**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**

**ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ**

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ,**

**МЕХАНИКИ И ОПТИКИ**

Информационных Технологий и Программирования

Факультет\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Прикладная математика и информатика

Направление (специальность)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Бакалавр прикладной математики и информатики

Квалификация (степень) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

-----------------------

Специализация \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

4528

Компьютерных технологий

Кафедра\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Группа\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ**

*Слияние изображений в ортофотоплан без видимых стыков на границах изображений с сохранением максимальной детализации* \_

*Буслаев А.В.*

Автор квалификационной работы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(подпись)

( Фамилия, И., О. )

*Родиков Д.Е.*

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(подпись)

( Фамилия, И., О. )

К о н с у л ь т а н т ы :

а) По экономике и орга-

низации производства \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(подпись)

( Фамилия, И., О. )

б) По безопасности жизне-

деятельности и экологии \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(подпись)

( Фамилия, И., О. )

в) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(подпись)

( Фамилия, И., О. )

**К з а щ и т е д о п у с т и т ь**

Зав. кафедрой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(подпись)

( Фамилия, И., О. )

“\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2013 г. Санкт-Петербург, 2013 г.

Квалификационная работа выполнена с оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата защиты “\_\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2013 г.

Секретарь ГАК \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Листов хранения \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Чертежей хранения \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**ОГЛАВЛЕНИЕ**

[ВВЕДЕНИЕ 5](#_Toc358743427)

[Глава 1. ПОСТАНОВКА ЗАДАЧИ 7](#_Toc358743428)

[Задача сшивки. 7](#_Toc358743429)

[Традиционный подход к решению задачи. 9](#_Toc358743430)

[Позиционирование фотографий. 9](#_Toc358743431)

[Общая схема алгоритма SFM 10](#_Toc358743432)

[Поиск линии реза. 13](#_Toc358743433)

[Классический подход. Оптимальные границы. 13](#_Toc358743434)

[Graph Cuts. 14](#_Toc358743435)

[Минимальный путь и минимальный разрез в графе. 16](#_Toc358743436)

[Смешивание фотографий. 17](#_Toc358743437)

[Методы смешивания изображений. 17](#_Toc358743438)

[Multi-resolution spline blending. 18](#_Toc358743439)

[Глава 2. ОПИСАНИЕ ПОДХОДА. 20](#_Toc358743440)

[Метрика качества линии реза. 20](#_Toc358743441)

[Разности интенсивностей. 20](#_Toc358743442)

[Разности градиентов. 20](#_Toc358743443)

[Градиенты. 21](#_Toc358743444)

[Создание кумулятивной метрики по нескольким характеристикам. 21](#_Toc358743445)

[Выбор оптимальной линии реза. 22](#_Toc358743446)

[Watershed segmentation. 23](#_Toc358743447)

[Глава 3. РЕЗУЛЬТАТЫ 26](#_Toc358743448)

[Тестовая система 26](#_Toc358743449)

[Способ тестирования 26](#_Toc358743450)

[Результаты тестирования 26](#_Toc358743451)

[ЗАКЛЮЧЕНИЕ 31](#_Toc358743452)

[Библиография 32](#_Toc358743453)

# ВВЕДЕНИЕ

Основной целью данной работы является реализация эффективного подхода к выбору линии разреза между фотографиями фотоплана и алгоритмическое улучшение реализованного подхода для получения выигрыша в качестве или скорости применимо к конкретной задаче. Для чего необходимо решение данной задачи?

В настоящее время в отделе интеллектуальных ГИС ЗАО Транзас-Технологии проводится разработка коммерческого продукта по сшивке фотопланов, предназначенного в основном для нужд фотограмметристов.

В задаче сшивки фотопланов наиболее важной подзадачей является создание эффекта целостности полученного изображения, а также достижение наибольшей точности фотоплана для дальнейшей привязки точек фотоплана к реальным точкам на земной поверхности по координатам GPS, ГЛОНАСС.

Однако целостность изображения не может быть получена без специального выбора линий стыковки изображений фотоплана из-за естественных проблем проецирования местности из трех измерений в два.

В данной работе предлагается идея оптимизации подхода к решению задачи выбора линии разреза между изображениями фотоплана путем сегментации области, в которой необходимо провести линию разреза, а также добавление к подходу особых характеристик, наиболее значимых для конкретной задачи.

Первая глава работы посвящена краткому обзору общей схемы работы алгоритма по сшивке мозаик, а также обзору существующих методов решения задачи нахождения линии разреза между изображениями фотоплана.

Во второй главе на основании анализа существующих решений приводится подробное описание работы реализованного алгоритма, а также реализованных идей по алгоритмической оптимизации времени работы и качества алгоритма.

В третьей главе приводятся результаты тестирования подхода на реальных данных и сравнение с существующими подходами.

В заключении дано описание текущего состояния разработки и подведены итоги работы.

# Глава 1. ПОСТАНОВКА ЗАДАЧИ

## Задача сшивки.

Фотограмметрия – технология дистанционного зондирования Земли, позволяющая определять геометрические, количественные и другие свойства объектов на поверхности земли по фотографическим изображениям, получаемым с помощью летательных аппаратов любых видов. В настоящее время изображения для фотограмметрии получают как кадровыми, щелевыми и панорамными фотоаппаратами, так и с помощью радиолокационных, телевизионных, тепловых и лазерных систем[1].

