На правах рукописи

Васильев Антон Игоревич

## ПРОГРАММНОЕ И АЛГОРИТМИЧЕСКОЕ ОБЕСПЕЧЕНИЕ СИСТЕМ КОМПЬЮТЕРНОГО ВИДЕНИЯ С НЕСКОЛЬКИМИ ПОЛЯМИ ЗРЕНИЯ

05.13.11 – Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей

ДИССЕРТАЦИЯ на соискание ученой степени кандидата физико-математических наук

Научный руководитель – д.ф.-м.н. Богуславский Андрей Александрович

Москва – 2013 год

## оглавление

ВВЕДЕНИЕ
Глава 1. СИСТЕМЫ КОМПЬЮТЕРНОГО ВИДЕНИЯ С НЕСКОЛЬКИМИ
ПОЛЯМИ ЗРЕНИЯ 13
1.1. Аэрофотосъемка с использованием много-объективной камеры
бокового обзора
1.2. Программно-технический комплекс для мониторинга местности в
реальном времени с использованием БПЛА
1.3. Оперативное картографирование с использованием стереокамер 17
1.4. Автоматизация процесса регистрации трещин с использованием
бытовой камеры 19
1.5. Анализ доступных программных средств применительно к системам
компьютерного видения с несколькими полями зрения
1.6. Анализ доступных программных средств для реализации
программных компонент
1.7. Выводы
Глава 2. ПРОЕКТИРОВАНИЕ ПРОГРАММНЫХ КОМПОНЕНТ СИСТЕМ
КОМПЬЮТЕРНОГО ВИДЕНИЯ С НЕСКОЛЬКИМИ ПОЛЯМИ
ЗРЕНИЯ
2.1. Математическая модель камеры
2.2. Минимизация ошибки перепроецирования
2.3. Классификация задач определения параметров камеры
2.4. UML-диаграмма классов компоненты для определения параметров
камеры
2.5. Сопоставление изображений на основе признаковых методов 34
2.6. Сопоставление аэрофотоснимков

2.7. Сопоставление снимков с БПЛА
2.8. UML-диаграмма классов компоненты для сопоставления
изображений
2.9. Выводы
Глава 3. ПАРАЛЛЕЛЬНЫЕ РЕАЛИЗАЦИИ АЛГОРИТМОВ СОПО-
СТАВЛЕНИЯ ИЗОБРАЖЕНИЙ 46
3.1. Программно-аппаратная архитектура параллельных вычислений на
основе графических процессоров NVidia 46
3.2. Параллельная реализация алгоритма сопоставления
аэрофотоснимков
3.3. Оценка производительности параллельной реализации детектора
Харриса на GPU 58
3.4. Параллельная реализация алгоритма сопоставления снимков с
БПЛА
3.5. Сравнение и оценки производительности разработанной реализации
SIFT-детектора
3.6. Оценки производительности сопоставления снимков с БПЛА 71
3.7. Выводы
Глава 4. СИСТЕМЫ КОМПЬЮТЕРНОГО ВИДЕНИЯ С НЕСКОЛЬКИМИ
ПОЛЯМИ ЗРЕНИЯ В ПРИКЛАДНЫХ ЗАДАЧАХ
4.1. Система для автоматического построения фотосхемы местности по
данным аэрофотосъемки много-объективной камеры бокового обзора 74
4.2. Система для автоматического построения фотосхем местности по
снимкам с двух объективной камеры БПЛА
4.3. Приложение для калибровки систем компьютерного видения с
несколькими полями зрения

	4.4.	Система	компьютерного	видения	для	оперативного
	картогј	рафирования	τ			
	4.5. Си	стема для ав	втоматизации проце	сса регистра	ции тре	щин в хрупких
	тензоч	увствительні	ых покрытиях			101
	4.6. Вы	іводы				
3	ВАКЛЮ	ЧЕНИЕ				
E	БЛАГОД	<b>ІАРНОСТИ</b>				
J	ІИТЕРА	ТУРА				

#### введение

#### Актуальность работы

В области автоматизированной обработки видеоданных можно выделить ряд задач, связанных с обработкой информации о наблюдаемой сцене по данным, поступающим из нескольких полей зрения. Примерами подобных задач являются наблюдение/контроль с использованием системы камер; формирование фотосхем на основе снимков, получаемых с много-объективных камер; определение пространственных характеристик объектов сцены на основе анализа стереизображений. Развитие вычислительных средств и датчиков изображений приводит к постоянному расширению сферы применения автоматизированных систем компьютерного видения с несколькими полями зрения.

В настоящее время разработан ряд коммерческих программных продуктов для анализа пространственной информации наблюдаемых объектов и сцен. По областям применения среди них выделяются пакеты фотограмметрического назначения (Erdas Imagine, Photomod), пакеты построения фотореалистичных 3D-моделей объектов (как с возможностью измерений – Photomodeller, так и преимущественно для целей визуализации – Autodesk 123D), программы построения панорамных изображений (Microsoft ICE) и другие.

Перечисленные программные пакеты не всегда удается применить для решения конкретных прикладных задач, особенно, связанных с обработкой зрительных данных, поступающих в режиме реального времени, а также требующих интеграции систем компьютерного видения в более крупные программно-аппаратные комплексы. В связи с этим актуальной является проблема проектирования и разработки программного обеспечения систем компьютерного видения с несколькими полями зрения, предназначенных для извлечения пространственной информации о наблюдаемых сценах. К пространственной информации относятся трехмерные координаты точек объектов интереса, их геометрические размеры, а также взаимное расположение объектов интереса и их частей.

Анализ существующих систем обработки изображений, представленных в опубликованной литературе, показывает, что для определения пространственной информации по данным изображений выполняются типовые операции:

- определение параметров видеокамер;
- стереоскопические измерения;
- сопоставление изображений (поиск соответствующих точек);
- построение карт глубины и 3D-моделей;
- геометрическое трансформирование и цветовая коррекция изображений.

Специальное программное обеспечение в прикладных задачах с целью извлечения пространственной информации о наблюдаемой сцене с возможностью учета априорной информации предметной области целесообразно создавать на основе повторно используемых программных компонент, решающих типовые задачи обработки. В настоящее время таких средств, охватывающих все перечисленные операции, не имеется. Для дорогостоящих фотограмметрических пакетов, как правило, не существует альтернативных открытых решений. Известные средства, предлагаемые в таких популярных библиотеках, как Intel Integrated Performance Primitives и OpenCV, содержат реализации алгоритмов, которые не всегда удается применить в прикладных задачах (например, в силу ограничений на размеры обрабатываемых изображений).

При обработке зрительных данных для обеспечения требуемого масштаба реального времени важным является вопрос выбора аппаратной структуры вычислительных средств системы компьютерного видения. В последние годы для вычислительных задач и обработки изображений расширяется применение универсальных графических процессоров (Graphic Processor Unit, GPU). Таким образом, для систем компьютерного видения актуальна разработка параллельных алгоритмов, рассчитанных на архитектуру GPU, с целью сокращения временных затрат на обработку зрительных данных.

В процессе выполнения диссертационной работы был решен ряд прикладных задач по обработке зрительных данных с целью определения пространственных характеристик и анализа объектов интереса наблюдаемой сцены:

- построение фотосхемы местности по данным аэрофотосъемки с много-объективной камеры бокового обзора;
- разработка программно-технического комплекса для мониторинга местности в режиме реального времени по информации, получаемой с беспилотного летательного аппарата (БПЛА);
- оперативное картографирование местности на основе системы видеокамер и бесплатформенной инерциальной навигационной системы (БИНС), установленных на транспортном средстве (TC);
- автоматизация процесса регистрации трещин в хрупком тензопокрытии, применяемом для анализа напряженно-деформируемого состояния (НДС) исследуемой конструкции.

На основе опыта решения прикладных задач была спроектирована архитектура компонент двух видов для систем компьютерного видения с несколькими полями зрения: для определения параметров системы камер и для сопоставления изображений. При этом для построения фотосхем были разработаны и реализованы алгоритмы параллельной обработки данных, рассчитанные на архитектуру GPU.

Целью диссертационной работы является исследование программного и алгоритмического обеспечения систем компьютерного видения с несколькими полями зрения, проектирование и разработка унифицированных повторно используемых программных компонентов для решения прикладных задач обработки зрительных данных. Для ускорения низкоуровневой обработки изображений ставилась цель разработки параллельных алгоритмов выделения точечных признаков и сопоставления изображений.

#### Научная новизна

Разработаны новые алгоритмы, оптимизированные для архитектуры GPU семейств NVidia GT200/Fermi, позволяющие производить следующие виды обработки зрительных данных:

 сопоставление узкополосных снимков, полученных при аэрофотосъемке с помощью много-объективной камеры бокового обзора;

 построение обзорных изображений по снимкам, получаемым с двух-объективной камеры БПЛА.

Разработано новое приложение для калибровки систем компьютерного видения с несколькими полями зрения, обеспечивающее выбор способа учета модели дисторсии в зависимости от особенностей прикладной задачи.

Впервые предложена методика автоматизации процесса регистрации трещин в хрупком тензопокрытии.

#### Практическая значимость работы

Спроектированные и реализованные программные компоненты были использованы для решения четырех прикладных задач:

- построение фотосхемы местности по данным аэрофотосъемки с много-объективной камеры бокового обзора;
- программно-технический комплекс для мониторинга местности в реальном времени с использованием БПЛА;

- оперативное картографирование местности на основе системы видеокамер и БИНС, установленных на ТС;
- автоматизация процесса регистрации трещин в хрупком тензопокрытии, применяемом для анализа НДС исследуемой конструкции.

