется при сопоставлении множества пар снимков. Также снимки городской застройки менее востребованы, поэтому более важными являются больший выигрыш в скорости и меньшая потеря точности для снимков сельской местности.

## **5. Трехмерная реконструкция крыш**

После того, как области интереса были найдены и произведено их сопоставление между изображениями, необходимо провести трехмерную реконструкцию строений, находящихся в выделенных областях.

Поскольку в рамках данной работы интерес представляет обнаружение и реконструкция строений, для которых характерны линейные черты, то предлагается для решения поставленной задачи использовать подход, основанный на анализе, сопоставлении отрезков и построения плоскостей, образующих сегменты крыш строений. В связи с тем, что основной фокус данной работы направлен на аэрофотосъемку, которая ведется с большой высоты, то строения практически полностью определяются крышами, а их стены либо сложно, либо невозможно восстановить. По этой причине предлагается сделать упор именно на восстановление крыш строений, а из стены строить как вертикальные плоскости, отступающие от края крыш.

### **5.1. Метод реконструкции крыш**

Для решения поставленной задачи предлагается использовать подход, опирающийся на метод перебора плоскостей [91]. Общая схема данного метода следующая.

- 1) Сопоставление отрезков между снимками. Для каждого из отрезков ищется соответствующий ему отрезок на других изображениях.
- 2) Построение трехмерной модели сопоставленных отрезков за счет использования элементов внешнего ориентирования

изображений и применения обратных проективных преобразований (Рис. 20).

- 3) Для каждого из отрезков выполняется перебор полуплоскостей, содержащих этот отрезок. Это в свою очередь сводится к перебору углов поворота полуплоскости вокруг рассматриваемого отрезка, как это проиллюстрировано на Рис. 21.
- 4) Для каждой гипотезы для полуплоскости выбираются точки, находящиеся на ней (в трехмерном пространстве). Производится проецирование этих точек на каждое из изображений и их сопоставление. На основе результатов сопоставления делается вывод о правдоподобности этой полуплоскости (Рис. 22).
- 5) Для построенных полуплоскостей выполняется их ограничение другими отрезками, в них попадающих и пересечениями с другими полуплоскостями.

Данная операция дает точные результаты, но полный перебор плоскостей занимает слишком много времени (минуты), особенно если использовать маленький шаг дискретизации угла и является самым длительным этапом алгоритма (до 50% вычислительного времени). По этой причине в данной работе разработана новая модификация метода перебора плоскостей, работающая значительно быстрее.

Ускорение метода предлагается достичь за счет использования информации о пересечении отрезков, полученной на этапе поиска контуров. Поскольку два пересекающихся отрезка однозначно задают плоскость, то

вместо полного перебора плоскостей можно сразу ее построить по двум отрезкам. Полученную плоскость нельзя использовать как конечный результат по нескольким причинам. Во-первых, пересечение рассматриваемых отрезков может быть результатом ошибки их построения. Во-вторых, точность построения плоскости по двум отрезкам будет меньше, чем точность построения плоскости методом перебора. По этой причине найденную по двум отрезкам плоскость будем использовать как первое приближение. Далее метод перебора плоскостей будем применять, рассматривая только отклонения в окрестности угла, образующего построенную плоскость-приближение, и используя один из отрезков как ось.

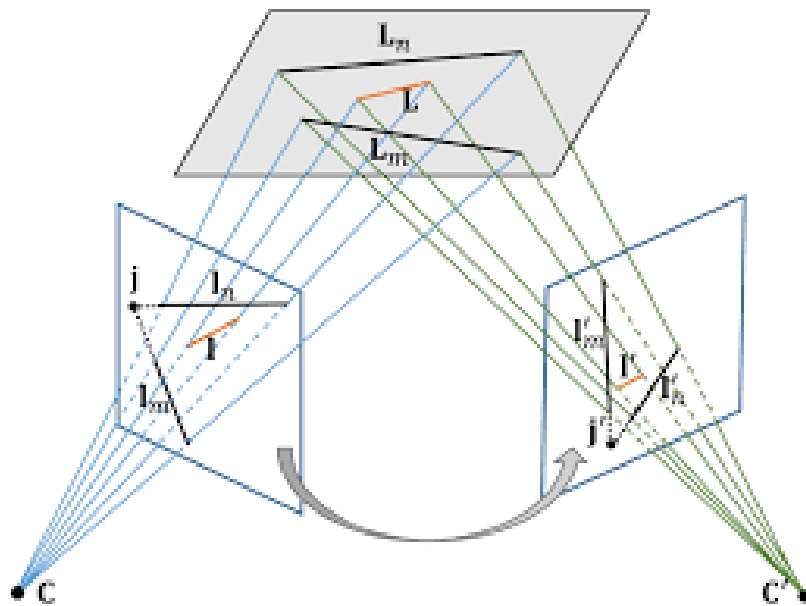


Рис. 20. Построение трехмерных отрезков  $L_n$   $L$   $L_m$  по соответствующим отрезкам  $l_n$   $l_m$  на одном снимке и  $l'_n$   $l'_m$  на другом

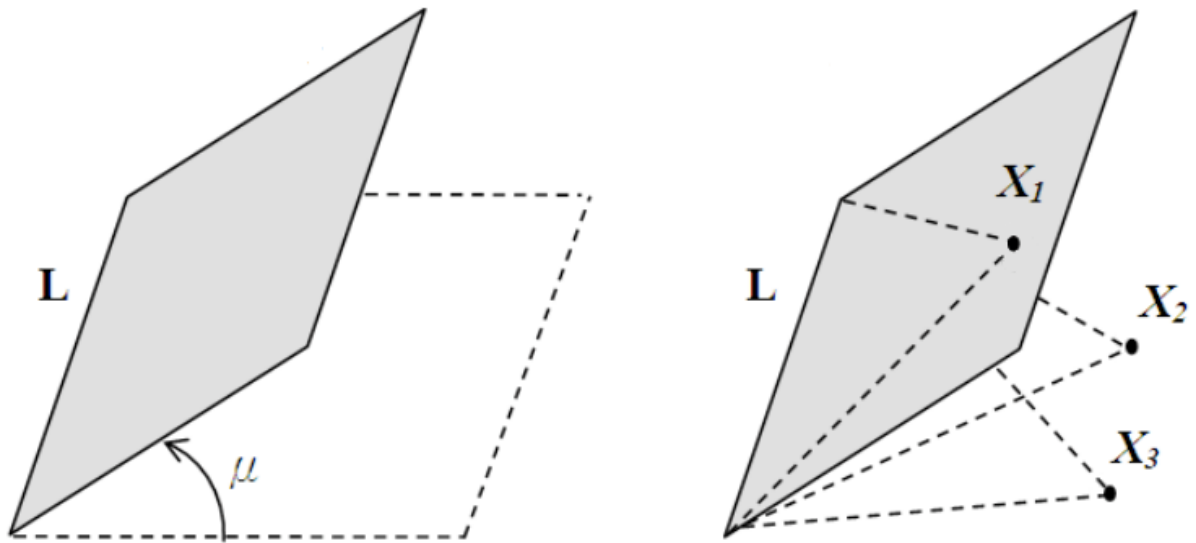


Рис. 21. Перебор полуплоскостей вокруг отрезка L. Точки  $X_1, X_2, X_3$  используются для проверки полуплоскости, на которой они лежат.

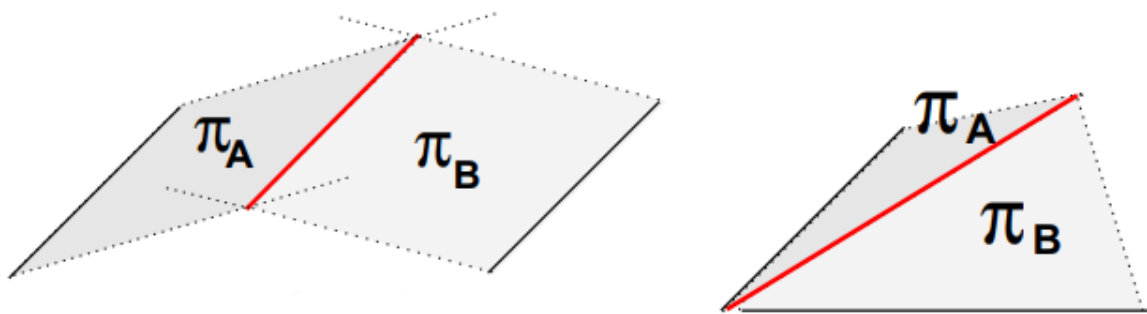


Рис. 22. Выполнение ограничения полуплоскостей на примере их взаимного пересечения.

При таком подходе можно получить уточненные данные и также выполнить контроль корректности, поскольку ни одна из перебираемых плоскостей не пройдет отбор по величине корреляции. Если у рассматриваемого отрезка есть несколько связанных с ним отрезков, то

описанная выше процедура применяется для каждого из них. Точность построения плоскостей можно еще больше увеличить, если применить следующий подход.

Рассматриваются отрезки, находящиеся на проекции плоскости на изображения. Из этих отрезков отфильтровываются короткие для исключения шума и деталей крыши (обычно достаточно фильтровать все отрезки короче трех метров в трехмерном пространстве, но конкретная величина зависит от качества и вида снимков). Далее отфильтровываются отрезки, расстояние от концов, которых до плоскости превышает погрешность при построении. Данная величина высчитывается на основе шага угла при переборе плоскостей и размера рассматриваемого отрезка: расстояние от одного из концов отрезка до плоскости при его отклонении от нее на угол, равный шагу перебора. Также не рассматриваются отрезки, находящиеся дальше удвоенного размера отрезка-оси.