При съемке местности с летательного аппарата, оборудованного кадровым фотоаппаратом, получается большое количество снимков с различных точек полетного задания. Для создания фотоплана эти снимки должны быть объединены в мозаику большого размера. Так как фотопланы создаются для определения различных свойств объектов на поверхности земли, крайне важно, чтобы фотопланы были максимально точными и плавными. В идеальном случае не должно быть видно, что фотоплан составлен из сотен и тысяч различных снимков, сделанных с различных точек над земной поверхностью[2]. Точность фотоплана зависит от множества факторов, таких как: точность позиционирования фотографий, наличие высотных объектов, наличие движущихся объектов, время съемки (Рис. 1).

Рис. 1. Виды артефактов.

Точность позиционирования фотографий друг относительно друга должно обеспечивать решение задачи позиционирования фотографий[3]. Компенсацию разностей яркостей фотографий на стыке, которые получаются вследствие неравномерного освещения поверхности земли с течением времени, обеспечивает смешивание(blending) фотографий. Для устранения же других типов артефактов на изображении, таких как нестыковка поверхностей высотных объектов, требуется проводить особые линии разреза, которые не совпадают с натуральной границей изображения, а проходят особым способом по общей части двух фотографий (Рис. 2).

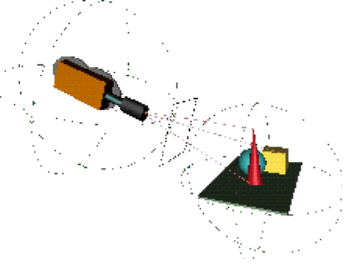
Рис. 2. Линия разреза.

## Традиционный подход к решению задачи.

### Позиционирование фотографий.

Задача позиционирования фотографий может быть решена множеством различных способов. Среди них: метод нахождения оптического потока[4], нахождение матрицы преобразования на основании найденных соответствий особых точек на изображениях[5-7] и другие.

Однако преобразования пространства при съемке могут быть много более сложными, чем то, что могут обеспечить методы, упомянутые выше. Преобразования пространства при аэрофотосъемке сильно зависят от рельефа местности, поэтому лучшим вариантом будет восстановление 3D поверхности, с последующим проецированием текстур на плоскость. Такую функциональность обеспечивает решение задачи SFM(Structure from motion)[8]. Задача SFM подразумевает под собой получение 3D структур из последовательности 2D изображений (Рис. 3).

Рис. 3. Постановка задачи SFM. Движение камеры относительно неподвижной сцены.

### Общая схема алгоритма SFM

На вход поступают изображения, обладающие достаточным перекрытием между собой, параметры объектива (фокусное расстояние, параметры главной точки изображения и коэффициенты радиальной дисторсии).

Инкрементальный алгоритм SFM выполняет следующие действия:

1. Поиск особых точек и создание их дескрипторов. Число особых точек напрямую зависит от характера подстилающей поверхности, запечатленной на снимках, и может быть недостаточным в случае квазирегулярной поверхности (водоемы, залесенные территории).
2. Перебор пар снимков с поиском соответствия особых точек, принадлежащих этим снимкам. Поиск соответствий между ключевыми точками для смежных аэрофотоснимков осуществляется с использованием kd-дерева и эвристики k ближайших соседей.
3. Выбор пары фотоснимков с максимальным числом соответствий.
4. Восстановление взаимного расположения и ориентации камер в момент экспозиции для выбранной пары снимков.
5. Выбранная пара снимков рассматривается как снимки стереопары с известной калибровкой, и применяется классический метод триангуляции для получения 3D облака точек.
6. Затем итеративно повторяются следующие шаги:
   1. Выбор снимков из числа необработанных снимков блока, имеющих достаточное число проекций построенных 3D точек. В случае недостаточного числа проекций у лучшего снимка итерации прекращаются.
   2. Вычисление местоположения и ориентации точки съёмки для данного кадра относительно обработанного набора кадров. На основе соответствий между 3D точками и их проекциями на кадре строится линейная система уравнений, решением которой является проективная матрица камеры.
   3. Для каждой 3D точки без вычисленного местоположения производится попытка вычислить его. Для вычисления положения точки выбираются снимки, на которых есть проекции этой точки и для которых известно положение и ориентация камер. Затем, по полученным данным решается задача триангуляции построением линейной системы с координатами 3D точки в качестве неизвестных параметров.
   4. Отсеиваются 3D точки, отклонение проекций которых от положений соответствующих точек интереса превышает заданный порог.
   5. Глобальная нелинейная оптимизация (sparse bundle adjustment) [9, 10] взаимного расположения точек съёмки и положений 3D точек в пространстве (для обработанных кадров). Критерием оптимизации выступает среднеквадратичное отклонение проекций построенных 3D точек на плоскости изображений от положений соответствующих им особых точек. Для решения разреженной системы нелинейных уравнений применяется метод нелинейной оптимизации Левенберга-Марквардта с оптимизацией для случая разреженной системы[11].

Шаги c-e повторяются несколько раз. Условием их прекращения является неизменное число 3D точек или совершение максимального заданного числа итераций.

Итерации прекращаются после завершения обработки всех снимков группы (блока).

Результатом работы алгоритма SFM является:

1. Набор матриц поворота и трансляции для каждого обработанного снимка, описывающие положение и ориентации камеры в точках съёмки. Данные матрицы заданы в локальной декартовой системе координат.
2. Набор локальных координат 3D точек.

Далее по набору локальных координат 3D точек строится рельеф, на который проецируется снимок. В последние годы данный алгоритм был модифицирован и улучшен[12]. Результат работы данного алгоритма представлен на рисунках 4,5.

Рис. 4. Примеры изображений, поступивших на вход SFM.