На основе приложения для программной калибровки камер были определены параметры ряда цифровых камер (в том числе для БПЛА) и ТВкамер, а также откалиброваны стереосистемы на ТС для оперативного картографирования и для измерений в лабораторных условиях.

Разработанные параллельные алгоритмы применялись в задачах построения фотосхем местности. В аппаратном обеспечении использовались графические процессоры NVidia Tesla s1070 для серверных станций, GPU GeForce GTX 260/280/480 для настольных ПК и GPU GeForce GT 420M/425M для ноутбуков.

Программное обеспечение для построения фотосхем было разработано в рамках опытно-конструкторских работ (ОКР) ОАО "НИИ Точных Приборов" (г. Москва), которое было поставлено на эксплуатацию.

### Апробация работы

Основные результаты работы докладывались и обсуждались на следующих конференциях, симпозиумах и семинарах:

- IV-ая Всероссийская научно-практическая конференция «Перспективные системы и задачи управления», Россия, Домбай, 2009
- V-ая Международная конференция «Космическая съемка на пике высоких технологий», Россия, Московская область, 2011
- Юбилейная научно-техническая конференция ОАО НИИ ТП, посвященная 50-летию полета в космос Ю.А.Гагарина, Россия, Москва, 2011

9

- Всероссийская молодёжная конференция "Новые материалы и технологии в ракетно-космической и авиационной технике", Россия, Звездный городок, 2011
- XVI Международный научно-технический симпозиум "Геоинформационный мониторинг окружающей среды: GPS и GIS технологии", Украина, Алушта, 2011
- 11-ая Международная научно-техническая конференция "From Imagery to Map: Digital Photogrammetric Technologies", Испания, Тосса-де-Мар, 2011
- 21-ая Международная конференция по компьютерной графике и машинному зрению GraphiCon'2011, Россия, Москва, 2011
- З-я Российская конференция ИПУ им. В.А.Трапезникова РАН "Технические и программные средства систем управления, контроля и измерения", Россия, Москва, 2012
- Объединенный семинар им. М.Р.Шуры-Буры по робототехническим системам и программированию ИПМ им. М.В.Келдыша РАН, Россия, Москва, 2012

## Публикации по теме диссертации

Материалы диссертации опубликованы в 11 печатных работах, из них: 3 статьи в рецензируемых журналах, рекомендованных ВАК [1-3], 1 перевод статьи в зарубежном журнале [10], 7 статей в сборниках трудов конференций [4-9,11].

## Объем и структура диссертации

Диссертация состоит из введения, четырех глав, заключения и списка литературы. Содержание работы изложено на 116 страницах. Список литературы включает 85 наименований. В работе содержится 73 рисунка и 4 таблицы.

В первой главе вводится понятие системы компьютерного видения с несколькими полями зрения. В таких системах несколько источников изображений используются для увеличения наблюдаемого объема сцены и/или для восприятия трехмерной структуры сцены. Рассматриваются особенности, решенных прикладных задач, в которых применялись системы компьютерного видения подобного класса.

В результате анализа программного обеспечения систем компьютерного видения с несколькими полями зрения выполнена декомпозиция решаемых задач на программные компоненты. Исследована возможность применения для реализации этих компонент готовых программных библиотек (библиотека OpenCV и Intel Integrated Performance Primitives). Определены ограничения доступных библиотек и предлагается спроектировать архитектуру программных компонент, допускающих настройку с учетом специфики решаемых задач обработки зрительных данных.

Во второй главе рассматривается постановка задачи определения параметров камеры как задачи минимизации ошибки перепроецирования (minimizing the reprojection error). Выделены различные классы прикладных задач определения параметров камеры в зависимости от количества и типов неизвестных параметров. Показана применимость каждого из классов для рассматриваемых систем компьютерного видения.

Для задачи сопоставления изображений сформулирована и проанализирована общая схема на основе признаковых методов, на примере двух прикладных задач (сопоставления снимков, при аэрофотосъемке и съемке с БПЛА) описаны два подхода к поиску характерных точек и формированию их дескрипторов.

Для программной реализации спроектированных программных компонент, решающих рассматриваемые в первой главе задачи, приводится структурная UML-диаграмма классов. В третьей главе с целью оптимизации вычислений и повышения производительности алгоритмов обработки изображений предлагается использовать графические процессоры (GPU). Описаны характерные особенности архитектур современных графических процессоров и методы их программирования. Основное внимание уделяется технологии программирования CUDA. Для рассмотренных во второй главе двух схем сопоставления аэрофотоснимков разработаны их новые параллельные реализации. Приведены сравнения оригинальных реализаций отдельных этапов алгоритмов с доступными существующими вариантами, показано превосходство новых реализаций, при обработке изображений большего размера (более 49 МПикс).

**Четвертая глава** посвящена описанию разработанных алгоритмов и реализованных систем компьютерного видения для решения прикладных задач. Приведены результаты работы программ. Показано, каким образом в каждой реализованной системе компьютерного видения используются разработанные программные компоненты.

В заключении представлены основные результаты работы.

# Глава 1. СИСТЕМЫ КОМПЬЮТЕРНОГО ВИДЕНИЯ С НЕСКОЛЬ-КИМИ ПОЛЯМИ ЗРЕНИЯ

Под системами компьютерного видения с несколькими полями зрения будем понимать такие системы, в которых несколько источников изображений используются для увеличения наблюдаемого объема сцены и/или для восприятия трехмерной структуры сцены. Системы подобного класса при выполнении данной работы были разработаны для решения четырех прикладных задач:

- построение фотосхемы местности по данным аэрофотосъемки с много-объективной камеры бокового обзора;
- разработка программно-технического комплекса для мониторинга местности в режиме реального времени по информации, получаемой с БПЛА;
- оперативное картографирование местности на основе системы видеокамер и БИНС, установленных на ТС;
- автоматизация процесса регистрации трещин в хрупком тензопокрытии, применяемом для анализа НДС исследуемой конструкции.

Далее приводятся основные характеристики перечисленных систем компьютерного видения, производится анализ доступных программных пакетов и средств для решения поставленных задач.

# 1.1. Аэрофотосъемка с использованием много-объективной камеры бокового обзора

Для увеличения покрытия наблюдаемой области за один пролет в настоящее время часто применяются много объективные камеры [12-14]. Такие камеры могут рассматриваться в качестве систем с несколькими полями зрения. Одним из распространенных способов обработки данных с таких камер является приведение всех снимков к единому кадру, эквивалентному камере с одним полем зрения. В данном случае, используя стандартные фотограмметрические пакеты (например, [15,16]), можно выполнять построение фотосхем. Под фотосхемой будем понимать изображение местности, полученное из набора аэрофотоснимков в результате совмещения по общим контурам.



Рис. 1.1.1.Схема проведения аэрофотосъемки (Н- высота полета).



Рис. 1.1.2. Пример единого кадра много-объективной камеры, формируемого по изображениям от отдельных ПЗС-матриц.

На рисунке 1.1.1 показана схема проведения аэрофотосъемки с помощью много-объективной камеры бокового обзора. Камера имеет 8 объективов и 32 матрицы, 4 матрицы соответствуют одному объективу. Формирование этих кадров осуществлялось с использованием встроенного ПО камеры. Это ПО также предоставляло параметры эквивалентной камеры, способной создавать единый кадр. Пример единого кадра представлен на рисунке 1.1.2. Существенной особенностью таких кадров является большой размер – 135 МПикс или 75000х1800 пикселей (в формате RGB, 8 бит на каждый цветовой канал) и наличие линий "сшивки" от смежных матриц (см. рис. 1.1.2).

Каждый кадр формируется в формате TIFF [17] с использованием сжатия JPEG [18] и сопровождается показаниями датчика GPS в момент фотографирования.



Рис. 1.1.3. Пример фотосхемы местности.

Для описанной системы с несколькими полями зрения необходимо решить задачу автоматического построения фотосхемы местности (рис. 1.1.3), при этом скорость обработки должна быть не менее 100 кадров в час.

# 1.2. Программно-технический комплекс для мониторинга местности в реальном времени с использованием БПЛА



Рис. 1.2.1 Схема мониторинга местности и пример фотосхемы местности, построенной по снимкам с двух-объективной камеры БПЛА. Размер кадра - 2008х1336 пикселей (RGB).

Создание систем для задач мониторинга и контроля местности на основе БПЛА с установленной на нем камерой находит все большее применение для различных прикладных задач. На рис. 1.2.1 показана схема программно технического комплекса реального времени на основе БПЛА "Типчак" [19]. Для получения зрительных данных с целью покрытия наблюдаемой области местности в этом комплексе использовалась специальная двух-объективная камера. Получаемые зрительные данные подвергались обработке посредством двух независимых процедур, осуществляемых двумя операторами:

- обзор пары кадров, получаемых оператором по радиолинии (пропускная способность 5Мбит/с);
- детальный анализ локального обзорного изображения местности оператором.

Для удобства детального анализа оператором требовалось разработать программное обеспечение, позволяющее формировать фотосхему (из 3-5 пар кадров). С целью отладки отдельных независимых этапов комплекса требовалось формирование фотосхемы без использования дополнительной информации о линейных и угловых параметрах БПЛА. Также для построения обзорных изображений требовалась скорость обработки, не превышающая 10 с на пару кадров. Размер кадров 2008х1336 пикселей (в формате RGB, 8 бит на каждый цветовой канал), формат – JPEG.Также ставилась задача исследовать возможность формирования бесшовных единых кадров.

#### 1.3. Оперативное картографирование с использованием стереокамер

В целях оперативного картографирования была разработана дорожная лаборатория на базе транспортного средства (рис. 1.3.1, 1.3.2), оборудованная парой стереосистем, GPS-приемником и датчиком БИНС, а также лазерным дальномером. Основная задача этой лаборатории заключалась в измерении размеров и координат наблюдаемых объектов инфраструктуры дорог.