Из оставшихся отрезков выбираются не более десяти, имеющих наименьшие отклонения. И по трехмерным координатам концов отрезков проводится линейная регрессия для уточнения плоскости. Данная операция сходится быстро благодаря точному начальному приближению, полученному на предыдущем этапе.

Заключительным этапом метода является финальное построение результата (крыш) по имеющимся плоскостям. Для этого в представленной работе предлагается использовать аналогичный этап метода перебора плоскостей без существенных изменений. Его суть заключается в ограничении полуплоскостей отрезками, лежащими на них, и выделении новых граней за счет пересечения полуплоскостей.

Также на этом этапе проходит фильтрация объектов по таким параметрам как размер (отсеиваются машины, гаражи), высота объектов и их протяженность (отсеиваются дороги и поля).

## 5.2. Результаты



Рис. 23. Исходный аэрофотоснимок.

К преимуществам приведенного алгоритма можно отнести субпиксельную точность построения результирующего объекта и высокую скорость работы. Это достигается за счет предварительной обработки, что

позволяет проводить на порядок меньше сопоставлений, которые имеют большую вычислительную сложность.

К недостаткам можно отнести пропускание мелких деталей на крышах, таких как, например, трубы. Однако для большинства случаев это не имеет принципиального значения и заменяется наложением текстуры на полученный объект (что является одним из направлений дальнейшего развития приведенной работы).

На Рис. 23 представлен один из изначальных снимков, по которому проводилась трехмерная реконструкция. На Рис. 24 показан результат, полученный описанным алгоритмом. По представленному примеру можно отметить несколько особенностей.

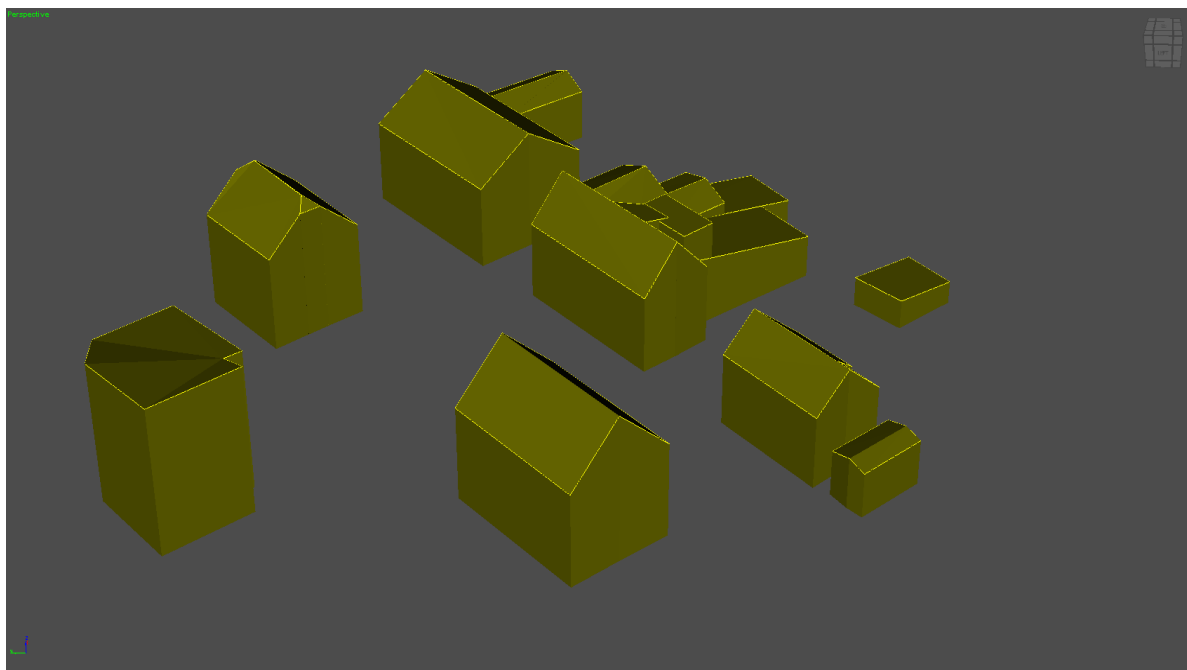


Рис. 24. Трехмерная модель строений, построенная из исходного снимка (Рис. 23) по представленному алгоритму.



Во-первых, представленный алгоритм позволяет воссоздать даже достаточно сложную геометрию. Примером является правая часть среднего дома на Рис. 24. Во-вторых, данный алгоритм не позволяет воссоздать такие элементы, как трубы, мансардные окна и прочие малоразмерные элементы.

В-третьих, при трехмерной реконструкции плоских крыш со сложной геометрией, как у дома слева на Рис. 24, возникают дефекты, связанные с построением нескольких плоскостей вместо одной, что, в свою очередь, вызвано неточностями при определении высоты. В результате вместо одной плоскости получилось несколько граней для одной крыши.

## 6. Результаты

Реализация представленного подхода проходила проверку на компьютере с параметрами, представленными в Табл. 4.

Процессор	AMD FX-8370
Частота	4.0 ГГц
Кол-во ядер	8
Оперативная память	2 x AMD DDR4
Объем	2 x 8 ГБ
Видеокарта	Radeon RX 470
Видеопамять	4ГБ
Жесткий диск	Samsung 860 EVO
Тип	SSD Sata 6 Гбит/с
Объем	500ГБ

Табл. 4. Параметры компьютера

Тестирование и оценка результатов предложенного метода проводилась на наборах данных (далее проектах), представленных в Табл. 5 и Табл. 6. Выбор проектов проводился на основе подбора широкого диапазона видов данных для наиболее широкого покрытия функциональности описанного подхода и получения более точных оценок при сравнении с другими алгоритмами, решающими схожую задачу. Стоит отметить, что проект UltraCamX\_part представляет из себя небольшой сегмент проекта UltraCamX [95]. Это было сделано для получения более комфортных для восприятия и сравнения данных, как визуальных, так и

численных (тех, где используются абсолютные величины, такие как количество найденных отрезков).

Название проекта	Кол-во снимков	Размер снимков (пикс)	Размер области (км <sup>2</sup> )	Размер пикселя (см)	Кол-во перекрытий	
					сред	макс
UltraCamX [95] (Рис. 25)	6	9420 x 14430	2,67	10,1	3	6
NewZealand (Рис. 26)	16	14336 x 14848	41,16	18,0	4	5
UAS_MINI (Рис. 27)	27	4912 x 3264	0,48	5,9	5	8
UltraCamX_part	6	700 x 800	0,006	10,1	6	6

Табл. 5. Используемые тестовые данные и их параметры.

Название проекта	Тип съемки	Кол-во строений	% областей с высокой застройкой
UltraCamX (Рис. 25)	Аэросъемка	81	~ 37%
NewZealand (Рис. 26)	Аэросъемка	593	~ 19%
UAS_MINI (Рис. 27)	Съемка БПЛА (UAS)	72	~ 89%
UltraCamX_part	Аэросъемка	9	~100%

Табл. 6. Используемые тестовые данные и их параметры.

Описанный в данной статье подход решает сразу две задачи: обнаружение строений и трехмерная реконструкция найденных строений. Ввиду особенностей работы представленного подхода, все объекты, построенные в процессе, классифицируются как строения. По этой причине данный метод можно рассматривать как метод поиска строений. Мало альтернативных подходов решают сразу обе эти задачи. По этой причине анализ полученных результатов и сравнения с существующими методами будут проходить по двум критериям:

- точность обнаружения строений
- точность трехмерной реконструкции



Рис. 25. Тестовые данные проекта UltraCamX со снимками, разложенными по ЭВО.