Рис. 5. Выравненные изображения

## Поиск линии реза.

Подходы к сшиванию изображений в литературе делятся на два основных класса: это смешивание изображений у границы и поиск линии реза. В большинстве случаев просто смешивание изображений у границы может создавать видимые артефакты на изображениях, так как сцена сильно неоднородна по высотам. Также при увеличении силы размывания на полученном изображении могут появляться артефакты размытия, которые нежелательны при построении точных фотопланов, использующихся в фотограмметрии.

С другой стороны поиск линии реза позволяет сшивать фотоизображения в тех местах, где это будет наименее заметно. После этого в работе применяются методы из первого класса с небольшим порогом.

### Классический подход. Оптимальные границы.

Формально классический подход к задаче нахождения линии разреза между фотографиями мозаики определен следующим образом:

Для данного набора из n изображений мозаики *I1, I2..In* и мозаики *P*, задача поиска линии разреза может быть определена как поиск дискретной маркировки *L(p) ∈ (1...n)* для всех пикселей мозаики *p ∈ P*, которая минимизирует ошибку перехода от одного изображения к другому. *L(p) = k* означает, что пиксель в положении p на панораме взят с изображения *Ik*. Переход определяется энергией кусочной гладкости *Es(p, q)*, где *p, q ∈ N* -маркировки соседних элементов, *N* множество всех пар пикселей. Необходимо минимизировать сумму энергий всех соседей, *E*. В задаче поиска линии реза эта энергия обычно[13] выражается следующим образом:

При минимизации переходов между пикселями:

или при минимизации в градиентах:

где *L(p)* и *L(q)* - маркировки двух пикселей. Видно, что *L(p) = L(q)* подразумевает *Es(p, q) = 0*. Минимизация энергии в терминах изменения значения пикселей работает хорошо в контексте плохо позиционированных фотографий или движущихся объектов на сцене, когда минимизация в градиентах предоставляет хорошие данные, подаваемые на вход для техник, таких как gradient domain blending. Также может использовать линейная комбинация двух техник.

### Graph Cuts.

Этот подход дает хорошие решения приведенной задачи. Доказано, что задача нахождения глобального минимума энергии является *NP-полной*[14] и алгоритм Graph cuts дает хорошую аппроксимацию. Этот алгоритм показывает хорошие результаты на множестве различных функций для минимизации[15] и был адаптирован для задачи нахождения линии разреза. Однако данный алгоритм является вычислительно сложным и требует больших объемов оперативной памяти, установленной на вычислительном устройстве. Таким образом, основные исследования этого алгоритма проводились с целью улучшения скорости работы. Существует реализация на *GPU*, распределенные системы и иерархическая система.

Каким образом задачу маркировки можно переформулировать в задачу Graph cut?

Возьмем энергию перехода между пикселями p, q:

и построим граф, в котором вершинами будут все пиксели изображения, а ребра будут соединять все смежные пиксели. Весами ребер будет являться энергия перехода. Тогда решением задачи маркировки будет хорошо зарекомендовавший себя алгоритм min cut/max flow поиска минимального разреза в графе[16].

Реализация данного алгоритма имеется в библиотеке с открытым исходным кодом *OpenCV*[17].

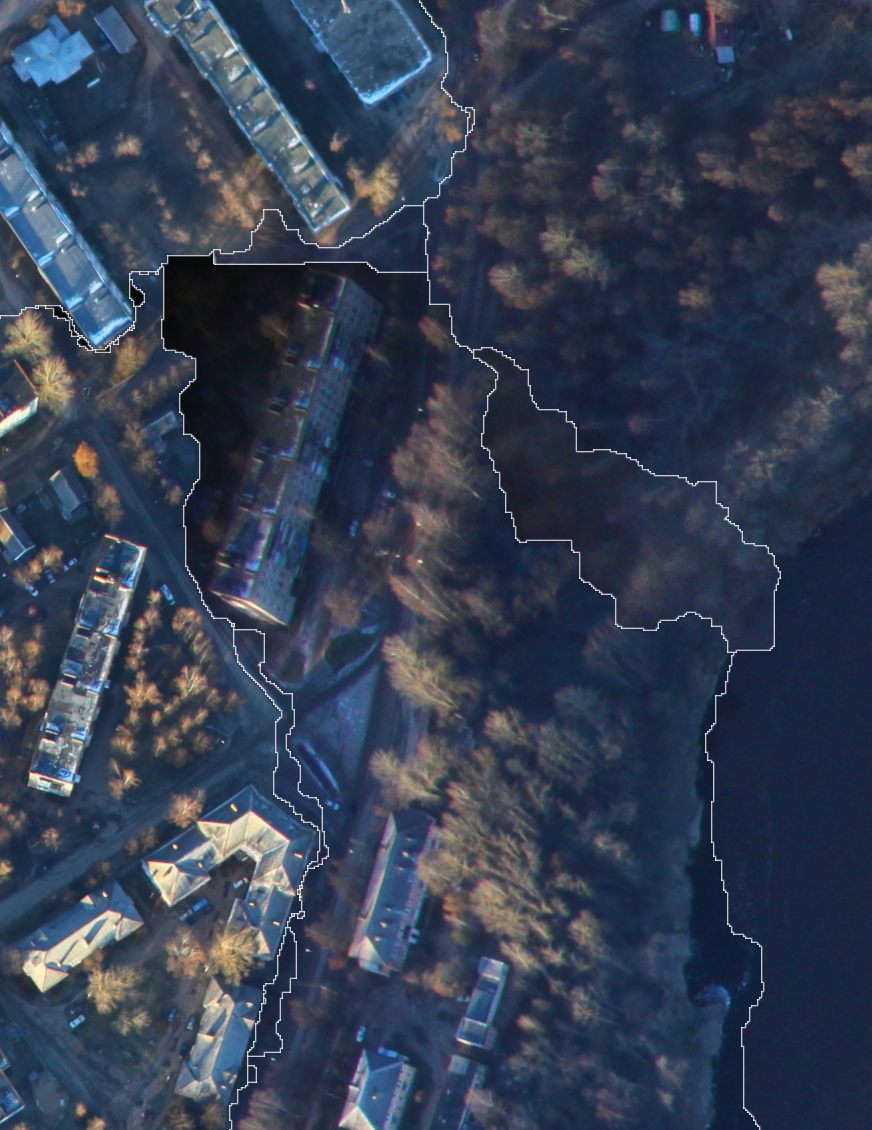
 Из-за специфики алгоритма возможно появление мелких “островов” на фотоплане, из-за которых может ухудшаться общее качество фотоплана, а также это может приводить к некорректной работе смешивания фотографий (рис. 6).