В качестве камер стереосистем использовались полутоновые аналоговые видеокамеры (720х576, 8бит для представления яркости, 25 Гц). При получении изображений синхронно выполнялся съем показаний БИНС (положение в географической СК WGS84 [20] и углы – тангаж/крен/рысканье). Для этой системы требовалось оценить возможность применения аналоговых видеокамер общего назначения и оценить точность измерения на дальности порядка 30-40 м.



Рис. 1.3.1.Дорожная лаборатория, оборудованная парой стереосистем.



Рис. 1.3.2. Расположение камер стереосистем для задачи оперативного картографирования местности на основе системы камер и БИНС.

1.4. Автоматизация процесса регистрации трещин с использованием бытовой камеры

В Институте Машиноведения им. А.А. Благонравого РАН была поставлена задача по разработке методики и программного обеспечения для автоматизации процесса регистрации трещин с использованием бытовой камеры (рис. 1.4.1). Известная до настоящего времени технология фиксации картин трещин в хрупких тензочувствительных покрытиях [21] заключается во фрагментном фотографировании мест образования трещин и их зарисовке на предварительно подготовленные эскизы исследуемой конструкции.

Съемка под углом 30-60°



Результат автоматизации



Рис. 1.4.1. Примеры изображений, обрабатываемых для решения задачи регистрации трещин в хрупком тензопокрытии.

Для фиксации трещин в хрупких тензочувствительных покрытиях обычно используется контролируемое направленное освещение. Наилучшая видимость трещин достигается при их наблюдении при освещении, направленном в угловом диапазоне 30-60° к исследуемой поверхности. Время, необходимое для ручной зарисовки картин трещин на этапах эксперимента, как правило, превышает длительность самих испытаний и занимает в среднем 2-3 часа. Точность зарисовки трещин в значительной степени зависит от квалификации оператора и конкретной картины их расположения.

## 1.5. Анализ доступных программных средств применительно к системам компьютерного видения с несколькими полями зрения

Характеристики рассмотренных выше прикладных задач приведены в таблице 1.5.1. Перечислим основные цели и требования к системам компьютерного видения для решения этих задач.

В рассматриваемых систем требуется обеспечить больший обзор сцены, а также определение координат и размеров объектов сцены. Для систем компьютерного видения, в которых требуется определять координаты наблюдаемых объектов, необходимо знать модель камеры, обеспечивающую выполнение измерений. Важным требованием к рассматриваемым системам компьютерного видения является требование оперативной обработки (в реальном времени или с ограничением по времени обработки).

Характеристика	Аэрофотосъемка	Съемка с	Оперативное карто-	Регистрация
задачи		БПЛА	графирование	трещин
Цель	Обзор и измере-	е- Обзор Обзор и измерения		ерения
	ния			
		2.4		
Модель камеры	Известна	Общая мо-	Требуется калибровка	
		дель		
Производитель-	100 кадров в час	5 пар кадров	Интерактивный ре-	Автоматизация
ность		за 30 с.	жим	(до 1 ч.)
Год постановки	2008 г.	2007 г.	2009 г.	2007 г.
задачи				
Известные ре-	Коммерческие фотограмметри-		Закрытые программ-	Нет
шения	ческие пакеты (Erdas Imagine,		но-аппаратные ком-	
	Photomod)		плексы (НПО "Реги-	
			он")	

Таблица 1.5.1. Характеристики рассматриваемых прикладных задач

Анализ доступных программных пакетов (Erdas Imagine, Photomod) для решения задач построения фотосхем на 2007-2009 г. показал, что эти пакеты ориентированы на обработку данных, полученных при горизонтальной аэрофотосъемке с использованием однообъективных камер. Кроме того, требования оперативного автоматического формирования фотосхем в этих пакетах, нацеленных на автоматизированную камеральную обработку, не предусмотрены.

Для решения задачи оперативного картографирования также известны закрытые программно-аппаратные комплексы (в частности, разработки НПО "Регион" [22]), адаптированные под конкретное аппаратное обеспечение.

Таким образом, анализ доступных программных средств позволяет заключить, что поставленные задачи не могут быть решены с использованием готовых программных пакетов. С учетом проведенного анализа систем компьютерного видения для перечисленных задач, была выполнена декомпозиция программного обеспечения этих систем на однотипные части. Для задач построения фотосхемы можно выделить следующие основные этапы обработки [15,16,23,24]:

- 1) подготовка проекта для обработки;
- 2) сопоставление изображений (поиск одноименных точек);
- уточнение параметров местоположения и ориентации камеры в момент фотографирования;
- 4) построение 3D-модели местности;
- 5) трансформирование снимков и построение фотосхемы.

Для стереоскопических измерений координат наблюдаемых объектов требуется предварительная калибровка стереокамер.

Таким образом, выделим определим назначение основных программных компонент, необходимых для решения поставленных задач:

 калибровка систем компьютерного видения с несколькими полями зрения;

- уточнение параметров местоположения и ориентации датчика изображения;
- сопоставление изображений;
- геометрическая коррекция изображений.

# 1.6. Анализ доступных программных средств для реализации программных компонент

Типовые компоненты систем компьютерного видения, а также доступные программные библиотеки для их реализации показаны в таблице 1.6.1. Операции калибровки и уточнения параметров местоположения и ориентации камер для реализации были объединены в одну программную компоненту (подробно рассматривается в Главе 2).

Таблица 1.6.1. Программные компоненты для выполнения типовых операций в системах компьютерного видения с несколькими полями зрения

Используемые ком-	Задачи	OpenCV	IntelIPP	
поненты				
Определение пара-	Bce	Калибровка камеры (мо-	Не реализовано	
метров камеры		дель центральной проек-		
		ции)		
Сопоставление изо-	Аэрофотосъемка	Ограничение на размер	Низкоуровневые ал-	
бражений	Съемка с БПЛА		горитмы	
Геометрическая	Bce	Нет отротрансформирования изображений на ос-		
коррекция		нове модели камеры и 3D модели сцены		
Аппаратная под-	Bce	GPU (начиная с версии	Процессоры с архи-	
держка		OpenCV 2.3, лето 2011г.)	тектурой Intel	

Одна из наиболее известных библиотек в области компьютерного зрения – библиотека OpenCV [25, 26]. Применительно к рассматриваемым задачам, недостатком данной библиотеки является ограничение на размер изображений. Структура библиотеки OpenCV включает ряд реализаций известных методов обработки изображений, но не предлагаются какиелибо шаблонные архитектуры, настраиваемые для прикладных задач рассматриваемого в диссертации класса. В библиотеке OpenCV реализована процедура калибровки камеры для модели центральной проекции (pinhole camera model) [27]. Аппаратная поддержка на GPU для отдельных методов была реализована в версиях, начиная с 2011 г.

Другой популярной библиотекой, широко используемой для обработки изображений, является Intel Integrated Performance Primitives[28]. В отличие от OpenCVв этой библиотеке реализованы низкоуровневые алгоритмы и произведена оптимизация для архитектуры процессоров Intel.

Таким образом, вследствие выявленных ограничений доступных программных библиотек применительно к решению задач разработки систем компьютерного видения с несколькими полями зрения, актуальной является задача проектирования и реализации высокоуровневых расширяемых программных компонент для таких систем. Они должны обеспечивать решение типовых задач обработки зрительных данных, и предусматривать возможности настройки для учета ограничений конкретной прикладной задачи.

#### 1.7. Выводы

В главе сформулированы решаемые прикладные задачи обработки зрительных данных, в которых используются системы компьютерного видения с несколькими полями зрения. Анализ доступных программных средств показал, что готовых средств для решения рассматриваемых задач нет. Были выделены основные операции обработки, которые должны обеспечиваться программными компонентами в решаемых задачах:

- определение параметров камеры;
- сопоставление изображений;
- геометрическая коррекция изображений.

Анализ особенностей доступных библиотек OpenCV и IntelIPP для решения прикладных задач позволяет заключить, что для систем компьютерного видения актуальной является проблема проектирования и реализации программных компонент высокого уровня, обеспечивающих настройку с учетом ограничений конкретных прикладных задач.

# Глава 2. ПРОЕКТИРОВАНИЕ ПРОГРАММНЫХ КОМПОНЕНТ СИСТЕМ КОМПЬЮТЕРНОГО ВИДЕНИЯ С НЕСКОЛЬКИМИ ПОЛЯМИ ЗРЕНИЯ

Важным этапом обработки в рассматриваемых задачах является этап определения параметров камер. Эти параметры используются в процессе последующей обработки зрительных данных. Параметры камеры принято разделять на *внешние* и *внутренние* [23,24]. Внешние параметры связаны с положением и ориентацией камеры в системе координат (СК) сцены (эту СК будем называть мировой и обозначать МСК). Внутренние параметры определяют математическую модель проецирования точек сцены на изображение. Координаты точки сцены задаются в системе координат камеры (СКК). Начало координат СКК совпадает с центром фотографирования камеры, а одна из осей, обычно ось Z, совпадает с оптической осью). В данной работе в качестве модели камеры используется известная перспективная модель [23,24,29,30].

Для движущейся видеокамеры определение параметров взаимного положения и взаимной ориентации возможно на основе анализа *связующих точек* (соответствующих точечных признаков на последовательности изображений). Методы поиска соответствий в существенной степени зависят от решаемой задачи.

Таким образом, в качестве повторно используемых компонент систем компьютерного видения с несколькими полями зрения были выделены компоненты для решения двух задач: определение параметров системы камер и сопоставление изображений.