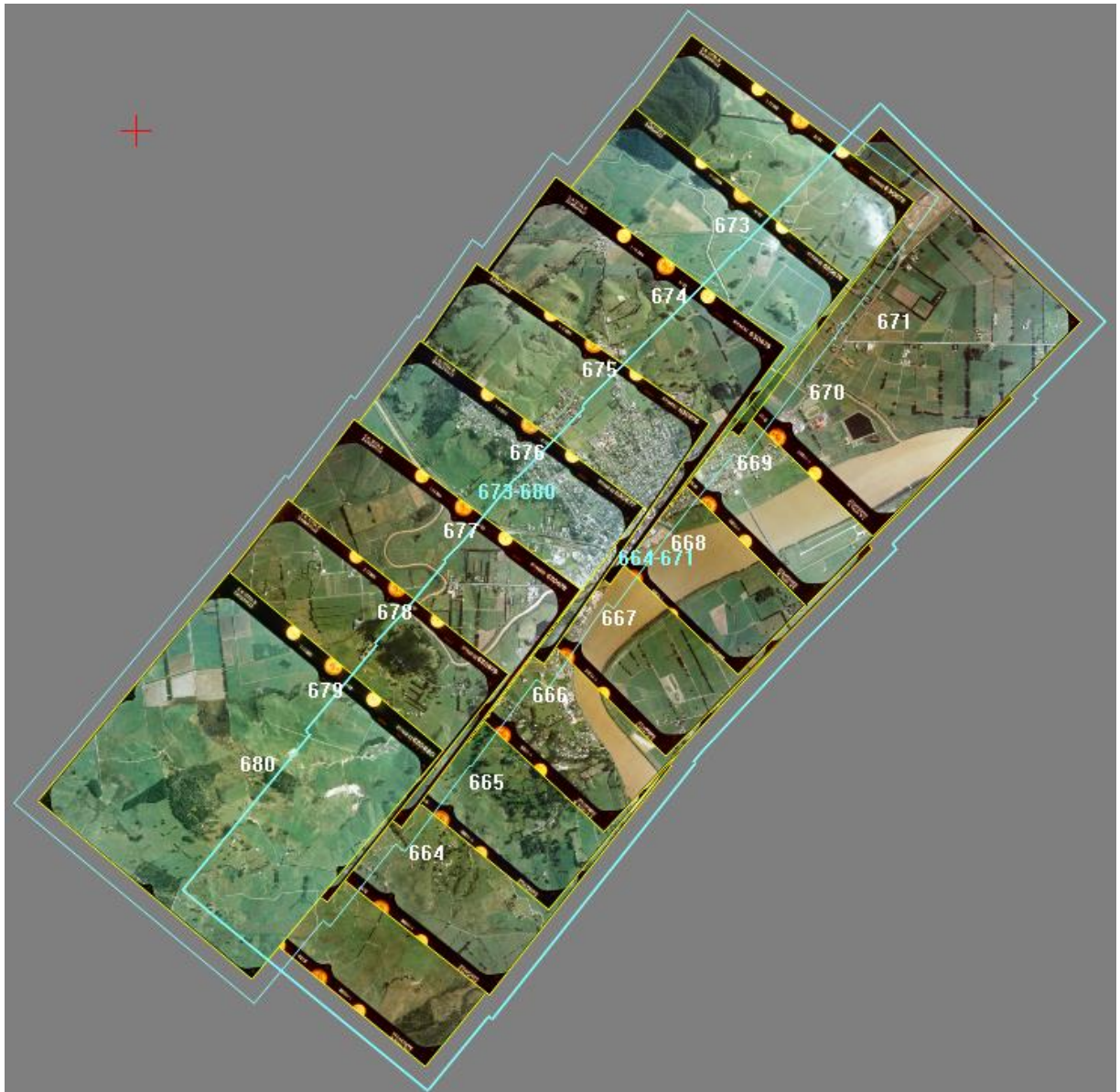


Рис. 26. Тестовые данные проекта NewZealand со снимками, разложенными по ЭВО.





Рис. 27. Тестовые данные проекта UAS\_MINI со снимками, разложенными по ЭВО.

### 6.1. Точность обнаружения

В данной работе предлагается использовать метрику F-мера [96] (также обозначаемая, как F1). Основными параметрами этой метрики являются точность (Precision, *Prec*) и полнота (Recall, *Rec*). Точность определяет насколько метод устойчив к ложным срабатываниям. Полнота определяет насколько рассматриваемый метод полно обнаруживает интересующие объекты, то есть устойчивость к пропуску целевых объектов. При этом только точности или полноты недостаточно для оценки метода, поэтому используется объединяющая метрика F1. Вычисление этих параметров, как и самой меры, приведено в формулах (2) — (4):

$$Prec = \frac{TP}{TP + FP} \quad (2)$$

$$Rec = \frac{TP}{TP + FN} \quad (3)$$

$$F_1 = 2 \times \frac{Prec \times Rec}{Prec + Rec} \quad (4)$$

где

- TP (true positive) – количество верных обнаружений;
- FP (false positives) – количество неверных обнаружений;
- FN (false negative) – количество необнаруженных объектов.

На Рис. 28 показаны результаты работы описанного подхода на каждом из представленных в Табл. 5 и Табл. 6 проектах. На основе этих данных, можно сделать вывод, что предложенный подход обладает высокой точностью и имеет очень маленькое количество ложных срабатываний. Ожидаемо, наилучшие результаты представленный подход дает для проекта UltraCamX, где самое высокое качество съемки.



Также можно заметить, что при обработке данных, снятыми беспилотными летательными аппаратами (проект UAS\_mini), значительно падает полнота. Это связано с тем, что съемка такого рода проектов ведется зачастую под наклоном, а не вертикально вниз (надир). Это и приводит к снижению качества обнаружения, хотя и незначительно.

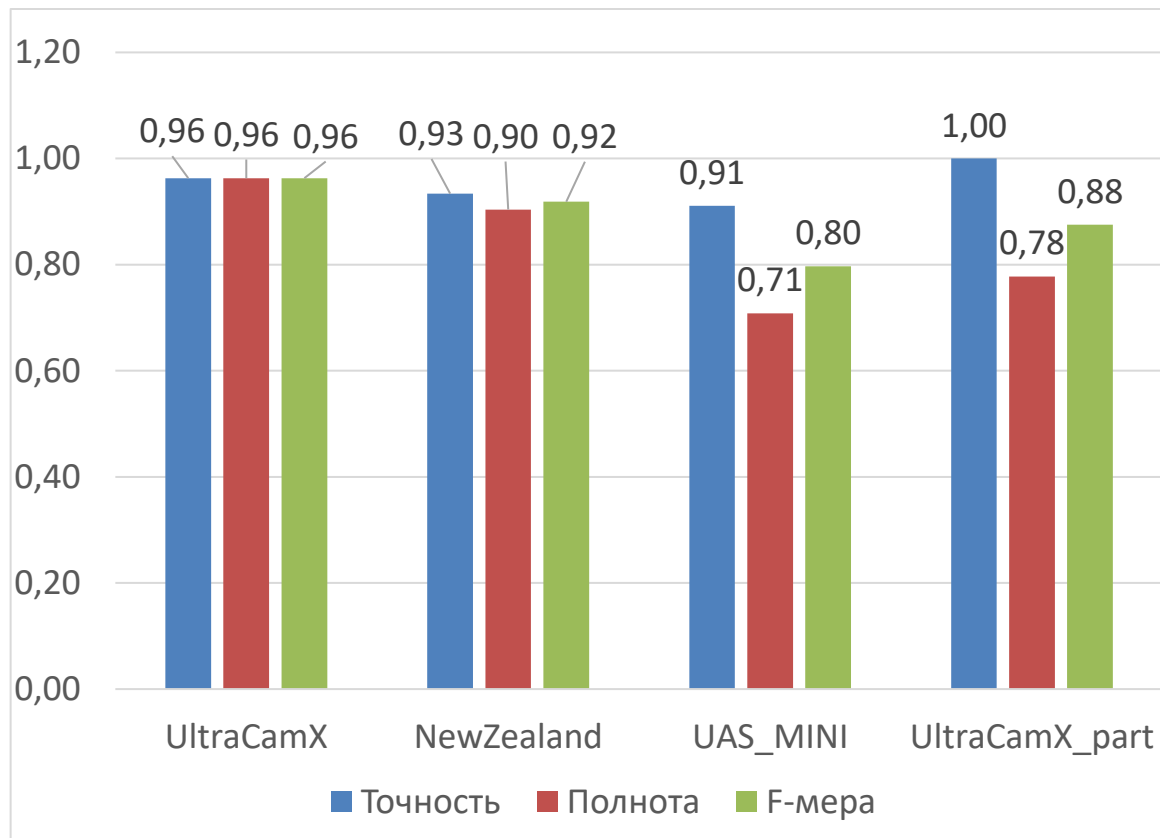


Рис. 28. Оценка результатов работы подхода.

Метрика	Метод Baillard'00	Метод Peng'16	Представленный метод
Точность	0.94	0.82	0.93
Полнота	0.96	0.86	0.90
F-мера (F <sub>1</sub> )	0.95	0.84	0.92

Табл. 7. Сравнение представленного метода с существующими по точности распознавания.

В таблице 7 представлено сравнение точности работы предлагаемого подхода с Baillard'00 [91] и Peng'16 [35] на примере проекта NewZealand.

Из Табл. 7 следует, что предложенный метод немного уступает по точности методу перебора плоскостей (Baillard'00), но по-прежнему превосходит метод, опирающийся на ортофото (Peng'16).

### 6.2. Точность трехмерной модели

На Рис. 29 представлены результаты работы рассматриваемого метода с точки зрения точности построения трехмерной модели. Для нормализации метрик все данные представлены в пикселях. Данные значения получаются путем приведения значений в метрической системе в пиксельные с использованием параметра «размер пикселя».

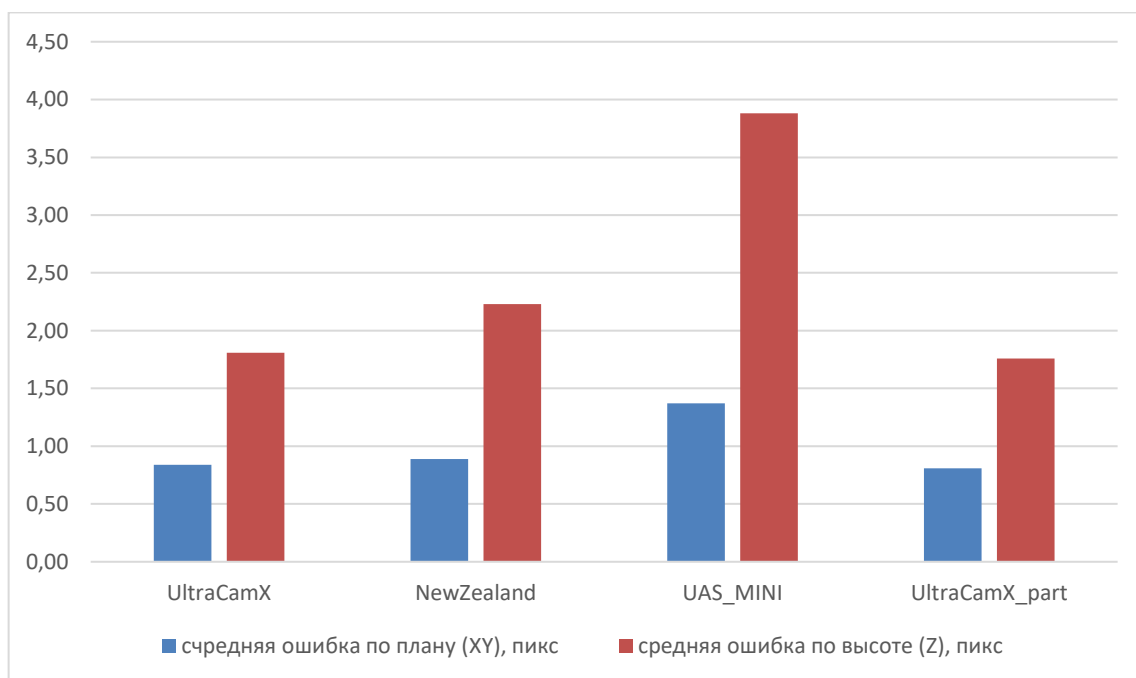


Рис. 29. Оценка результатов предложенного подхода.