Рис. 6. “Острова”

### Минимальный путь и минимальный разрез в графе.

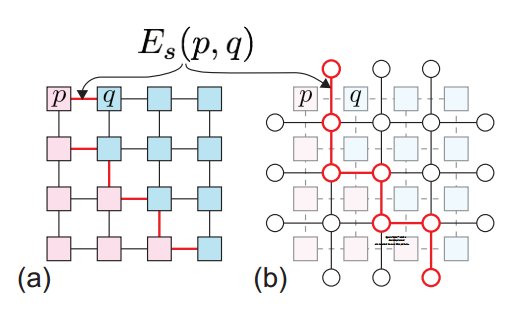
При вычислении оптимальной линии разреза между двумя изображениями задача бинарной маркировки эквивалентна вычислению минимального разреза в графе, чьими вершинами являются пиксели, а ребра соединяют пиксели с их соседями. Тогда веса ребер является энергией, которую требуется минимизировать, см. Рис. 7a. Если у нас четырехсвязность и мы имеем двойственный граф (Рис. 7b), мы видим, что решение задачи о минимальном разрезе эквивалентно решению задачи о минимальном пути в двойственном графе. Это было доказано для SSSD графов [18]. Эти подходы эквивалентны в смысле, что решение задачи о минимальном разрезе определяет маркировку пикселей L(p), в то время как минимальный путь определяет линию, которая разделяет пиксели, имеющие разные маркеры.

Рисунок 7. Граф области сшивки.

Этот важный факт дает возможность уменьшить алгоритмическую сложность нахождения линии реза при условии, что каждый раз алгоритм будет сшивать друг с другом ровно два изображения. При последовательном алгоритме сшивки, реализованном в тестирующей системе, это не является ограничением, а наоборот предоставляет возможность для адаптации подхода к сшивке изображений в реальном времени.

Последовательный алгоритм заключается в следующем: изображения подаются по одному и пристыковываются к уже существующему фотоплану. При каждой новой стыковке из большого изображения фотоплана вырезается область, к которой будет пришиваться поступившее изображение. Далее происходит поиск точек пересечения двух изображений проходом по границе одного из них и проводится поиск линий реза между всеми парами точек пересечения.

## Смешивание фотографий.

### Методы смешивания изображений.

Выбор линии реза предоставляет оптимальную линию, по которой нужно соединить два изображения, но пиксели по разные стороны от этой линии до сих пор могут иметь некоторую степень различности. Для минимизации видимости артефактов такого рода следует применить операцию смешивания к полученным изображениям. Простейшие подходы, такие как Alpha Blending / Feathering действуют следующим образом: на границе каким-либо образом берется взвешенное среднее двух снимков или градиентный переход, из-за чего стык становится плавным. Но такой подход в простейшем случае может приводить к появлению “призраков”(ghosting). Для лучшего смешивания пикселей на границах фотоизображений применяется следующий метод.

### Multi-resolution spline blending.

Техника *multi-resolution spline blending*[19] была выбрана, так как смешивание с ее помощью сохраняет высокочастотные края, что предотвращает появление ореолов на изображениях. Алгоритм состоит из нескольких шагов.

1. Поиск масок изображений с помощью линии реза. Все пиксели по одну сторону от линии реза объявляются черными, а по другую - белыми.
2. Запуск полосовых фильтров (*band-pass ﬁlters*[20]) на исходных изображениях и их масках для получения разложения Фурье.
3. Смешивание каждой полосы изображения A с соответствующей полосой изображения B, используя соответствующую маску для данной полосы частот. Низкие частоты определены серыми пикселями на маске и имеют взвешенное среднее, основанное на уровне серого. Это позволяет смешивать низкочастотные компоненты более плавно, чем высокочастотные.
4. Суммируем все изображения из каждой частотной полосы и получаем финальное изображение.

Смешивание изображений таким способом позволяет низкочастотным компонентам быть рассеянным по большому радиусу и смешанными, что ведет к увеличению плавности перехода на границе изображений. Пример работы данного алгоритма можно увидеть на рисунке 8.