#### 2.1. Математическая модель камеры

Связь между координатами наблюдаемой точки в СКК и положением точки на снимке задается в виде нелинейного преобразования [23,24,29,30]

$$\vec{p} - \vec{p}_0 = -f \cdot \left[\frac{x}{z} \quad \frac{y}{z}\right]^{\mathrm{T}}$$
(2.1.1)

где  $\vec{p}$  – координаты точки на плоскости изображения (в метрических единицах),  $\vec{p}_0$  – координаты главной точки (точка пересечения оптической оси и плоскости изображения), f – фокусное расстояние камеры,  $[x \ y \ z]^{T}$  – пространственные координаты наблюдаемой точки в СКК.



Рис. 2.1.1. Геометрическая модель формирования снимка в модели перспективной проекции.

С использованием однородных координат пространственные координаты наблюдаемой точки в МСК и пиксельные координаты ее образа на растровом изображении связаны соотношением [29-31]

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \cong \begin{bmatrix} -f_x & 0 & C_x \\ 0 & f_y & C_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \mathbf{A} | \vec{T} \end{bmatrix} \cdot \begin{bmatrix} X & Y & Z & 1 \end{bmatrix}^{\mathrm{T}}$$
(2.1.2)

где  $[X \ Y \ Z]^{T}$  – координаты наблюдаемой точки в МСК,  $[u \ v]^{T}$  – пиксельные координаты образа наблюдаемой точки,  $[f_x \ f_y]$  – пиксельное фокусное расстояние камеры (устанавливает связь между размером пикселя растрового изображения по ширине/высоте и величиной фокусного расстояния в физических единицах),  $\begin{bmatrix} C_x & C_y \end{bmatrix}$  – пиксельные координаты главной точки снимка.

 $[\mathbf{A}|\vec{T}]$  – матрица перехода из МСК к СКК (внешние параметры камеры),  $\mathbf{A}: OXYZ \rightarrow Sxyz$  – матрица поворота,  $\vec{T} = \mathbf{A} \cdot (-\vec{R}_S)$  – положение начала МСК в СКК.

Существенное влияние на результаты извлечения пространственной информации по зрительным данным оказывает *дисторсия* объектива [32,33]. Для учета дисторсионных искажений в модели камеры в виде центральной проекции наиболее распространены два подхода:

 Искажение идеальной проекции [27] – заключается в деформировании спроецированных на плоскость снимка координат точек сцены

$$\vec{p}_d - \vec{p}_0 = \vec{D} \left( -f \cdot \left[ \frac{x}{z} \quad \frac{y}{z} \right]^T \right)$$
(2.1.3)

 Восстановление идеальной проекции [34] – заключается в коррекции координат наблюдаемых на снимках образов точек сцены – приведение соответствий наблюдаемых точек сцены и их образов на снимке к перспективной модели камеры

$$\vec{D}(\vec{p}_d - \vec{p}_0) = -f \cdot \left[\frac{x}{z} \quad \frac{y}{z}\right]^T$$
(2.1.4)

В выражениях (2.1.3) и (2.1.4)  $\vec{p}_d$  – координаты наблюдаемой (и искаженной вследствие дисторсии) точки на изображении относительно его центра,  $\vec{D}$  – вектор-функция, определяющая модель дисторсии [32-34].

В качестве математической модели дисторсии в работе использовалась модель Брауна [34], получившая широкое распространение [25,30]. Она заключается в учете радиальных искажений, влияние которых увеличивается с удалением от центра изображения, и тангенциальных искажений, возникающих вследствие неперпендикулярности оптической оси и плоскости матрицы камеры. Зависимость смещенных наблюдаемых координат точки  $\vec{\rho} = [\rho_x \quad \rho_y]^T$  и координат после компенсации дисторсии в этой модели будем описывать выражением:

$$\vec{D}(\vec{\rho}) = \vec{\rho} + \vec{\rho} \cdot \sum_{i=1}^{N} k_i \cdot \|\vec{\rho}\|^{2\cdot i} + \begin{bmatrix} \|\vec{\rho}\|^2 + 2\rho_x^2 & 2 \cdot \rho_x \rho_y \\ 2 \cdot \rho_x \rho_y & \|\vec{\rho}\|^2 + 2\rho_y^2 \end{bmatrix} \cdot \begin{bmatrix} t_1 \\ t_2 \end{bmatrix}$$
(2.1.5)

 $k_i$  – параметры радиальной дисторсии (обычно не более трех параметров),  $t_i$  – параметры тангенциальной дисторсии.

Таким образом, для проекции наблюдаемой точки сцены из МСК на плоскость изображения можно записать следующее выражение

$$\begin{bmatrix} u \\ v \end{bmatrix} = \vec{P}_D(\vec{\gamma}, \vec{R})$$
(2.1.6)

 $\vec{\gamma}$  – вектор параметров съемки (включающих внешние и внутренние параметры, а именно матрицы перехода от МСК к СКК и параметров камеры),  $[u \ v]^T$  – пиксельные координаты наблюдаемой точки,  $\vec{R} = [X \ Y \ Z]^T$  – координаты наблюдаемой точки в МСК,  $\vec{P}(\bullet, \bullet) : \mathbf{R}^3 \to \mathbf{R}^2$  – проекционное отображение.

### 2.2. Минимизация ошибки перепроецирования

Рассмотрим некоторую сцену из *N* точек K = { $\vec{R}_1,...\vec{R}_i,...\vec{R}_N$ }, координаты которых заданы в MCK *OXYZ* (см. рис. 2.2.1). Допустим, производится съемка сцены с помощью камеры *M* раз с различными параметрами съемки  $\Gamma = {\vec{\gamma}_1,...\vec{\gamma}_j,...\vec{\gamma}_M}$  (внутренние и внешние параметры камеры меняются в процессе съемки)



Рис. 2.2.1. Получение изображений сцены с нескольких точек наблюдения.

Определим функцию многих переменных, устанавливающую связь между координатами наблюдаемых точек, параметрами съемки и образами точек на изображениях

$$\vec{F}(\vec{\gamma}_{1},...\vec{\gamma}_{i},...,\vec{R}_{1},...\vec{R}_{j},...) = \sum_{i} \sum_{j} \left\| \begin{bmatrix} u_{ij} \\ v_{ij} \end{bmatrix} - \vec{P}_{D}(\vec{\gamma}_{i},\vec{R}_{j}) \right\|^{2}$$
(2.2.1)

где  $\begin{bmatrix} u_{ij} & v_{ij} \end{bmatrix}^{T}$  – пиксельные координаты *i*-ой точки сцены, наблюдаемой с помощью камеры с *j*-м набором параметров съемки.

Неизвестными параметрами в (2.2.1) являются набор параметров съемки и координаты точек наблюдаемых сцены в МСК. Их можно определить с помощью численного метода путем минимизации суммарной ошибки перепроецирования (2.2.1). Под ошибкой перепроецирования понимается квадрат разности между координатами образа точки сцены  $\begin{bmatrix} u_{ij} & v_{ij} \end{bmatrix}^{T}$  на изображении камеры и координат проекции точки  $K_{i}$  на ту же плоскость, вычисленными с использованием параметров съемки  $\bar{\gamma}_{i}$ 

$$\sum_{i} \sum_{j} \left\| \begin{bmatrix} u_{ij} \\ v_{ij} \end{bmatrix} - \vec{P}_D(\vec{\gamma}_i, \vec{R}_j) \right\|^2 \to \min \qquad (2.2.2)$$

Анализ опубликованной литературы [23,24,29,30,38] показал, что для численного решения задачи минимизации ошибки перепроецирования обычно применяется метод Гаусса-Ньютона и его модификация – Левенберга-Марквардта [35-37].

#### 2.3. Классификация задач определения параметров камеры

Задачи определения параметров камеры можно разделить на несколько типов, в зависимости от того, какие параметры известны и какие требуется определить (см. рис. 2.3.1). Все эти задачи можно решить на основе минимизации ошибки перепроецирования [29, 30, 38]. В качестве основных типов таких задач можно выделить следующие:

- Калибровка камеры определение внутренних параметров камеры (фокусное расстояние, координаты главной точки, параметры дисторсии). Выполняется при известных координатах наблюдаемых точек сцены, при этом в процессе минимизации определяются/уточняются внешние параметры камеры для каждого снимка;
- Внешняя калибровка определение внешних параметров камеры (матрица перехода из МСК в СКК), выполняется при известных координатах наблюдаемых точек сцены и известных параметрах камеры;
- Фототриангуляция определение внешних параметров камеры каждого снимка и координат наблюдаемых точек сцены (восстановление структуры сцены), выполняется для заданных точечных соответствий на снимках;

Самокалибровка – определение внешних и внутренних параметров камеры каждого снимка и координат наблюдаемых точек сцены (восстановление структуры сцены). Выполняется при условии, что между снимками заданы точечные соответствия.



Рис. 2.3.1. Классификация задач определения параметров камеры в зависимости от входных и выходных данных.

В задачах восстановления структуры сцены по соответствующим точкам (фототриангуляция и самокалибровка) может использоваться ограниченный набор известных точек сцены – *опорных точек* (ground control point или GCP) для получения решения в окрестности требуемого минимума [24].

Для решения задачи минимизации ошибки перепроецирования требуется задание начальных параметров [38]. В зависимости от решаемой прикладной задачи существуют различные подходы к оценке начального приближения:

 для калибровки камеры существуют как средства оценки параметров с использованием лабораторных установок, например [39], так и аналитические методы на основе съемки метрологических стендов и полигонов [27, 30, 34, 40-41];

- для задачи внешней калибровки параметры матрицы перехода могут оцениваться с помощью натурных измерений, также существуют аналитические методы решения этой задачи [43,44];
- для фототриангуляции параметры матрицы перехода оцениваются с использованием датчиков GPS и навигационных систем, также для взаимного ориентирования камер по двум снимкам существуют методы оценки на основе набора точечных соответствий [29,45-47];

На рисунке 2.3.1 для предложенной классификации задач определения параметров камеры показаны те прикладные задачи (рассматриваемые в первой главе), которые задействуют выделенные классы.