На представленной диаграмме видно, что предложенный метод работает заметно хуже на проекте БПЛА (UAS\_mini). Это вызвано тем, что метод рассчитан и оптимизирован с расчетом на вертикальную (надирную) аэросъемку. Съемка с БПЛА не имеет строго вертикального направления, ведется с меньшей высоты и при помощи более дешевого оборудования и, как следствие, имеет менее качественный результат. Также при съемке с БПЛА может существенно меняться высота (на проекте UAS\_mini эти изменения не существенны, что является исключением из правил для БПЛА съемки), что противоречит допущениям разработанного подхода, таким как вертикальность съемки (надир) и постоянная высота съемки.

В Табл. 8 приведены сравнение результатов, полученных представленным методом, методом перебора плоскостей (Baillard'00) [91] и методом, основанном на использовании облака точек (Dahlke'15) [97]. Все методы, используемые для сравнения, были реализованы автором работы. Сравнение проводилось на проекте UltraCamX\_part. Для сравнения точности указанных подходов используется метрика, основанная на расчете ошибки автоматического построения относительно построенной вручную «эталонной» модели.

Метрика	Метод Baillard'00	Метод Dahlke'15	Представленный метод
Ошибка по плану (пикс)	0.89	0.68	0.81
Ошибка по высоте (пикс)	2.98	2.01	2.76
Время (мин)	86	197	43

Табл. 8. Сравнение представленного метода с существующими по точности построения и времени работы на проекте UltraCamX\_part.

Из данных, представленных в таблице 8, можно сделать вывод, что представленный метод работает быстрее, чем метод перебора плоскостей [91], при точности, не уступающей этому методу. Но метод, основанный на облаке точек превосходит по точности описанный подход. Данная разница объясняется компромиссом между точностью и скоростью работы. Помимо этого, метод на основе построения облака точек, восстанавливает не только строения, а всю сцену целиком, что также сказывается на скорости работы.

## Основные результаты

1. Предложен новый подход, решающий задачу выделения и построения трехмерных моделей строений по снимкам, полученным в результате надирной аэрофотосъемки в автоматическом режиме, за счет выделения и сопоставления отрезков на снимках. Представленный метод имеет высокую точность сопоставления и распознавания, не уступающую существующим методам. При этом предложенный подход имеет большую скорость работы.
2. Предложена оригинальная модификация метода LSD, основанная на разбиении изображения на области и последующем объединении отрезков. Данный подход позволяет увеличить количество распознанных отрезков на снимках за счет незначительного увеличения времени работы, потраченного на объединение отрезков.
3. Представлен оригинальный метод ускорения обработки аэрофотоснимков за счет выделения областей интереса и их сопоставления. Это позволяет отсеять до 80% изображения. При этом количество пропущенных строений не превышает 2%.
4. На основе предложенного подхода реализована программная компонента, используемая в индустриальном программном продукте Photomod, что приводит к повышению производительности операторов при решении задач, связанных с трехмерной реконструкцией населенных пунктов, зданий и строений.

## Приложение 1. Детали программной реализации.

В рамках данной работы была реализована программная компонента, составляющая порядка 15 000 строк кода, написанная на языке программирования C++. При этом использовался стандарт языка C++17 [98]. Этот язык был выбран, поскольку для реализации описанного подхода необходима высокая оптимизация выполняемых команд, так же, как и возможность структурировать код, ввиду сложности системы и особенностей внедрения в существующие программные продукты. C++ позволяет достичь обеих целей за счет использования объектно-ориентированного подхода и высокого уровня оптимизации кода компиляторами.

При реализации использовались следующие средства разработки и библиотеки.

**Visual Studio и MSBuild 2017** [99]. Данное средство разработки предоставляет обширный спектр инструментов разработки и отладки, но основная причина выбора этого инструмента – компилятор. Этот компилятор поддерживает стандарт C++17 [98]. Поскольку разработка данной работы ведется преимущественно для Windows, то данный компилятор является оптимальным решением. Единственная альтернатива – компилятор от Intel [100], но он при лучшей поддержке архитектуры процессоров Intel имеет худшую поддержку процессоров AMD.

При том, что в качестве основного инструмента сборки для реализации выбран MSBuild, при написании кода поддерживается совместимость с GCC для кроссплатформенности. Помимо этого, кроссплатформенность обеспечивают следующий набор инструментов и библиотек.

**CMake** [101]. Данный инструмент не имеет полноценных аналогов, представляет из себя кроссплатформенный скриптовый язык для генерации проектов необходимого типа для последующей сборки в зависимости от платформы, на которой происходит сборка и целевой платформы (Рис. 30).

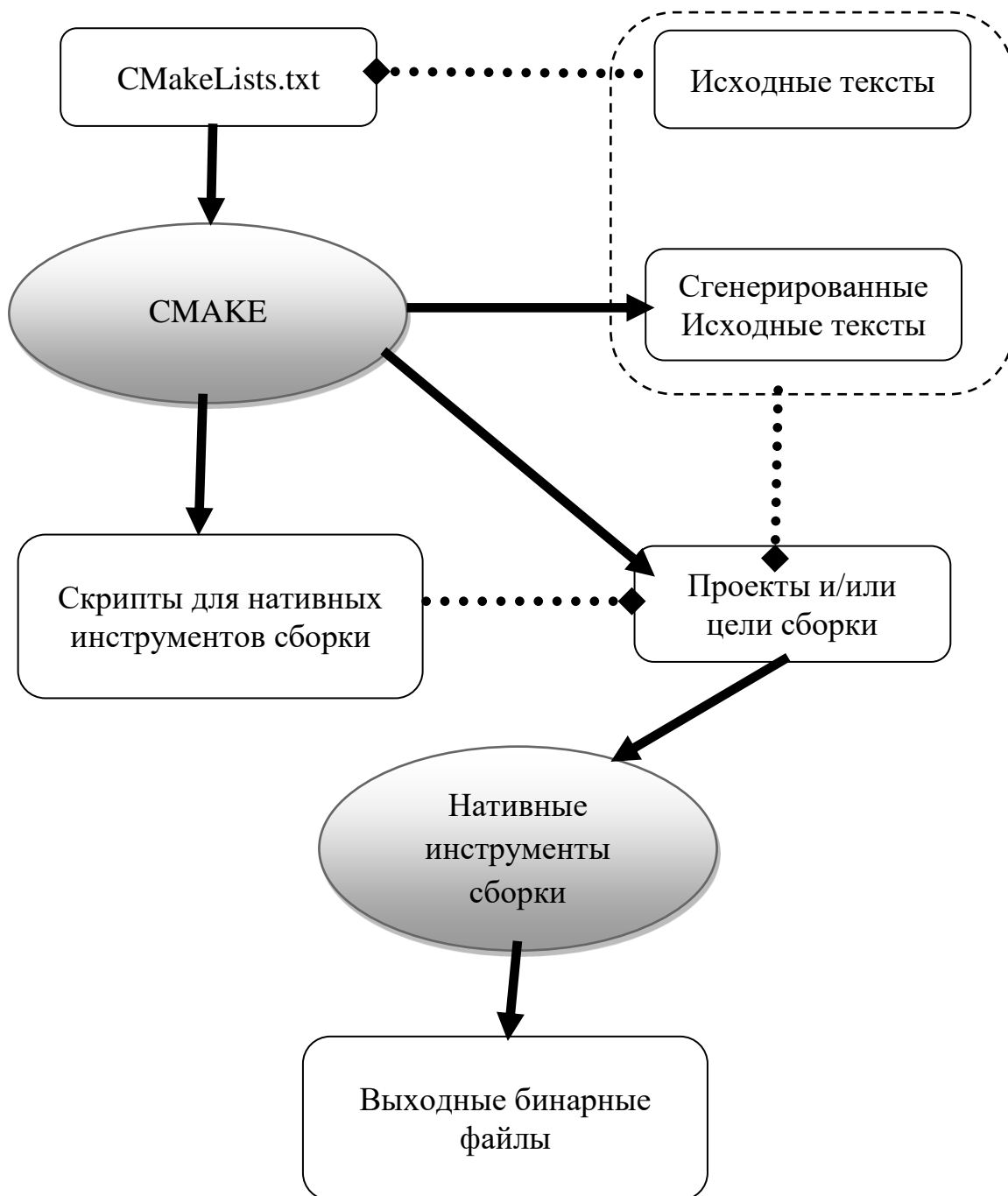


Рис. 30. Схема работы CMake.

Также CMake позволяет выставлять зависимости от внешних библиотек, не опираясь на конкретный механизм, используемый в тех или иных языках программирования или операционных системах.

**OpenMP** [102]. Библиотека, используемая для распараллеливания вычислений в рамках одного компьютера на несколько процессорных потоков. В данной работе используется ввиду своей простоты, эффективности и простоты подключения (большинство компиляторов позволяют использовать OpenMP путем добавления флага компиляции).