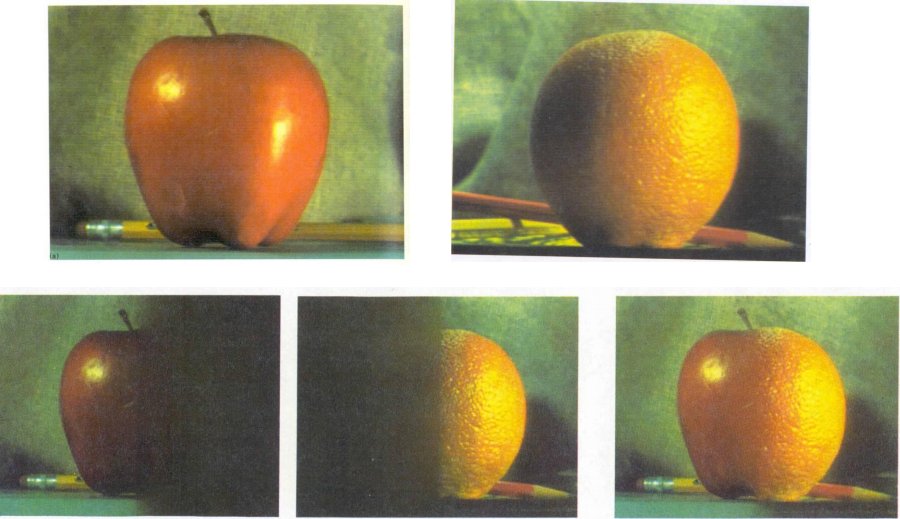


Рис. 8. Multi-resolution spline blending

# Глава 2. ОПИСАНИЕ ПОДХОДА.

## Метрика качества линии реза.

Для решения задачи проведения линии реза между кадрами необходимо найти такую кривую, разделяющую два изображения, которая бы минимизировала визуальные артефакты на финальном плане. Как определить, будет ли визуальный артефакт после сшивки двух кусков изображения?

Для выбора линии реза нам необходимо ввести какие-либо меры для определения качества пикселей, через которые она проходит. В данной работе выбрано три вида метрик: разность интенсивностей, градиенты и разности градиентов.

### Разности интенсивностей.

Понятно, что если два пикселя в одном месте сильно отличаются по интенсивности друг от друга, то в этом месте изображения сошлись не очень качественно. Поэтому через такие пиксели линию реза проводить не следует. На этом шаге происходит поиск изображения, являющегося разностным для региона пересечения, после чего это изображение нормализуется и сохраняется в ОЗУ. Нормализация изображения позволяет уменьшить влияние шума на работу алгоритма. Далее изображение отражается. Такая метрика хорошо зарекомендовала себя в решении задачи сшивки, когда на сцене присутствуют движущиеся объекты[21].

### Разности градиентов.

Самой сильной характеристикой должна быть информация о том, что две границы объектов не сходятся. Эту информацию обеспечивает разностное изображение градиентов. Данный подход также был предложен ранее[22].

### Градиенты.

Очень часто наиболее видимые графические артефакты возникают на границах объектов изображений. При том, чем четче граница, тем сильнее заметен артефакт. Таким образом, имеет смысл добавить к метрике также и информацию о границах объектов на изображении. Информация о границах объектов может быть получена с изображений, полученных с помощью оператора градиента.

### Создание кумулятивной метрики по нескольким характеристикам.

Для учета всех предложенных метрик с разными весами можно взять взвешенную сумму полученных изображений (Рис. 9). Веса регулируются в зависимости от характера рельефа местности.

Рис. 9. Кумулятивная метрика.

## 

## Выбор оптимальной линии реза.

Линия реза не должна создавать видимых артефактов. То есть она должна проходить только по “хорошим” пикселям (которые не принадлежат границе объектов, в которых маленькая разность интенсивностей). Формально это означает, что линия разреза должна проходить по всем тем пикселям, которые в метрическом изображении имеют наименьшее значение. Таким образом, можно представить изображение в виде графа, на котором вершинами являются пиксели, а ребрами – цвета пикселей, в которые входит ребро. Тогда можно найти минимальные пути от стартового пикселя до всех остальных с помощью алгоритма Дейкстры поиска минимального пути в графе[23]. После этого выбирается минимальный путь до финальной вершины (Рис. 10).

Рис. 10. Минимальный путь в графе

Стоит отметить, что с небольшим увеличением сложности алгоритма есть возможность проведения интерактивных линий разреза. Суть заключается в том, чтобы посчитать минимальные пути в графе от стартовой и от финишной вершины до всех остальных, после чего можно выбрать на изображении точку, через которую должна пройти линия разреза и без дополнительных подсчетов провести оптимальную линию разреза через эту точку.

## Watershed segmentation.

Для обеспечения наиболее точных и качественных линий реза необходимо запускать алгоритм на изображениях без уменьшения размера. Но при этом время работы алгоритма все равно остается очень большим (старая реализация могла работать до нескольких часов на паре фотографий). Поэтому очень важно увеличивать скорость работы алгоритма, чтобы при построении наиболее качественных фотопланов время работы было удовлетворительным. Для ускорения работы подхода предлагается использовать следующую идею:

При сегментации разностного изображения, на котором мы ищем линию реза, мы получаем регионы, перепады яркости в которых незначительны.

Алгоритм сегментации watershed представляется следующим образом: Изображение представляет собой топологическую карту местности, где более темные места являются впадинами, а более яркие являются вершинами. Источники воды помещаются во все локальные минимумы, и плато начинает наполняться водой, которая растекается по ландшафту. Когда вода из одного истока встречается с водой из другого истока – в этом месте возникает барьер[24](Рис. 11).

Рис. 11. Сегментация Watershed.

Таким образом, если начинать “лить воду” из мест, имеющих большой вес для алгоритма Дейкстры поиска минимального пути в графе, тогда границы сегментов в худшем случае будут проходить на границе областей, по которым не следует проводить линию реза. Для небольшого отступления от этих границ можно использовать операции эрозии(erode)[25].

Также такое отступление повышает вероятность того, что линия разреза не пройдет по пикселям, имеющим большой вес. По этой причине возможно повышение качества сшивки.

Далее строится граф, в котором вершинами теперь являются только пиксели, которые лежат на границах сегментированных областей и запускается алгоритм Дейкстры поиска минимального пути в графе (Рис. 12).