# 2.4. UML-диаграмма классов компоненты для определения параметров камеры

На основе проведенного анализа можно выделить две основные особенности разрабатываемой программной компоненты: (1) оценка начальных параметров определяется спецификой прикладной задачи, (2) уточнение параметров является повторяемой процедурой, для которой может быть разработана унифицированная схема.

На рисунке 2.4.1 показана структурная UML-диаграмма классов компоненты для определения параметров камеры.

Класс CGaussNewtonMethod и дочерний от него класс CLevenberg-Method выполняют процедуру минимизации вектор-функции, переопределяемой через абстрактный класс I\_DiffVectorFunction. Для метода Левенберга-Марквардта, в классе CLevenbergMethod, переопределен способ приращения в итерационном процессе с помощью виртуальной функции-член step.



Рис. 2.4.1. UML-диаграмма программной компоненты для определения параметров камеры.

Класс CExtCalibration используется для решения задачи внешней калибровки, a CPhotoTriangulation – для задачи фототриангуляции. Поскольку для каждого из этих классов (наследуемых от абстрактного класс I DiffVectorFunction) не требуется конкретизировать математическую мореализована через абстрактный интерфейс дель камеры, то она I CameraModel. Этот интерфейс содержит два метода: формирование проецирующего луча в СКК для заданного пикселя изображения и вычисление ошибки перепроецирования между точкой сцены (заданной в СКК) и наблюдаемым образом этой точки на изображении.

Для цифровых камер общего назначения (класс CDigitalCamera, унаследованный от класса I\_CameraModel), процедура калибровки параметров камеры и самокалибровки представлены в классах CCameraCalibration и CSelfCalibration (классы является унаследованными от абстрактного класса I DiffVectorFunction).

Учет дисторсии реализован методом восстановления идеальной проекции (2.1.4) с помощью класса CBrownDigitalCamera, использующим множественное наследование от классов CDigitalCamera и CBrownDistortion. Последний класс реализует расчет координат пикселей с использованием модели дисторсии Брауна (2.1.5).

## 2.5. Сопоставление изображений на основе признаковых методов

Общую схему признаковых методов сопоставления изображений (поиска связующих точек) можно описать следующей последовательностью действий [48]:

- обнаружение характерных точек (низкоуровневых признаков) на изображении;
- 2) формирование дескрипторов окрестностей признаковых точек;
- 3) сопоставление дескрипторов на основе принятой метрики;
- 4) фильтрация соответствий на основе априорной модели.

Под характерной точкой изображения  $\vec{p}$  с окрестностью  $N_p$  будем понимать область изображения, которую можно явно идентифицировать в рассматриваемой локальной области изображения, т.е.  $\{\forall \vec{r}: 0 < \|\vec{p} - \vec{r}\| < \varepsilon \rightarrow d(N_p, N_r) > \Delta\}$ , где  $d(\bullet, \bullet)$  – метрика сопоставления окрестностей [49]. В настоящее время для поиска характерных точек на изображениях разработано большое количество методов, например, на основе анализа дисперсии окрестности, анализа локальной кривизны контуров и другие [50-52].

Таким образом, первые два этапа сопоставления изображений выполняют попиксельную обработку изображений. Выбор метода обнаружения характерных точек и вид дескриптора окрестности во многом определяется спецификой прикладной задачи (качеством изображений, особенностями наблюдаемых сцен и др.). Среди часто используемых дескрипторов можно назвать оконные дескрипторы (в виде прямоугольных фрагментов изображения), инвариантные моментные характеристики, дескрипторы на основе гистограмм ориентаций градиента яркости, на основе разложений Хаара и другие [50,53-57].





На заключительном этапе сопоставления пары изображений производится фильтрация *выбросов* (неправильно найденных соответствий) с учетом некоторой модели ограничений (см. рис. 2.5.1). Распространенным подходом для решения этой задачи является метод RANSAC и его модификации [29,58,59]. В основе этого метода лежит критерий наилучшего согласования всего множества пар с выбранной моделью, параметры которой оцениваются по случайным выборкам из всего множества пар. Таким образом, качество найденных соответствий между признаками разных изображений зависит от выбора модели физических ограничений.

Для корректной работы этого метода требуется выбор адекватной модели, описывающей допустимую возможность соответствия найденных

пар так, чтобы количество параметров модели было минимально (большее количество параметров требует большего количества итераций для достоверной оценки параметров модели) [29].

Рассмотрим ряд моделей, используемых для фильтрации выбросов (введем обозначения и допущения:  $\{\vec{p}_1, \vec{p}_2\}$  – пара соответствий на снимках, точки заданы в однородных координатах, модели камер неизвестны, дисторсия слабо выражена и ею можно пренебречь):

- фундаментальная матрица  $\vec{p}_2^T \cdot F \cdot \vec{p}_1 = 0$  применяется, при наблюдении статической сцены общего вида [29,60];
- гомография *p*<sub>2</sub> = *H* · *p*<sub>1</sub> применяется при допущении, что центры фотографирования для двух объективов совпадают (вращательная модель движения камеры) либо наблюдаемую сцену можно аппроксимировать плоскостью [29,61];
- эпиполюс (*p*<sub>1</sub> × *p*<sub>2</sub>) · *e* = 0 применяется при допущении, что матрица поворота между снимками единичная (плоскопараллельная модель движения камеры) [29].

#### 2.6. Сопоставление аэрофотоснимков

Для сопоставления аэрофотоснимков, получаемых в процессе полета самолета, мало подверженного внешним возмущениям атмосферы (траектория полета стабильна, угловые колебания самолета в пределах нескольких градусов, перекрытие снимков не менее 50%), применение находят методы на основе площадной корреляции [62,63]. В целом можно выделить основные особенности исходных данных, которые учитываются в алгоритмах обработки таких снимков:

- наличие навигационных данных для каждого снимка;
- перекрытие снимков не менее 50%;
• значительные размеры изображений (до 2Гб).

Учитывая специфику поставленной в первой главе задачи – наклонной аэрофотосъемки, а также с учетом обзора, выполненного в работе [63] применялся следующий алгоритм сопоставления:

 Для обнаружения низкоуровневых признаковых на изображении применялся детектор угловых точек Харриса [51], основанный на поиске окрестностей, локальная кривизна в которых в анализируемом окне максимальна

$$R = \det M + k \cdot (traceM)^{2}$$
(2.6.1),  
$$M = \sum_{x,y} w(x,y) \begin{bmatrix} I_{x}^{2} & I_{x}I_{y} \\ I_{x}I_{y} & I_{y}^{2} \end{bmatrix}$$
(2.6.2),

R – величина отклика – характеризует локальную кривизну функции яркости изображения в рассматриваемой окрестности, k = 0.04 – экспериментально определяемый коэффициент,  $\nabla I = \begin{bmatrix} I_x & I_y \end{bmatrix}^T$  – градиент изображения  $I : \mathbb{R}^2 \to \mathbb{R}$ , w(x, y) – взвешивающая функция.



Рис. 2.6.1 Пример обработки красного канала изображения (аэрофотоснимка) с помощью детектора Харриса (карта "откликов" представлена в псевдоцветах: ярче - сильнее).

 В качестве дескриптора характерной точки был выбран оконный дескриптор (см. рис. 2.6.2)



Рис. 2.6.2 Пример дескриптора типа "окно изображения".

 В качестве метрики сопоставления дескрипторов использовалась кросс-корреляция [64]

$$C(\vec{q}, \vec{p}) = \left\langle \frac{\vec{q}^{*}}{\|\vec{q}^{*}\|}, \frac{\vec{p}^{*}}{\|\vec{p}^{*}\|} \right\rangle$$
(2.6.3),

где верхний индекс «\*» применялся для обозначения вектора, элементы которого вычисляются в соответствии с выражением

$$q_k^* = q_k - \frac{1}{d} \sum_{i=1}^d q_i, \ d = \dim \vec{q}, k = 1...d$$
 (2.6.4)

Если заданы два некоторых множества векторов  $Q = \{\vec{q}_1, \vec{q}_2, \vec{q}_3 \dots \vec{q}_N\}$ и  $P = \{\vec{p}_1, \vec{p}_2, \vec{p}_3 \dots \vec{p}_M\}, \vec{q}, \vec{p} \in \mathbb{R}^d$ , то будем считать, что вектор  $\vec{p}_k$ соответствует вектору  $\vec{q}_i$ , при условии

$$\begin{cases} k = \underset{j=1...M}{\operatorname{arg\,max}} C(\vec{q}_i, \vec{p}_j) \\ C(\vec{q}_i, \vec{p}_k) > [C] \end{cases}$$
(2.6.5)

 При аэрофотосъемки наблюдаемую область местности можно считать плоской, таким образом, в качестве априорной модели фильтрации выбросов применима модель гомографии [63].



Рис. 2.6.3 Результат сопоставления двух аэрофотоснимков. Зелеными крестообразными маркерами показаны признаковые точки, для которых были обнаружены соответствия.

Одним из типичных приемов повышения скорости обработки аэрофотоснимков является использование пирамидального представления изображения [65].