К недостатку OpenMP относительно других средств распараллеливания является возможность распараллеливать только на центральном процессоре, в то время как MPI [103-105] и OpenCL [106, 107] также позволяют распараллеливать на кластерах и графическом процессоре. Но так как создаваемый программный продукт, в который происходило внедрение результатов диссертационной работы и сравнения их с существующими методами, не предполагается распараллеливать на графическом процессоре или кластере, то OpenMP был достаточен.

**OpenCV** [108]. Библиотека алгоритмов компьютерного зрения, обработки изображений и численных алгоритмов общего назначения с открытым кодом. На данный момент самая распространенная библиотека в данной области. В данной работе она была использована преимущественно для реализации чужих существующих методов, чтобы провести сравнения с разработанным подходом. Аналогичные методы, используемые при реализации своего подхода, были либо написаны автором с оптимизацией под конкретную задачу, либо реализованы в рамках проприетарных библиотек с закрытым кодом.

**OpenBLAS** [109]. Библиотека алгоритмов линейной алгебры. В данной работе из многих реализаций интерфейса BLAS [110] был выбран



OpenBLAS из-за высокой оптимизации и ориентированности на производительность. Кроме того, в отличие от MKL [111], OpenBLAS является кроссплатформенным.

**QT** [112]. Кроссплатформенный фреймворк для разработки программного обеспечения на языке программирования C++. Данная библиотека используется автором для создания интерфейса и кроссплатформенной реализации таких системно-зависимых операций как работа с файлами, взаимодействие с ресурсами компьютера и сетевое взаимодействие. Они позволяют полностью отказаться от системных вызовов при разработке приложения. Единственный кроссплатформенный аналог – wxWidgets [113] – существенно проигрывает QT по стабильности, скорости работы и удобству интерфейса.

**GDAL** [114]. Библиотека для чтения и записи растровых и векторных гео-пространственных форматов данных (изображения, матрицы высот и тд). Используется в данной работе для поддержки чтения и записи различных форматов снимков и изображений.

В процессе реализации наибольшие трудности вызвали следующие подзадачи: работа с изображениями большого размера (2GB и более) и работа с отрезками.

Первая проблема связана с тем, что невозможно держать все изображения в памяти, кроме того, с этими изображениями необходимо проводить преобразования (например, фильтры шумов и контрастности, получения изображения в градациях серого). Это привело к необходимости реализовать структуру классов для работы с сегментами изображения. Диаграмма некоторых классов для этой структуры представлена на Рис. 31.

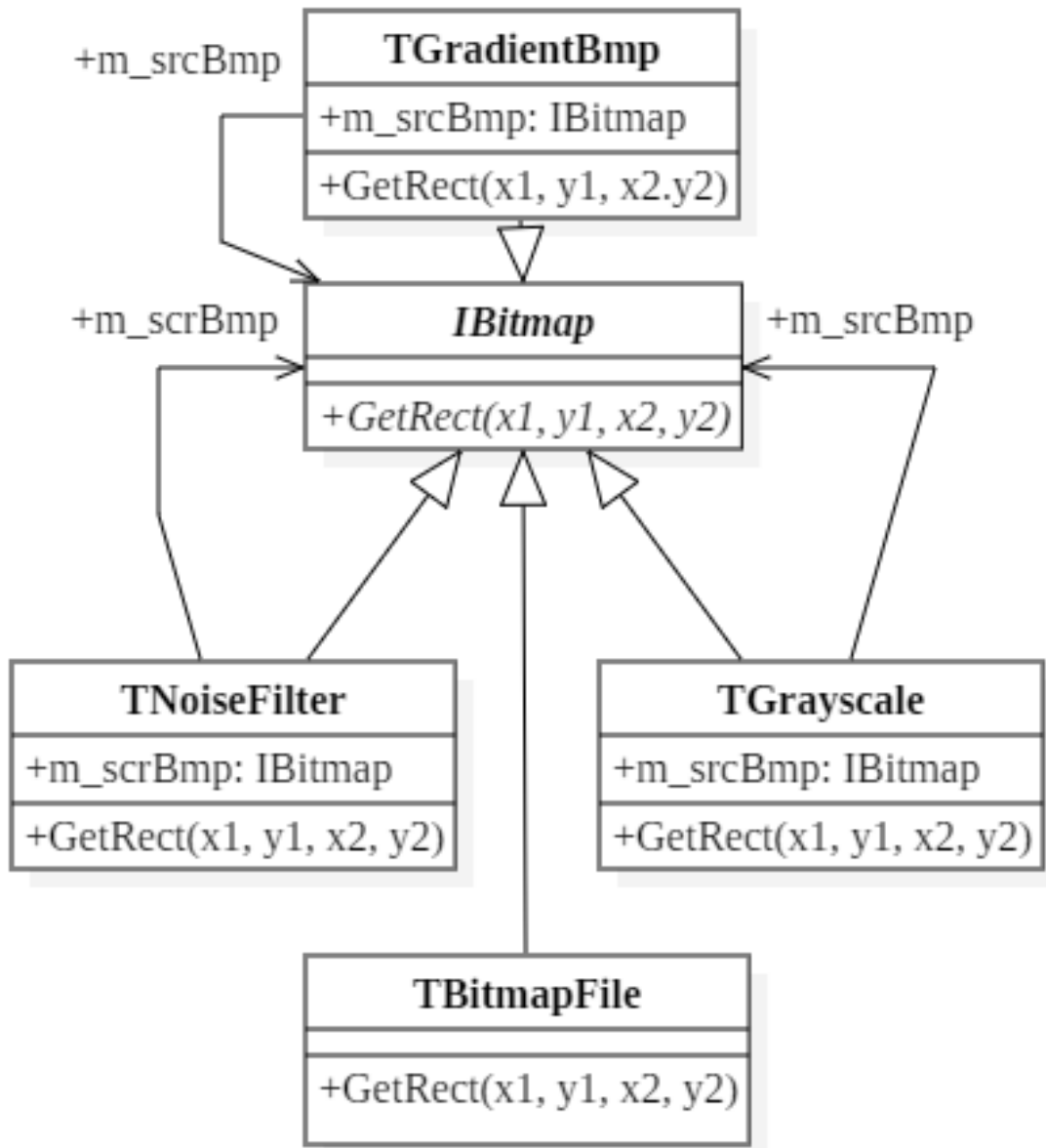


Рис. 31. Диаграмма классов для операций с изображениями.

Главной особенностью приведенной структуры является виртуальная функция `GetRect`, предоставляющая доступ к произвольному прямоугольнику на изображении. Использование виртуальности позволяет абстрагироваться от того, как именно устроено изображение. Это позволяет в процессе исследования легко подменять преобразования в исследуемых

алгоритмах, передавая им ту или иную реализацию интерфейса `IBitmap`. Данный интерфейс хранит только базовые параметры изображения, такие как размеры, наличие альфа-канала, информация о пирамиде, количество каналов, битность этих каналов, выравнивание, способ представления пикселя, а также предоставляет методы доступа к данным полям.

В дальнейшем эта же особенность позволила получать прямоугольники изображения не по пиксельным, а по геодезическим координатам с заданной аппроксимацией (или уровнем пирамиды, если исходные данные поддерживают ее).

Реализованная структура также позволяет использовать гибкие «цепочки» преобразований изображения. В качестве исходного изображения (`m_srcBmp`) передается указатель на класс-наследник `IBitmap`, который в свою очередь может быть либо другим преобразованием, либо классом для чтения из файла. Каждый из классов-наследников реализует метод `GetRect(...)` для получения массива данных, находящихся в заданном прямоугольнике. Данная структура позволяет производить обработку изображений по сегментам, что значительно снижает затраты оперативной памяти. Согласно представленной схеме загрузка и преобразования изображения будут проводиться не поэтапно (сначала загрузка всего изображения, потом поочередное применение ко всему изображению методов), а по запросу. Иными словами, загрузка сегмента изображения из файла и все необходимые преобразования этого сегмента будут производиться тогда, когда будет запрошено получение результата какого-либо метода (к примеру, получение граней на снимке).

На Рис. 32 представлена упрощенная диаграмма вызовов, иллюстрирующая работу этой схемы. В данном примере метод `CreateImage` создает последовательно классы `TBitmapFile`, `TGrayscale`, `TGradient`, `TEdges`, каждый из которых агрегирует предыдущий. `TBitmapFile` занимается

чтением изображения из файла и предоставляет доступ к сегментам этого изображения в виде растра. Класс TGrayscale использует в качестве исходных данных агрегированный им наследник интерфейса IBitmap, в данном примере – это TBitmapFile. Аналогично с остальными классами в цепочке.