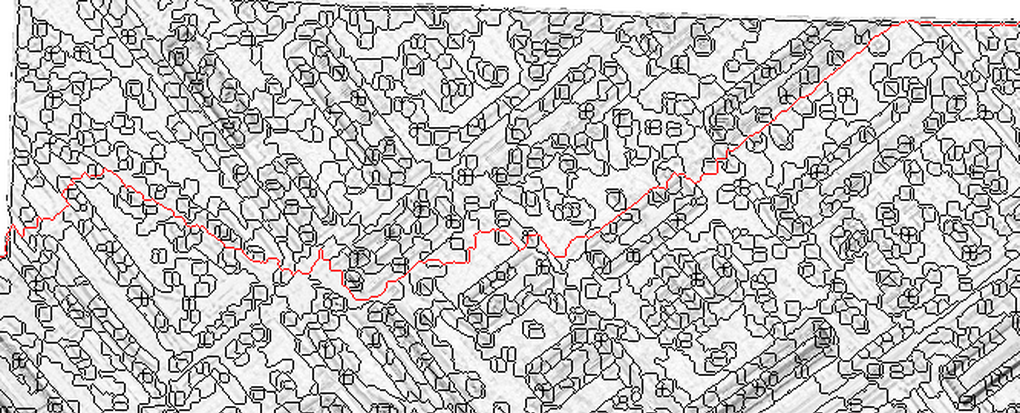


Рис. 12. Минимальный путь в watershed-графе.

Такой подход позволяет получить некоторый выигрыш в скорости поиска линии реза относительно классического подхода, основанного на поиске линии реза алгоритмом Дейкстры поиска минимального пути в графе.

Однако у алгоритма watershed есть ряд недостатков. Одним из важных недостатков является отсутствие очевидного способа поиска локальных минимумов на изображении. В данной работе для нахождения локальных минимумов строится равномерная сетка на изображении, в каждой из ячеек которой производится поиск локального минимума. Размер сетки зависит от размера изображения. Наилучшие результаты получены, когда шаг сетки составляет приблизительно 1/300 – 1/500 наибольшего измерения изображения.

Также алгоритм watershed очень неустойчив к шумам. Этот недостаток следует компенсировать гауссовым размытием[25].

# Глава 3. РЕЗУЛЬТАТЫ

## Тестовая система

Для первоначального тестирования разработанного алгоритма использовалась машина со следующей конфигурацией: Intel core i5 2500S, 4Gb RAM. Все вычисления проводились на CPU, алгоритмы работают в однопоточном режиме. Средний размер изображений 8.5Mpix. Время в секундах.

## Способ тестирования

Тестирование проводилось следующим образом: алгоритм запускается 10 раз на паре изображений, считается время работы алгоритма (без времени считывания с диска и записи файлов на диск). Далее результат усредняется и заносится в таблицу. Для тестирования выбраны изображения из внутренних наборов данных – пролеты «Боровичи» и «Новгород».

## Результаты тестирования

Таблица 1. Результаты.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Image | OpenCV graph cut | Dijkstra's | Watershed | memory OpenCV | memory Dijkstra | ocv/dij | dij/wshed | ocv/wshed |
| Borovichi1 | 70.973 | 2.71 | 1.77 | 1.4 | 0.4 | 26.19 | 1.53 | 40.10 |
| Borovichi2 | 226.33 | 3.13 | 1.85 | 1.3 | 0.4 | 72.31 | 1.69 | 122.34 |
| Borovichi3 | 191.006 | 3.12 | 1.83 | 1.4 | 0.4 | 61.22 | 1.70 | 104.37 |
| Borovichi4 | 66.1748 | 3.77 | 2.01 | 1.4 | 0.4 | 17.55 | 1.88 | 32.92 |
| Borovichi5 | 135.115 | 2.91 | 1.88 | 1.4 | 0.4 | 46.43 | 1.55 | 71.87 |
| Borovichi6 | 284.738 | 3.46 | 1.86 | 1.3 | 0.4 | 82.29 | 1.86 | 153.08 |
| Borovichi7 | 234.409 | 3.32 | 1.84 | 1.3 | 0.4 | 70.61 | 1.80 | 127.40 |

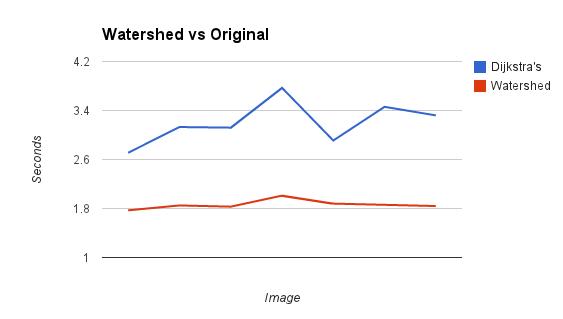
 Данные, указанные в таблице 1, визуализируются в следующих графиках:

График 1. Сравнение подходов, основанных на алгоритме Дейксты и подхода, основанного на Graph cut.

График 2. Сравнение оригинального и модифицированного подхода, основанного на алгоритме Дейкстры.

Полученные результаты показывают, что прирост скорости относительно реализации из библиотеки OpenCV составляет в среднем от 30 до 150 раз, что считается ощутимым приростом производительности. Прирост скорости модифицированного подхода, основанного на алгоритме Дейкстры, относительно оригинального подхода составляет в среднем 1.7 раза.