#### 2.7. Сопоставление снимков с БПЛА

В отличие от аэрофотосъемки, легкие БПЛА, как правило, сильно подвержены возмущениям атмосферы. Таким образом, можно обозначить ряд особенностей фотосъемки с БПЛА:

- перекрытие между снимками может отсутствовать, что исключает возможность построения фотосхем (для рассматриваемого в первой главе комплекса мониторинга местности с использованием БПЛА "Типчак" съемка выполнялась в режиме 30% перекрытия фотоснимков);
- возможны существенные изменения ракурса съемки (направления оптической оси камеры);
- низкая точность навигационных данных GPS, не позволяющая учитывать их при задаче сопоставления при некоторых воздействия внешних атмосферных условий (ветер).

Для задачи сопоставления снимков с БПЛА были применены алгоритмы на основе признаков SIFT и SURF [54,55,66]. Для выбора характерных точек, соответствующим различным размерам окна (или уровням детализации) был применен анализ изображений с использованием масштабного пространства [67,68].

Под масштабным пространством изображения (МПИ) будем понимать набор (рис. 2.7.1) изображений, имитирующих удаление точки наблюдения от объекта наблюдения и удовлетворяющих уравнению диффузии [67,68]

$$\frac{\partial L}{\partial t} = \frac{1}{2} \nabla^2 L \qquad (2.7.1)$$

где L(x, y; t) = I(x, y) \* g(x, y; t) – изображение сглаженное фильтром Гаусса (одно из изображений набора МПИ), определяется на основе операции свертки изображения  $I: \mathbb{R}^2 \to \mathbb{R}$  с ядром ( $t = \sigma^2$  – дисперсия, рассматривается как масштабный параметр)

$$g(x, y; t) = \frac{1}{2\pi \cdot t} e^{-(x^2 + y^2)/2t}$$
(2.7.2)



Рис. 2.7.1. Пример изображений, входящих масштабного пространства изображения ния

Таким образом, для сопоставления снимков с БПЛА применялся следующий алгоритм [54]:  Поиск характерных точек на МПИ. Для этого на МПИ применяется оператор Лапласа (аппроксимация, обеспечивающая формирование многомасштабной карты откликов, см. рис. 2.7.2)

$$DoQ(x, y; t_i) \approx L(x, y; t_{i+1}) - L(x, y; t_i)$$
 (2.7.3),



Рис. 2.7.2. Пример многомасштабной карты краев

2) Поиск локальных экстремумов на массиве изображений лапласианов и их субпиксельное уточнение на МПИ [54].



Рис. 2.7.3. Схема формирования гистограммного дескриптора HoG и определение доминирующей ориентации для характерной точки.

3) В качестве дескрипторов характерных точек использовалась гистограмма ориентаций векторов градиентов яркости (histogram of gradients, HoG, рис. 2.7.3). Для достижения инвариантности к повороту для каждой характерной точки назначался доминирующей вектор градиента (см. рис. 2.7.3);

4) В качестве метрики сопоставления использовался метод ближайшего соседа, т.е. вектор  $\vec{P}_{k_1}$  считается соответствующим вектору  $\vec{q}_i$ , при условии

$$D(\vec{q}_{i}, \vec{p}_{k_{1}}) = \frac{\left\|\vec{q}_{i} - \vec{p}_{k_{1}}\right\|}{\left\|\vec{q}_{i} - \vec{p}_{k_{2}}\right\|} > [D]$$

$$\begin{cases} k_{1} = \underset{j=1...M}{\operatorname{arg\,min}} \left\|\vec{q}_{i} - \vec{p}_{j}\right\| \\ k_{2} = \underset{j=1...M, j \neq k_{1}}{\operatorname{arg\,min}} \left\|\vec{q}_{i} - \vec{p}_{j}\right\| \end{cases}$$
(2.7.4) W
$$(2.7.5)$$



Рис. 2.7.4 Результат сопоставления двух снимков с БПЛА. Маркерами зеленого цвета показаны найденные соответствующие признаковые точки.

 Наблюдаемую область местности можно аппроксимировать плоскостью, и в качестве априорной модели фильтрации выбросов применима модель гомографии [63].

### 2.8. UML-диаграмма классов компоненты для сопоставления изображений

На рис. 2.8.1 представлена структурная UML-диаграмма классов программной компоненты сопоставления изображений. Абстрактный

класс I\_Correlator реализует последовательную схему обработки набора снимков:

- 1) создание коррелятора и инициализация параметров;
- 2) обработка очередного кадра;
- 3) обновление текущих параметров коррелятора;
- 4) получить массив соответствий.



Рис. 2.8.1 UML-диаграмма программной компоненты для сопоставления изображений.

В классе I\_FeatureCorrelator (унаследованном от I\_Correlator) переопределен метод обработки снимка ProcessShot и введены следующие абстрактные виртуальные методы:

 FindKeyPoints – поиск характерных точек и формирование массива дескрипторов характерных точек

- Matching сопоставление дескрипторов, с возможностью использования бинарной матрицы физических ограничений (для ее определения существует метод PhysicalLimitsMask);
- Filtering фильтрация найденных соответствий на основе модели ограничений (переопределяемая для класса-интерфейса I\_TransformModel);

4) Tracking – прослеживание соответствий на паре снимков.

Для двух описанных схем корреляции реализованы классы CHarris-Correlator и CSIFTCorrelator (унаследованные от абстрактного класса I\_FeatureCorrelator).

Метод RANSAC реализован в классе CFiltering как статическая функция, этот метод используется в функции-члене Filtering класса I\_FeatureCorrelator.

Рассмотренные модели ограничений реализованы в классах CFundamentalMatrix (фундаментальная матрица), CHomography (гомография) и CEpipole (эпиполюс). Эти классы унаследованы от абстрактного класса I\_TransformModel. Класс I\_TransformModel содержит четыре абстрактные виртуальные функции:

- получение минимального размера выборки для определения параметров преобразования;
- 2) получение матрицы преобразования;
- 3) вычисление параметров преобразования;
- 4) оценка согласованности преобразования на наборе соответствий.

#### 2.9. Выводы

С целью проектирования программной компоненты для определения параметров камеры была рассмотрена математическая постановка задачи минимизации ошибки перепроецирования. Были выделены различные классы прикладных задач в зависимости от количества и типов неизвестных параметров. Описанная классификация применима для решения прикладных задач обработки зрительных данных, описанных в первой главе.

Для задачи сопоставления изображений аэрофотосъемки рассмотрена общая схема на основе признаковых методов. Применительно к задаче сопоставления аэрофотоснимков, формируемых при различных физических условиях, рассматриваются два алгоритма сопоставления. В четвертой главе на основе описанных алгоритмов будут разработаны новые алгоритмы для решения прикладных задач обработки зрительных данных, описанных в первой главе.

Для каждой из рассмотренных программных компонент спроектированы UML-диаграммы классов. При этом зависящие от предметной области требования прикладной задачи учитываются посредством механизмов объектно-ориентированного программирования.

## Глава 3. ПАРАЛЛЕЛЬНЫЕ РЕАЛИЗАЦИИ АЛГОРИТМОВ СОПОСТАВЛЕНИЯ ИЗОБРАЖЕНИЙ

Одним из распространенных подходов к оптимизации вычислительного процесса в реализации алгоритмов сопоставления изображений является использование аппаратных средств, обеспечивающих параллельное исполнение алгоритмов. В применении этого подхода в прикладных задачах можно выделить два основных аспекта:

- выбор архитектуры аппаратных средств с учетом способов представления и возможностей параллельной обработки данных;
- разработка и оптимизация алгоритмов применительно к выбранной архитектуре аппаратного обеспечения.

Для обработки изображений, данные которых допускают матричное представление, в последнее время широко применяются графические процессоры общего назначения (General Purpose Graphic Processor Unit, GP GPU) [69-71].

В данной главе описываются новые параллельные реализации алгоритмов сопоставления аэрофотоснимков на основе подходов, рассмотренных во второй главе. Производится сравнение предложенных реализаций с существующими аналогами.

# 3.1. Программно-аппаратная архитектура параллельных вычислений на основе графических процессоров NVidia

Возможности программирования GPU общего назначения делают их привлекательными для применения в задачах компьютерного зрения с целью ускорения обработки и снижения нагрузки на центральный процессор вычислительной системы. Одной из распространенных технологий программирования GPU является NVidia CUDA [69-71]. В ее основе лежит унифицированная архитектура семейства графических процессоров NVidia GT200/Fermi и модель программирования с применением Си-подобного языка.



Рис. 3.1.1. Схема поточно-параллельной обработки данных на GPU архитектуры NVidia.

Архитектура современных GPU рассматриваемого семейства – унифицирована. Обработку данных с использованием технологии CUDA в общем виде можно представить в виде поточно-параллельного конвейера (см. рис. 3.1.1):

- Поток данных разбивается на множество одинаковых по размеру блоков данных. Таким образом формируется обрабатываемый поток блоков;
- GPU содержит фиксированный набор ядер (количество ядер зависит от модели GPU). Каждое ядро является мультипроцессорным вычислителем;

- Ядро GPU обрабатывает некоторый (очередной) блок данных (см. рис. 3.1.1). Данные блока на ядре GPU обрабатываются параллельно;
- 4) Поток блоков обрабатывается последовательно-параллельно на GPU (рис. 3.1.1). Механизм управления загрузкой ядер, в частности, назначение обрабатываемых блоков исполняющим ядром GPU скрыт от программы прикладного уровня.



Рис. 3.1.2. Представление обработки данных на GPU: сетка из блоков нитей.

Модель программирования GPU CUDA предполагает, что все данные представляются в виде сетки из блоков нитей (grid of thread blocks, рис. 3.1.2). Программирование GPU заключается в написании программной реализации для одной функции выполняемой всеми нитями CUDA. Для модели программирования CUDA можно выделить следующие основные выполняемые операции [69]:

- 1) выделение памяти на GPU;
- 2) копирование данных из основной памяти (память CPU) в выделенную память на GPU;
- 3) вызов функций, исполняемых на GPU;
- 4) копирование результатов обработки из памяти GPU в область памяти CPU
- 5) освобождение памяти на GPU.