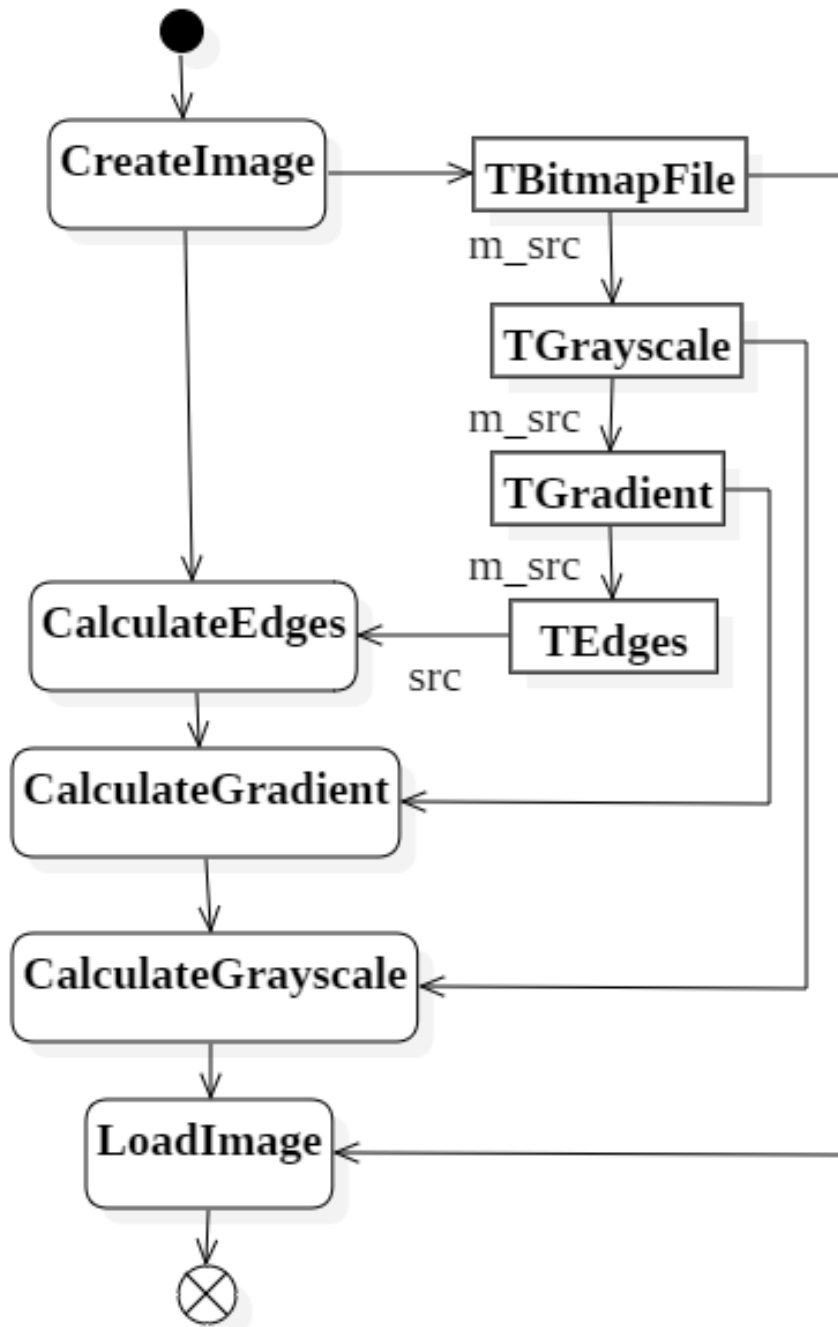


Рис. 32. Блок-схема вызова «цепочки» преобразований изображения.

При вызове функции CreateEdges (от заданного сегмента изображения) вызывается метод GetRect у класса TEdges, реализующего обнаружение границ на снимке, который в свою очередь для получения исходных растровых данных вызывает GetRect у класса TGradient, занимающимся вычислением градиентов на изображении, и далее по «цепочке» до вызова TBitmapFile::GetRect, который непосредственно читает сегмент растра из файла на жестком диске.

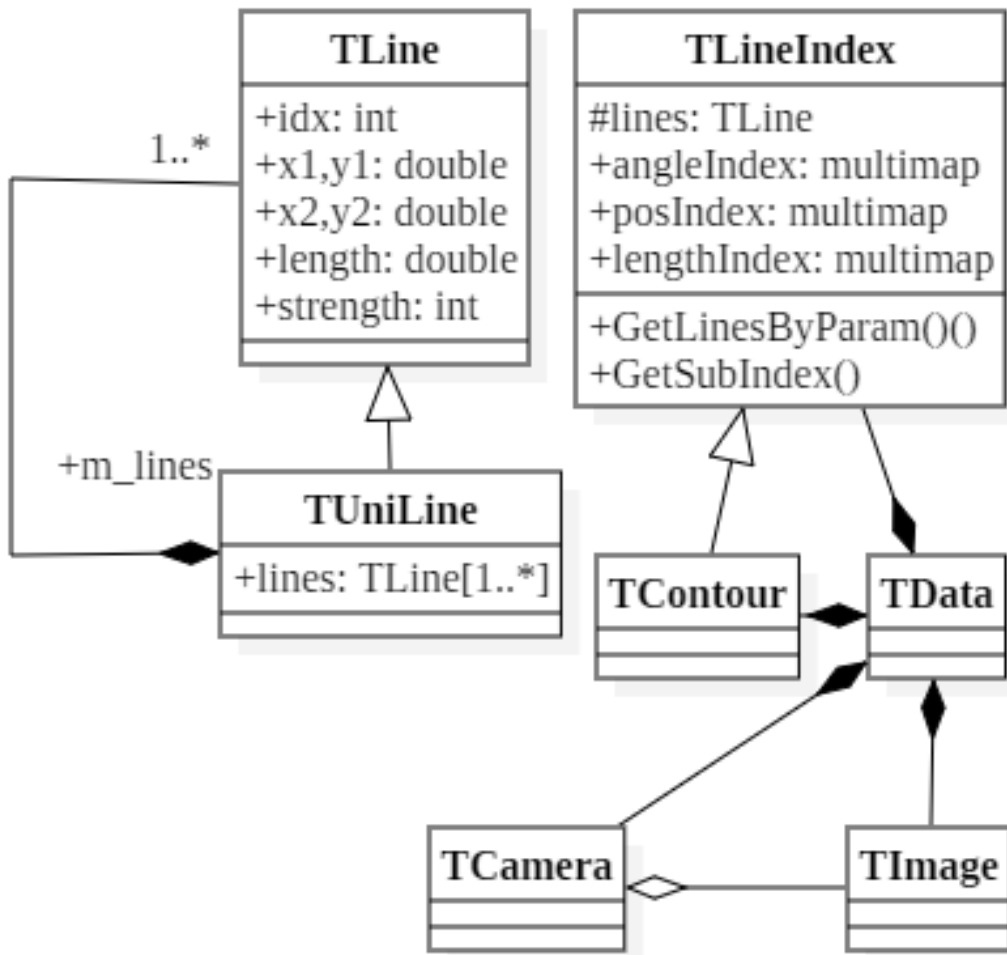


Рис. 33. Диаграмма классов работы с данными.

Вторая проблема связана с операциями над отрезками. Практически на всех этапах предложенного подхода используется поиск отрезков по заданным критериям (угол поворота, положение). Кроме того, часть отрезков являются объединением из других отрезков (они получаются на этапе объединения отрезков). Учитывая, что в основном используются операции по чтению, то было принято решение использовать следующую структуру классов для работы с данными (Рис. 33).

Данная структура позволяет осуществлять быстрый поиск и доступ к отрезкам по заданным углу наклона, положению или длине за счет использования класса `TLineIndex`, осуществляющего данный функционал. Это достигается следующим образом. Каждый из параметров (угол наклона, положение, размер) разбивается на непересекающиеся сегменты в которые складываются индексы отрезков, относящиеся к этому сегменту. В дальнейшем при поиске отрезков выполняется в первую очередь вычисление, в каких сегментах находятся интересующие отрезки, что является быстрой операцией, так как она не включает в себя перебор. После этого выполняется фильтрация отрезков, попавших в эти сегменты посредством перебора.

## Список литературы

1. Bulatitskiy, D.I., Buyval, A.K. and Gavrilencov, M.A., 2019. Building Recognition in Air and Satellite Photos. Proceedings of the 29th International Conference on Computer Graphics and Vision. Bryansk, Russia, September 23-26, 2019, P.173-177.
2. Алексапольский Н.М. Фотограмметрия: Часть 1. Москва: Геодезиздат, 1956. С 412.
3. Берлянт А.М. Образ пространства: карта и информация. Москва: Мысль, 1986. С 240.
4. Burns J. B., Hanson A. R., Riseman E. M. Extracting straight lines // IEEE transactions on pattern analysis and machine intelligence, Vol. 4, 1986. P. 425-455.
5. ЦФС PHOTOMOD // Ракурс (<https://racurs.ru/program-products/tsfs-photomod>). Дата обращения 15.03.2020.
6. Новоторцев Л.В., Волобой А.Г. Нахождение отрезков на аэрофотоснимках с использованием преобразования Хафа // Труды XXIV международной конференции по компьютерной графике и машинному зрению «Графикон». Ростов-на-Дону. 2014. С. 170—172.
7. Новоторцев Л. В., Волобой А. Г. Улучшенный алгоритм обнаружения отрезков для реконструкции строений по аэрофотоснимкам / Приволжский научный журнал, № 4, 2014. С. 49-53.

8. Новоторцев Л.В., Волобой А.Г. Нахождение областей, содержащих здания, на аэрофотоснимках // Труды XXV международной конференции по компьютерной графике и машинному зрению «Графикон». Протвино. 2015. С. 85—88.
9. Новоторцев Л.В., Волобой А.Г. Модификация фильтра Кэнни в рамках задачи обнаружения строений // Новые информационные технологии в автоматизированных системах: материалы девятнадцатого научно-практического семинара - М.: ИПМ им. М.В.Келдыша, 21 апреля 2016. С. 22-25.
10. Новоторцев Л.В., Волобой А.Г. Ускорение алгоритма распознавания зданий по нескольким аэрофотоснимкам за счет предобработки изображений // Препринты ИПМ им. М.В. Келдыша. 2016. № 35. С 12.
11. Новоторцев Л.В., Волобой А.Г. Постобработка Прямых и Ломаных, Полученных Методом Хафа // Труды XXVI международной конференции по компьютерной графике и машинному зрению «Графикон». Нижний Новгород. 2016. С. 100—104.
12. Новоторцев Л.В., Волобой А.Г. Обнаружение и сопоставление областей, содержащих строения, на аэрофотоснимках // Труды XXVI международной конференции по компьютерной графике и машинному зрению «Графикон». Нижний Новгород. 2016. С. 404—408.
13. Новоторцев Л.В., Волобой А.Г. Автоматическое распознавание зданий на аэрофотоснимках // Труды 16-ой международной конференции CAD/CAM/PDM-2016 "Системы проектирования, технологической подготовки производства и управления этапами



жизненного цикла промышленного продукта", Москва, ИПУ им. В.А. Трапезникова РАН, октябрь 17-19, 2016. С. 68-72.