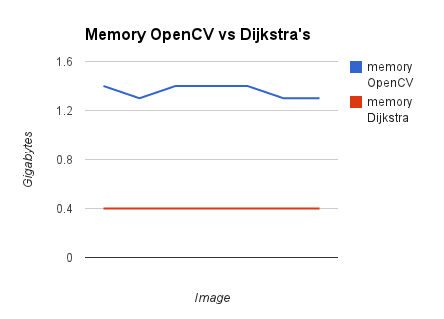
Из-за ограничений оперативной памяти на тестируемой машине изображения имеют такое низкое разрешение. Из графика 3 видно, что по количеству потребляемой оперативной памяти алгоритм, использующий нахождение минимального пути в графе дает некий выигрыш.

График 3. Потребляемая оперативная память.

Таким образом дополнительное сравнение оригинального и модифицированного подхода было проведено на изображениях размером 32-38Mpix(максимальный размер изображений, на которых алгоритм не выходит за рамки 4Gb оперативной памяти). Тестирование алгоритма на больших размерах изображений считается нецелесообразным, так как время работы алгоритма будет более ощутимым.

Следует отметить, что за один сеанс сшивается в среднем 700-800 фотоизображений. Результаты дополнительного тестирования представлены в таблице 2 и на графике 4.

Таблица 2. Тестирование подходов, основанных на алгоритме Дейкстры на изображениях в высоком разрешении.

|  |  |  |  |
| --- | --- | --- | --- |
| Test | Dijkstra’s algorithm | Watershed modification | Dijkstra/watershed |
| Novgorod1 | 12.5 | 7.93 | 1.58 |
| Novgorod2 | 15.82 | 8.43 | 1.88 |
| Novgorod3 | 19 | 9.89 | 1.92 |
| Novgorod4 | 16.38 | 11.14 | 1.47 |
| Novgorod5 | 16.21 | 9.15 | 1.77 |
| Novgorod6 | 18.54 | 11.75 | 1.58 |

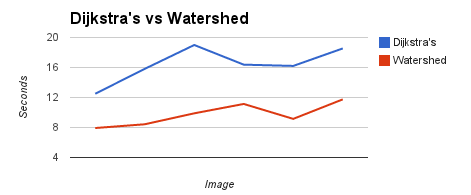


График 4. Тестирование подходов, основанных на алгоритме Дейкстры на изображениях в высоком разрешении.

Из последней таблицы и графика видно, что время работы алгоритмов на изображениях, имеющих впечатлительные размеры, остается приемлемым, а выигрыш модифицированного подхода почти не меняется в зависимости от разрешения.

# ЗАКЛЮЧЕНИЕ

В ходе выполнения данной работы реализован быстрый и качественный алгоритм поиска линии разреза между изображениями фотоплана, а также предложен подход по алгоритмическому увеличению быстродействия.

Реализованный подход дает очень ощутимый выигрыш в скорости работы и используемой оперативной памяти, что позволяет запускать его с гораздо меньшим уменьшением размеров изображений, что в свою очередь сильно увеличивает качество проведенных линий разреза.

Предложенная оптимизация хоть и не улучшает асимптотическую сложность разработанного алгоритма, но дает порядочное уменьшение константы, а также не является эвристикой для конкретного случая и может быть использована на сколь угодно различных по размеру изображениях для любых задач.

Разработанная система интегрирована в коммерческий продукт ЗАО Транзас-Технологии “Express Mosaic”(Рис. 13).

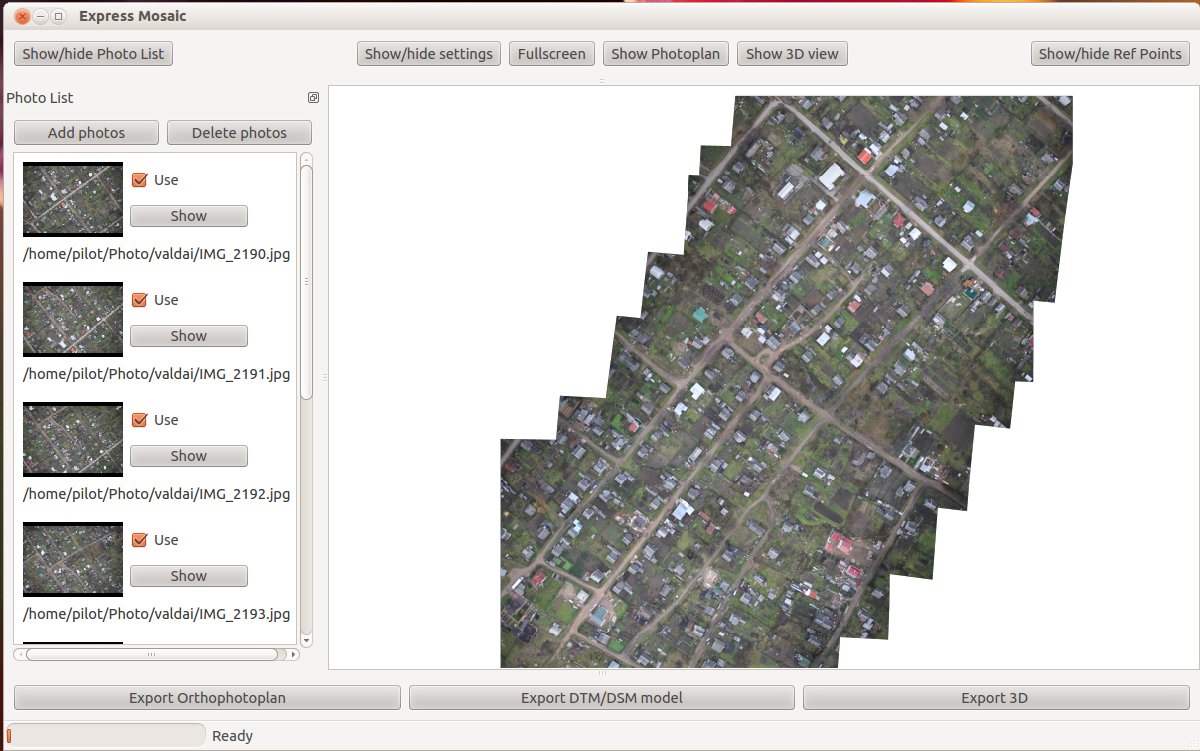


Рис. 13. Express Mosaic.