В данной модели существенные временные затраты связаны с копированием данных по шине, к которой подключены GPU, – шине PCI-Express. Уменьшение количества вызовов операций копирования между памятью CPU↔GPU является типичным приемом ускорения выполнения алгоритмов [69]. Как правило, значительное ускорение возможно при изначальном проектировании алгоритмов с учетом возможностей GPU, а также преимущества CPU над GPU при реализации итерационных (последовательных) алгоритмов.

В технологии CUDA выделено несколько уровней памяти GPU [69]. Повышение быстродействия достигается за счет эффективного сочетания различных уровней:

- Наиболее быстродействующей является регистровая память (register memory, объем 64Кб), которая имеется у каждого процессора ядра (мультипроцессорного вычислителя). Время жизни регистровой памяти определяется временем жизни нити CUDA.
- 2) Следующей по быстродействию и также ограниченной по размерам (16Кб для GPU GT200 и 48Кб для GPU Fermi) является разделяемая память (shared memory). Такая память имеется у каждого ядра GPU, время жизни в течение выполнения блока нитей.

- Константная память (constant memory) быстрая кэшируемая память GPU, ограниченная 64Кб. Доступ к этой памяти возможен из любой нити блока только для операций чтения данных.
- 4) Текстурная память (texture memory) память, оптимизированная для задач текстурирования (наложение картинок на полигональные модели). CUDA предполагает механизм доступа и кэширования основной памяти GPU заранее зарезервированной программистом как текстурной. Доступ к этой память возможен только с применение специальных типов данных и функций API CUDA.
- 5) Наименее быстродействующая медленная память это основная память GPU.

При проектировании новых реализаций алгоритмов сопоставления изображений на основе технологии CUDA использовались следующие основные приемы и требования, перечисленные в порядке приоритетности:

- 1) оптимальное использование доступного объем памяти GPU, определяемого размером входных данных;
- 2) минимизация обмена данных по шине PCI-Express;
- оптимизация алгоритмов на основе совместного использования СРU и GPU;
- минимизация количества тактов обращения к данным, за счет выравнивания памяти, используемой для хранения матричных данных;
- 5) использование высокоскоростной разделяемой памяти и кэшируемой константной памяти.

### 3.2. Параллельная реализация алгоритма сопоставления аэрофотоснимков

Рассматривавшийся в п. 2.6 алгоритм сопоставления аэрофотоснимков на основе определения пар соответствующих точечных признаков относится к широко распространенному семейству алгоритмов [48,63]. Алгоритмы такого типа относительно несложно реализовать для исполнения на СРU ПК общего назначения. Однако для оперативной обработки аэрофотоснимков, имеющих значительные размеры, актуально построение параллельных реализаций таких алгоритмов.

Анализ опубликованных работ показал, что известно несколько вариантов реализаций детекторов угловых признаков, ориентированных на различные аппаратные архитектуры [72-75]. Среди опубликованных реализаций имеется и вариант для исполнения на GPU [73]. В библиотеке OpenCV 2.3 (2011г.) содержится открытая реализация детектора Харриса с использованием технологии CUDA (сравнение библиотечной реализации с новым параллельным алгоритмом будет представлено в п. 3.3).

В данном параграфе описывается новая параллельная реализация, которая включает следующие этапы:

- 1) поиск точечных особенностей с помощью детектора Харриса;
- переход от матричного представления характерных точек к линейному массиву координат;
- 3) формирование массива дескрипторов типа "окно изображения";
- 4) сопоставление двух массивов дескрипторов на основе метрики кросс-корреляции.

Существенной особенностью предложенной реализации является оптимизация обработки данных, объемы которых сравнимы с объемом доступной памяти на GPU. <u>Параллельная реализация детектора Харриса</u> с учетом особенностей архитектуры NVidia CUDA была выполнена с применением следующих типов данных:

- Входное изображение представляется в виде одноканального изображения (отдельный цветовой канал цветного изображения или полутоновое изображение). Данные хранятся в памяти GPU в матричном виде с элементами типа integer (4байта для хранения целого числа).
- Матрица значений откликов детектора Харриса содержит вещественные элементы типа float (4байта для хранения числа с плавающей точкой). Размеры этой матрицы, вычисляемой за один проход, совпадают с размерами входного изображения;
- 3) Матрица локальных максимумов для матрицы откликов содержит элементы типа integer (4байта для хранения целого числа).

Таким образом, в зависимости от размера входного изображения можно определить объемы памяти RAM GPU (в байтах), достаточной для обработки изображения детектором Харриса

$$M_{GPU} \ge H \cdot W \cdot 12 + \delta \tag{3.2.1}$$

где  $M_{GPU}$  – количество байт доступных в памяти RAM GPU,  $\{W, H\}$  – размеры изображения в пикселях (ширина/высота),  $\delta$  – количество байт, требуемых для выравнивания строк изображения в памяти (для уменьшения количества таков обращения к данным).

При использовании однобайтного представления пикселей для входного изображения и матрицы локальных максимумов, требования к необходимой памяти RAM GPU могут быть значительно снижены

$$M_{GPU} \ge H \cdot W \cdot 6 + \delta \tag{3.2.2}$$



Рис. 3.2.1. Схема обработки растровых данных в параллельной реализации детектора Харриса.

Формирование матрицы откликов осуществляется без применения дополнительных буферов за счет использования разделяемой памяти (рис. 3.2.1):

 Копирование блока растрового изображения с перекрытием в 2 пикселя на границе CUDA-блока в разделяемую память. Для копирования пикселей блока изображения в разделяемую память CUDA-нитям назначаются соответствующие пиксели, из условия что количество пикселей, считываемых каждой CUDA-нитью не более

$$N_{th} = ceil\left[\frac{(B_x + 4) \cdot (B_y + 4)}{B_x \cdot B_y}\right]$$
(3.2.3)

где  $N_{th}$  – максимальное количество пикселей, считываемое CUDA-нитью,  $\{B_x, B_y\}$  – количество CUDA-нитей в блоке (по ширине/высоте), *ceil*(•) – оператор округления дробного числа до большего целого числа.



Рис. 3.2.2. Двух этапное вычисление градиента (оранжевым цветом показаны вычисляемые пиксели на каждом из этапов).

- 2) Вычисление градиента (свертка с маской Собела размерами 3х3 [64]) для всех пикселей маски, отмеченной на рис. 3.2.1 пурпурным цветом (за исключением пикселей, лежащих на границе считанного блока и отмеченных синим цветом на рис. 3.2.1). Поскольку количество CUDA-нитей меньше количества обрабатываемых пикселей, то вычисление выполнялось в два этапа (рис. 3.2.2): внутри и снаружи блока.
- Вычисление отклика с учетом взвешивающего гауссовского ядра размерами 3х3 [64] (рис 3.2.1).

Операция поиска локальных экстремумов выполняется также с использованием разделяемой памяти. В этом случае разделяемая память применяется для повышения производительности. Схема выполнения этой операции представлена на рис. 3.2.3.



Рис. 3.2.3. Схема параллельной реализации операции поиска локального максимума в матрице откликов детектора Харриса.



Рис. 3.2.4. Пример преобразования массива экстремумов из матричного представления в линейное

<u>Преобразование массива локальных экстремумов из матричного</u> <u>представления к линейному</u> показан на рис. 3.2.4. При выполнении этой операции производятся следующие основные действия (при этом обработка может выполняться как по строкам, так и по столбцам):

- Подсчет количества экстремумов в каждой строке, при этом в каждой строке для каждого экстремума назначается его порядковый индекс (в пределах этой строки).
- Для сформированного столбца сумм в строках выполняется расчет интегральной суммы средствами центрального процессора (CPU).
- Матрица, содержащая индексированные экстремумы в строках, суммируется с вектором интегральных сумм (построчно). В результате для каждого экстремума назначается его порядковый индекс в линейном массиве.
- Преобразованием из матричного представления к линейному посредством параллельного чтения.

Таким образом, объем памяти GPU для формирования линейного массива определяется соотношением

$$W \cdot H \cdot (E_l + E_{idx}) + M_{pts} + \delta \le M_{GPU}$$
(3.2.4),

где  $E_l$  – количество байт, требующихся для хранения единичного элемента матрицы локальных максимумов,  $E_{idx}$  – количество байт, требующихся для хранения единичного элемента матрицы индексированных маркеров (см. рис. 3.2.4),  $M_{pts}$  – количество байт, требующихся для хранения линейного массива координат характерных точек,  $\delta$  – количество байт, требующихся для выравнивания строк матричных данных в памяти.



Рис. 3.2.5. Схема операции формирования массива дескрипторов типа "окно изображения".

Схема операции <u>параллельного формирования массива дескрипторов</u> <u>типа "окно изображения"</u> показана на рис. 3.2.5. Для реализации этой операции объем доступной памяти GPU должен удовлетворять соотношению

$$(W \cdot H + N \cdot D) \cdot E_{img} + M_{pts} + \delta \le M_{GPU}$$
(3.2.5),

где N – количество найденных характерных точек, D – размер дескриптора в байтах (количество пикселей, которое содержится в "окне изображения", см. рис. 3.2.5).

Далее выполняется <u>параллельное сопоставление дескрипторов на ос-</u> нове метрики кросс-корреляции, представленное на рис. 3.2.6. Существенной особенностью предлагаемой схемы является возможность учета физических ограничений:

- Выполняется нормировка дескрипторов (на основе предварительного вычисления среднего значения для каждого дескриптора, см. п. 2.6.3, 2.6.4).
- Вычисление матрицы расстояний в виде матрицы скалярных произведений (с учетом матрицы физических ограничений).
- Поиск наилучших соответствий (максимальных значений метрики кросс-корреляции) в строках/столбцах среди физически допустимых соответствий.
- Выполнение пороговой обработки (бинаризация) с использованием допустимого порога соответствий.