14. Новоторцев Л.В., Волобой А.Г. Выделение отрезков на аэроснимках в рамках задачи обнаружения строений // Труды XXVII международной конференции по компьютерной графике и машинному зрению «Графикон». Пермь. 2017. С. 228-232.
15. Новоторцев Л.В., Волобой А.Г. Сопоставление областей на аэрофотоснимках в задаче распознавания строений // Новые информационные технологии в автоматизированных системах: материалы девятнадцатого научно-практического семинара - М.: ИПМ им. М.В.Келдыша, 20 апреля 2018. С. 17-23.
16. Новоторцев Л.В., Волобой А.Г. Ускорение метода перебора плоскостей при трехмерной реконструкции строений по аэроснимкам // Труды XXVIII международной конференции по компьютерной графике и машинному зрению «Графикон». Томск. 2018. С. 226-229.
17. Новоторцев Л.В., Волобой А.Г. Ускорение алгоритма распознавания зданий по нескольким аэрофотоснимкам за счет предобработки изображений // Препринты ИПМ им. М.В. Келдыша. 2018. № 207. С 12.
18. Новоторцев ЛВ, Волобой АГ. Обнаружение и трехмерная реконструкция строений по аэрофотоснимкам. Программирование, № 6, 2019. С. 13-22.  
English translation:  
L.V. Novotortsev, A.G. Voloboy. Detection and 3D Reconstruction of Buildings from Aerial Images // Programming and Computer

Software, vol. 45, No. 6, 2019, pp. 311-318,

DOI:10.1134/S0361768819060069

19. Новоторцев Л.В., Игнатенко А.В. Тональная компрессия изображений широкого динамического диапазона, использующая бинокулярное зрение человека // Новые информационные технологии в автоматизированных системах: материалы семнадцатого научно-практического семинара - М.: ИПМ им. М.В.Келдыша, 2014. С. 176-184.
20. Novotortsev, L., Voloboy, A.: Automated Detection of Buildings on Aero Images // 23rd International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision WSCG'2015, Poster paper proceedings. Czech Republic, Plzen. 2015. P. 23-28.
21. Huertas A., Nevatia R. Detecting buildings in aerial images // Computer Vision, Graphics, and Image Processing, 41(2), 1988. P. 131-152.
22. Roux M., McKeown D. M. Feature matching for building extraction from multiple views // IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR) 1994. P. 46-53.
23. Chaudhuri D. et al. Automatic building detection from high-resolution satellite images based on morphology and internal gray variance // IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 9(5), 2016. P. 1767-1779.
24. Cohen J. P. et al. Rapid building detection using machine learning // Applied Intelligence, 45(2), 2016. P. 443-457.
25. Lemmens M. J., Deijkers H., Looman P. A. Building detection by fusing airborne laser-altimeter DEMs and 2D digital maps //

- International Archives of Photogrammetry and Remote Sensing, 32(3), 1997. P. 42-50.
26. Fradkin M., Maitre H., Roux M. Building detection from multiple aerial images in dense urban areas // Computer vision and image understanding, 82(3), 2001. P. 181-207.
  27. Wang Q. et al. A hierarchical connection graph algorithm for gable-roof detection in aerial image // IEEE Geoscience and Remote Sensing Letters, 8(1), 2011. P. 177-181.
  28. Hofmann, Alexandra D., Hans-Gerd Maas, and André Streilein. Knowledge-based building detection based on laser scanner data and topographic map information // International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences 34(3/A), 2002. P. 169-174.
  29. Tarsha-Kurdi F. et al. Hough-transform and extended ransac algorithms for automatic detection of 3d building roof planes from lidar data // Proceedings of the ISPRS Workshop on Laser Scanning, 36, 2007. P. 407-412.
  30. Akçay H. G., Aksoy S. Building detection using directional spatial constraints // Geoscience and Remote Sensing Symposium (IGARSS), 2010. P. 1932-1935.
  31. Sidike P. et al. Automatic building change detection through adaptive local textural features and sequential background removal // Geoscience and Remote Sensing Symposium (IGARSS), 2016. P. 2857-2860.

32. Liu W., Prinnet V. Building detection from high-resolution satellite image using probability model // International Geoscience and Remote Sensing Symposium, 6, 2005. P. 3888.
33. Song Z. et al. Building roof detection from a single high-resolution satellite image in dense urban area // Proc. ISPRS Congress, 2008. P. 271-277.
34. Лидары // Портал Laser-Portal. ([http://laser-portal.ru/content\\_449](http://laser-portal.ru/content_449)).  
Дата посещения: 08.09.2017.
35. Peng D., Zhang Y. Building Change Detection by Combining LiDAR Data and Ortho Image // ISPRS-International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 2016. P. 669-676.
36. Ramiya A. M., Nidamanuri R. R., Krishnan R. Segmentation based building detection approach from LiDAR point cloud // The Egyptian Journal of Remote Sensing and Space Science. – 2016.
37. Rottensteiner F. et al. Building detection by Dempster-Shafer fusion of LIDAR data and multispectral aerial imagery // IEEE Proceedings of the 17th International Conference on Pattern Recognition, Vol. 2, 2004. P. 339-342.
38. Shirowzhan S., Trinder J. Building Classification from Lidar Data for Spatio-temporal Assessment of 3D Urban Developments // International High- Performance Built Environment Conference. – 2017.
39. Tseng Y. H., Hung H. C. Extraction of building boundary lines from airborne LIDAR point clouds // ISPRS International Archives of the

Photogrammetry, Remote Sensing & Spatial Information Sciences,  
Vol. 41, 2016. P. 957-962.

40. Verma V., Kumar R., Hsu S. 3d building detection and modeling from aerial LIDAR data // IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 2, 2006. P. 2213-2220.
41. USGS EROS Archive - Products Overview // U.S. Geological Survey (<https://www.usgs.gov/centers/eros/science/usgs-eros-archive-products-overview>). Дата обращения 0.1.01.2019.
42. Новоселов В.Г. Использование модуля PHOTOMOD Mosaic для построения качественных ортофотопланов // Internet Archive WaybackMachine (<https://web.archive.org/web/20170603184024/http://www.racurs.ru/?page=145>). Дата обращения 11.02.2020.
43. ESRI (<https://www.esri.com/news/arcuser/1001/standup.html>). Дата обращения 15.03.2020.
44. TrueOrtho современная картографическая основа ГИС крупных городов // портал ГИС-Ассоциации (<http://gisa.ru/48774.html>). Дата обращения: 08.09.2017.
45. Baltsavias E., Mason S., Stallmann D. Use of DTMs/DSMs and orthoimages to support building extraction // Automatic extraction of man-made objects from aerial and space images, 1995. P. 199-210.
46. Berthod M. et al. High-resolution stereo for the detection of buildings // Automatic Extraction of Man-Made Objects from Aerial and Space Images, 1995. P. 135-144.

47. Brunn A., Weidner U. Extracting buildings from digital surface models // International Archives of Photogrammetry and Remote Sensing, 32(3) sect. 4W2, 1997. P. 27-34.
48. Chai D. A Probabilistic Framework for Building Extraction from Airborne Color Image and DSM // IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing. Vol. PP (99), 2016. P. 1-12.
49. Cord M., Paparoditis N., Jordan M. Dense, reliable and depth discontinuity preserving DEM computation from HRV urban stereopairs // International Archives of Photogrammetry and Remote Sensing, 32, 1997. P. 49-56.
50. Girard S. et al. Building detection from high-resolution color images // Remote Sensing – International Society for Optics and Photonics, 1998. P. 278-289.
51. Dornaika F. et al. Building detection from orthophotos using a machine learning approach: An empirical study on image segmentation and descriptors // Expert Systems with Applications, Vol. 58, 2016. P. 130-142.
52. Lu Y. H., Trinder J. C., Kubik K. Automatic building detection using the Dempster-Shafer algorithm // Photogrammetric Engineering & Remote Sensing, Vol. 72(4), 2006. P. 395-403.
53. Wegner J. D. et al. Building detection from one orthophoto and high-resolution InSAR data using conditional random fields // IEEE Journal of selected topics in applied Earth Observations and Remote Sensing, Vol. 4(1), 2011. P. 83-91.

54. Habib A. F., Kim E. M., Kim C. J. New methodologies for true orthophoto generation //Photogrammetric Engineering & Remote Sensing, Vol. 73(1), 2007. P. 25-36.
55. Rau J. Y., Chen N. Y., Chen L. C. True orthophoto generation of built-up areas using multi-view images //Photogrammetric Engineering and Remote Sensing, Vol. 68(6), 2002. P. 581-588.
56. Schickier W., Thorpe A. Operational procedure for automatic true orthophoto generation // International Archives of Photogrammetry and Remote Sensing, Vol. 32, 1998. P. 527-532.
57. Zhou G., Wang Y. Occlusion detection for urban aerial true orthoimage generation // IEEE International Geoscience and Remote Sensing Symposium (IGARSS), 2016. P. 3009-3012.
58. Hu D. T. et al. An approach of DSM generation from multi-view images acquired by UAVs // The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. 41(B7), 2016. P. 821-825.
59. Li Z., Gruen A. Automatic DSM generation from linear array imagery data // International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. 35(B3), 2004. P. 128-133.
60. Pepe M., Prezioso G. Two approaches for dense DSM generation from aerial digital oblique camera system // Proceedings of the 2nd International Conference on Geographical Information Systems Theory, Applications and Management, Rome, Italy, 2016. P. 26-27.
61. Zhang L., Gruen A. Multi-image matching for DSM generation from IKONOS imagery // ISPRS Journal of Photogrammetry and Remote Sensing, Vol. 60(3), 2006. P. 195-211.