# Библиография

1. Алексапольский Н. М. Фотограмметрия: Часть 1 / Под общ. ред. доктора технич. наук проф. А. Н. Лобанова. — М.: Геодезиздат, 1956
2. ГКИНП 36 Инструкция по фотограмметрическим работам при создании топографических карт и планов. Москва, ЦНИИГАиК, 2002
3. Lisa Gottesfeld Brown, A survey of image registration techniques (abstract), ACM Computing Surveys (CSUR) archive, Volume 24, Issue 4, December 1992) <http://dl.acm.org/citation.cfm?id=146374>
4. B. D. Lucas and T. Kanade (1981), An iterative image registration technique with an application to stereo vision. <http://cseweb.ucsd.edu/classes/sp02/cse252/lucaskanade81.pdf>
5. Lowe D. G. Object recognition from local scale-invariant features. 1999. <http://robotics.caltech.edu/readinggroup/vision/LoweICCV99.pdf>
6. Terryberry T. B., French L. M., Helmsen J. GPU accelerating speeded-up robust features. 2008. <http://people.xiph.org/~tterribe/pubs/gpusurf.pdf>
7. E. Rosten. Faster and better: a machine learning approach to corner detection, 2008. <http://arxiv.org/pdf/0810.2434.pdf>
8. Robertson D. P., Cipolla R.. Structure from motion. 2008. <http://mi.eng.cam.ac.uk/~cipolla/publications/contributionToEditedBook/2008-SFM-chapters.pdf>
9. Konolige K.. Sparse Sparse Bundle Adjustment. 2010. <http://www.willowgarage.com/sites/default/files/ssba.pdf>
10. Lipsky C., Bose D., Eisemann M., Berger K., Magnor M.. Sparse Bundle Adjustment Speedup Strategies. 2010.

<http://www.cg.tu-bs.de/media/publications/sparse-bundle-adjustment-speedup-strategies.pdf>

1. Agarwal1 S., Snavely N., Simon I., Seitz S. M., Szeliski R.. Building Rome in a Day. 2009. <http://grail.cs.washington.edu/rome/rome_paper.pdf>
2. Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, Факультет информационных технологий и программирования, Кафедра компьютерных технологий «Иерархическое восстановление разреженной структуры пространства и точек съёмки по набору фотографий», Д.Е. Родиков (Магистерская диссертация).
3. AGARWALA, A., DONTCHEVA, M., AGRAWALA, M., DRUCKER, S. M., COLBURN, A., CURLESS, B., SALESIN, D., AND COHEN, M. F. 2004. Interactive digital photomontage. ACM Trans. Graph 23, 3, 294–302 <http://grail.cs.washington.edu/projects/photomontage/photomontage.pdf>
4. BOYKOV, Y. Y., AND JOLLY, M. P. 2001. Interactive graph cutsfor optimal boundary and region segmentation of objects in N-D images. In ICCV. <http://vision.stanford.edu/teaching/cs231b_spring1213/papers/ICCV03_BoykovJolly.pdf>
5. BOYKOV, Y., AND KOLMOGOROV, V. 2004. An experimental comparison of min-cut/max-ﬂow algorithms for energy minimization in vision. IEEE Trans. Pattern Anal. Mach. Intell 26, 9. <http://www.csd.uwo.ca/~yuri/Papers/pami04.pdf>
6. Ford, L. R.; Fulkerson, D. R. (1956). "Maximal flow through a network". Canadian Journal of Mathematics 8: 399–404.
7. <http://opencv.org/>
8. HASSIN, R. 1981. Maximum ﬂow in (s; t)-planar networks. Inform. Proc. Lett. 13, 107.
9. PETER J. BURT and EDWARD H. ADELSON, A Multiresolution Spline With Application to Image Mosaics. <http://www.cs.princeton.edu/courses/archive/fall05/cos429/papers/burt_adelson.pdf>
10. Belle A. Shenoi (2006). Introduction to digital signal processing and filter design. John Wiley and Sons. <http://www.cse.iitk.ac.in/users/chrt/EE627/books%20for%20ee627/104529856-Introduction-to-Digital-Signal-Processing-and-Filter-Design.pdf>
11. J. Davis, Mosaics of scenes with moving objects, in: Computer Vision and Pattern Recognition, 1998. Proceedings 1998 IEEE Computer Society Conference on, 1998, pp. 354–360. <http://users.soe.ucsc.edu/~davis/panorama/cvpr98_moving_objects.pdf>
12. Alec Mills, Gregory Dudek, Image stitching with dynamic elements, Image and Vision Computing 27 (2009). <http://ar.newsmth.net/att/1d7d1e272c96/Image_stitching_with_dynamic_elements.pdf>
13. Dijkstra, E. W. (1959). "A note on two problems in connexion with graphs". Numerische Mathematik 1
14. Luc Vincent and Pierre Soille. Watersheds in digital spaces: an efficient algorithm based on immersion simulations. In IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 13
15. Шапиро Л., Стокман Дж.. Компьютерное зрение. 2006.