62. Ghaffarian S., Ghaffarian S. Automatic building detection based on supervised classification using high resolution Google Earth images // The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 40(3), 2014. P. 101-106.
63. Jin X., Davis C. H. Automated building extraction from high-resolution satellite imagery in urban areas using structural, contextual, and spectral information // EURASIP Journal on Applied Signal Processing, 2005, P. 2196-2206.
64. Ok A. O. Automated Extraction of Buildings and Roads in a Graph Partitioning Framework // ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. (II-3/W3), 2013. P. 79-84.
65. Singhal S., Radhika S. Automatic Detection of Buildings from Aerial Images Using Color Invariant Features and Canny Edge Detection // International Journal of Engineering Trends and Technology (IJETT), 11(8), 2014. P. 393-396.
66. Collins R. T. A space-sweep approach to true multi-image matching // Computer Society Conference on Computer Vision and Pattern Recognition, 1996. P. 358-363.
67. Qin R., Tian J., Reinartz P. Spatiotemporal inferences for use in building detection using series of very-high-resolution space-borne stereo images // International Journal of Remote Sensing, 2016. Vol. 37(15). P. 3455-3476.
68. Remondino F. et al. State of the art in high density image matching // The Photogrammetric Record, Vol.29(146), 2014. P. 144-166.



69. Stentoumis C. et al. On accurate dense stereo-matching using a local adaptive multi-cost approach // ISPRS Journal of Photogrammetry and Remote Sensing, Vol. 91, 2014. P. 29-49.
70. Hough P. V. C. Method and Means for Recognizing Complex Patterns // U.S. Patent 3069654, Dec. 18, 1962.
71. I. Sobel and G. Feldman. A  $3 \times 3$  Isotropic Gradient Operator for Image Processing. R. Duda and P. Hart (Eds.) // Pattern Classification and Scene Analysis, 1973. P. 271–272.
72. Canny J. A computational approach to edge detection // IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 6, 1986. P. 679-698.
73. Dollár P., Zitnick C. L. Fast edge detection using structured forests // Pattern Analysis and Machine Intelligence, IEEE Transactions, Vol. 37(8), 2015. P. 1558-1570.
74. Perona P., Malik J. Scale-space and edge detection using anisotropic diffusion // IEEE Transactions on pattern analysis and machine intelligence, Vol. 12(7), 1990. P. 629-639.
75. Xie S., Tu Z. Holistically-nested edge detection // Proceedings of the IEEE International Conference on Computer Vision. – 2015. – P. 1395-1403.
76. Bao P., Zhang D., Wu X. Canny edge detection enhancement by scale multiplication // IEEE transactions on pattern analysis and machine intelligence, Vol. 27(9), 2005. P. 1485-1490.
77. Deng C. X., Wang G. B., Yang X. R. Image edge detection algorithm based on improved canny operator // International Conference on

- Wavelet Analysis and Pattern Recognition (ICWAPR), 2013. P. 168-172.
78. Er-Sen L. et al. An adaptive edge-detection method based on the canny operator // International Conference on Environmental Science and Informatio, Vol. 1, 2009. P. 465-469.
  79. Rong W. et al. An improved CANNY edge detection algorithm // IEEE International Conference on Mechatronics and Automation (ICMA), 2014. P. 577-582.
  80. Li H., Lavin M. A., Le Master R. J. Fast Hough transform: A hierarchical approach // Computer Vision, Graphics, and Image Processing, 36(2), 1986. P. 139-161.
  81. Du, S., van Wyk, B. J., Tu, C., & Zhang, X. An improved Hough transform neighborhood map for straight-line segments // IEEE Transactions on Image Processing, 19(3), 2010. P. 573-585.
  82. O'Rourke J. Dynamically Quantized Spaces for Focusing the Hough Transform // IJCAI, 1981. P. 24-28.
  83. Matas, J. and Galambos, C. and Kittler, J.V., Robust Detection of Lines Using the Progressive Probabilistic Hough Transform // CVIU 78(1), 2000. P. 119-137.
  84. Von Gioi R. G. et al. LSD: A fast line segment detector with a false detection control // IEEE transactions on pattern analysis and machine intelligence, Vol. 32(4), 2010. P. 722-732.
  85. Von Gioi R. G. et al. LSD: a line segment detector // Image Processing on Line, Vol. 2, 2012. P. 35-55.

86. Desolneux A., Moisan L., Morel J. M. Computational gestalts and perception thresholds // *Journal of Physiology-Paris*, Vol. 97(2), 2003. P. 311-324.
87. Desolneux A., Moisan L., Morel J. M. *The Helmholtz Principle // From Gestalt Theory to Image Analysis: A Probabilistic Approach*, 2008. P. 31-45.
88. Akinlar C., Topal C. EDLines: A real-time line segment detector with a false detection control // *Pattern Recognition Letters*, Vol. 32(13), 2011. P. 1633-1642.
89. Binary descriptors for lines extracted from an image // Документация по OpenCV. ([https://docs.opencv.org/trunk/dc/ddd/group\\_\\_line\\_\\_descriptor.html](https://docs.opencv.org/trunk/dc/ddd/group__line__descriptor.html)). Дата обращения: 05.06.2018.
90. How big is a house? Average house size by country // Shrink That Footprint (<http://shrinkthatfootprint.com/how-big-is-a-house>). Дата обращения 16.03.2020.
91. Baillard C., Zisserman A. A plane-sweep strategy for the 3D reconstruction of buildings from multiple images // *International Archives of Photogrammetry and Remote Sensing*, Vol. 33(B2) part 2, 2000. P. 56-62.
92. Ng, P. C., & Henikoff, S. (2003). SIFT: Predicting amino acid changes that affect protein function. *Nucleic acids research*, 31(13), 3812-3814.
93. Miksik O., Mikolajczyk K. Evaluation of local detectors and descriptors for fast feature matching // *IEEE International Conference on Pattern Recognition (ICPR)*, 2012. P. 2681-2684.

94. Bowring B. R. Transformation from Spatial to Geographical Coordinates // *Surv. Rev.*, Vol. 23(181), 1976. P. 323-327.
95. Photomod Lite // Ракурс (<https://racurs.ru/program-products/photomod-lite/>). Дата обращения 10.11.2020.
96. Chinchor N. MUC-4 Evaluation Metrics // *Proceedings of the Fourth Message Understanding Conference*, 1992. P. 22–29.
97. Dahlke, D., Linkiewicz, M., & Meissner, H. True 3D building reconstruction: façade, roof and overhang modelling from oblique and vertical aerial imagery // *International Journal of Image and Data Fusion*, 6(4), 2015. P. 314-329.
98. Таблица соответствия Microsoft Visual C++ стандартам языка // Microsoft Docs (<https://docs.microsoft.com/ru-ru/cpp/overview/visual-cpp-language-conformance>). Дата обращения: 15.12.2019.
99. Visual Studio // портал Microsoft (<https://visualstudio.microsoft.com/ru>). Дата обращения: 15.12.2019.
100. Intel c++ compiler // Intel Software (<https://software.intel.com/en-us/c-compilers>). Дата обращения: 15.12.2019.
101. CMake (<https://cmake.org>). Дата обращения: 15.12.2019.
102. About OpenMP // The OpenMP API specification for parallel programming (<https://www.openmp.org/about/about-us>). Дата обращения: 15.12.2019.
103. Open MPI: Open Source High Performance Computing (<https://www.open-mpi.org>). Дата обращения: 15.12.2019.
104. MPI Documents // MPI Forum (<https://www.mpi-forum.org/docs>). Дата обращения: 15.12.2019).

105. MPICH Overview //MPICH (<http://www.mpich.org/about/overview>).  
Дата обращения: 15.12.2019.
106. OpenCL Overview // The Khronos Group. Connecting Software to Silicon (<https://www.khronos.org/opencl>). Дата обращения: 15.12.2019.
107. О Проекте // OpenCL (<http://opencl.ru/about>). Дата обращения: 15.12.2019.
108. About // OpenCV (<https://opencv.org/about>). Дата обращения: 15.12.2019.
109. Wiki страница проекта OpenBLAS // Портал системы контроля версий GitHub (<https://github.com/xianyi/OpenBLAS/wiki>). Дата обращения: 15.12.2019.
110. BLAS (Basic Linear Algebra Subprograms) // Netlib Repository at UTK and ORNL (<http://www.netlib.org/blas>). Дата обращения: 15.12.2019.
111. Intel® Math Kernel Library // Intel Software URL: <https://software.intel.com/en-us/mkl>. Дата обращения: 15.12.2019.
112. Qt Product // Qt. Cross-platform software development for embedded & desktop (<https://www.qt.io/product>). Дата обращения: 15.12.2019.
113. Overview - wxWidgets // wxWidgets Cross-Platform GUI Library (<https://www.wxwidgets.org/about>). Дата обращения: 15.12.2019).
114. GDAL documentation // GDAL (<https://gdal.org>). Дата обращения: 15.12.2019.