Рис. 3.2.6. Операция сопоставления дескрипторов точечных признаков на основе кросс-корреляции.

Объем доступной памяти GPU для реализации параллельного сопоставления дескрипторов определяется на основе соотношения

$$(N_1 + N_2) \cdot (D + D^*) + N_1 \cdot N_2 \cdot (E_D + E_P + E_B) + \delta \le M_{GPU}$$
 (3.2.6),

где  $\{N_1, N_2\}$  – количество дескрипторов характерных точек двух изображений найденных характерных точек,  $D^*$  – размер центрированного и норми-

рованного дескриптора в байтах (см. рис. 3.2.6),  $E_D$  – количество байт, требующихся для хранения единичного элемента матрицы скалярных произведений,  $E_P$  – количество байт, требующихся для хранения единичного элемента матрицы физических ограничений,  $E_B$  – количество байт, требующихся для хранения единичного элемента бинарной матрицы соответствий (см. рис. 3.2.6).

В предложенной новой параллельной реализации алгоритма сопоставления аэрофотоснимков основной процессор задействуется только для итеративного вычисления интегральных сумм, при переходе от матричного представления данных к линейным массивам. В результате практически все вычисления выполняются аппаратно и обмен данными по шине PCI-Express незначителен.

### 3.3. Оценка производительности параллельной реализации детектора Харриса на GPU

Сравнение предложенной реализации детектора Харриса и открытых реализаций на СРU и GPU, доступных в библиотеке OpenCV 2.3 (2011 г.) показано на рис. 3.3.1. Синий график соответствует предложенной реализации, красный – реализации OpenCV (GPU), зеленый – реализации OpenCV (CPU). По оси абсцисс показан размер изображения по ширине (высота принята равной ширине), по оси ординат – время обработки в мс.



Рис. 3.3.1. Сравнение реализаций детектора Харриса при размерах изображения от 500х500 до 8000х8000 пикселей.



Рис. 3.3.2. Сравнение реализаций детектора Харриса на GPU при размерах изображения от 500х500 до 2000х2000 пикселей.

Результаты экспериментов позволяют заключить, что реализация OpenCV (CPU) значительно уступает в производительности реализациям на GPU: время обработки данных на CPU экспоненциально возрастает в зависимости от размера изображения. Характеристики производительности реализаций на GPU соизмеримы: реализация OpenCV задействует текстурную память, тогда как собственная реализация построена на основе использования выровненной памяти RAM GPU (cudaMallocPitch – функция для выделения такой памяти [63]). Из рис. 3.3.2 следует, что предложенная реализация несколько превосходит по скорости реализацию на OpenCV(GPU), при размерах изображений менее 1500х1500 пикселей. Ограничением библиотеки OpenCV, критичным для рассматриваемых прикладных задач, является ограничение размеров обрабатываемых изображений до 7000x7000 пикселей. Кроме того, в предложенной реализации вычисления выполняются с учетом взвешивания с гауссовским ядром. При отдельном выполнении этой операции возникают дополнительные накладные расходы копирования данных в разделяемую память и двухэтапного вычисления градиента (рис. 3.2.2).

#### 3.4. Параллельная реализация алгоритма сопоставления снимков с БПЛА

Для сопоставления снимков с БПЛА в п. 2.7 был предложен алгоритм на базе детектора признаковых точек с гистограммными дескрипторами SIFT Д.Лова [54]. Сложность применения этого подхода на практике, и в особенности, в системах реального времени, связана с существенными вычислительными затратами на предобработку и построение дескрипторов. По этой причине применение GPU для реализации алгоритмов сопоставления с использованием SIFT-признаков является актуальной задачей.

По проблеме реализации детектора точечных признаков SIFT на графических процессорах опубликован ряд статей, например, [74,76-81], в том числе с описанием открытых реализаций на языке Си [76] и с использованием технологии CUDA [80, 81]. Реализация [80] является недокументированной и представлена без временных оценок работы алгоритма. В работе [81] наиболее высокопроизводительные временные оценки приведены для реализации на языке GLSL (такая реализация аппаратно незави-

сима к производителю графического процессора, но обладает рядом ограничений характерных для этого языка: неудобство написания кода и его отладки). Особенностью большинства описанных реализаций является акцент на время выполнения процедуры общего вида, тогда как в прикладных задачах сокращение затрат оказывается возможным за счет априорных сведений из предметной области задачи.

Таким образом, при проектировании собственной реализации ставились цели обеспечения возможностей настройки детектора с учетом особенностей конкретной прикладной задачи и возможностей повторного использования в модульных системах компьютерного зрения реального времени. В целом, можно сказать, что в выбранном подходе к реализации алгоритма производилось ускорение процесса обработки на GPU при наибольшем использовании памяти GPU (реализация этого детектора требует существенных буферов для хранения промежуточных результатов обработки).



Рис. 3.4.1. Схема поиска характерных точек на МПИ (зеленым цветом отмечены буферы, использующие память GPU).

Для поиска характерных точек используется последовательная схема формирования уровней многомасштабного представления изображения (МПИ) и поиска характерных точек на каждом уровне (рис. 3.4.1). Такой подход обеспечивает экономию памяти на промежуточных буферах хранения многомасштабного представления изображения и многомасштабной карты краев (см. рис. 2.7.1-2.7.2). При этом с целью ускорения обработки формируется модификация МПИ, которое разбивается на множество октав [54], так что при переходе к соседней октаве размер изображения набора МПИ уменьшается/увеличивается в 4 раза. Под октавой понимается подмножество изображений модификации МПИ, в котором размер изображений считается одинаковым. Постоянный коэффициент изменения масштабного параметра между смежными изображениями согласуется с количеством октав и количеством уровней в октаве. Например, если *s* – обрабатываемый уровень МПИ октавы о и S – количество уровней в октаве, то масштабный параметр можно задать следующим соотношением  $\sigma(o,s) = \sigma_0 \cdot 2^{o+s/s}$ , a  $\sigma(s) = \sigma(s-1) \cdot 2^{1/s}$  – рекуррентное соотношение между смежными уровнями в октаве [76].





В процессе обработки МПИ свертка изображения осуществляется в соответствии со свойствами сепарабильности гауссовского фильтра g(x, y) = g(x)g(y) и полугруппы  $g(\bullet, \bullet; t_1) * g(\bullet, \bullet; t_2) = g(\bullet, \bullet; t_1 + t_2)$ , т.е. приме-

няется рекуррентная схема на основе двух проходной операции сверки (рис. 3.4.2), при этом используется вспомогательное изображение для хранения промежуточного результата. В процессе выполнения свертки ядро фильтра хранится в константной памяти GPU, а данные изображения считываются в разделяемую памяти GPU.

Вычисление матрицы характерных точек (поиск экстремумов и их субпиксельное уточнение на основе карты краев смежных слоев, при обработке очередного слоя) осуществляется за один вызов исполняемой на GPU функции. Далее переход от матричного представления характерных точек к линейному массиву выполняется подобно схеме показанной на рис. 3.2.4. Для такого перехода вместо бинарной матрицы используется матрица экстремумов, в которой экстремумы максимумов/минимумов маркированы значениями  $\pm 1$ . Наиболее существенным отличием (от схемы, показанной на рис. 3.2.4) является игнорирование знака маркированной на рис. 3.2.4) является игнорирование знака маркирован-ного элемента матрицы экстремумов (рис. 3.4.3).



Рис. 3.4.3. Пример индексирования позиций экстремумов в линейном массиве.

В результате обработки всех октав формируется линейный массив характерных точек. С целью параллельного вычисления доминирующих ориентаций, а также формирования дескрипторов, предлагается в процессе поиска характерных точек заполнять МПИ в заранее зарезервированном буфере памяти. Таким образом, в зависимости от требований к обрабатываемым объемам изображений выполняется либо последовательное вычисление ориентаций и формирование дескрипторов характерных точек каждого слоя/октавы либо параллельное (при небольших по сравнению с объемом памяти GPU объемах входных изображений). В последнем случае предельно допустимый размер обрабатываемых изображений можно оценить из следующего условия

$$M_{SS} + M_{buf} + M_{pts} + \delta \le M_{GPU} \tag{3.4.1},$$

где  $M_{GPU}$  – количество байт доступных в памяти RAM GPU,  $M_{SS}$  – количество байт, требующееся для хранения МПИ,  $M_{buf}$  – количество байт вспомогательного буфера памяти, при обработки изображения МПИ,  $M_{pts}$  – количество байт, требующихся для хранения линейного массива структур найденных характерных точек,  $\delta$  – количество байт, требующихся для выравнивания строк матричных данных в памяти (с целью повышения производительности – уменьшения количества тактов обращения к данным).

Для неравенства (3.4.1) имеют место следующие соотношения

$$M_{SS} = W \cdot H \cdot N_{Lv} \cdot E_{SS} \cdot \sum_{i=0}^{N_O} \frac{1}{4^i}$$
(3.4.2),  
$$M_{buf} = (3 \cdot E_{DoG} + E_{SIFT} + E_{idx}) \cdot W \cdot H$$
(3.4.3)  
$$M_{pts} = N \cdot E_{SIFT}$$
(3.4.4),

где  $\{W, H\}$  – размеры изображения в пикселях (ширина/высота),  $N_o$  – число октав МПИ,  $N_{Lv}$  – число слоев в октаве МПИ, N – число найденных характерных точек на МПИ,  $E_{ss}$  – количество байт, требующихся для хранения единичного элемента уровня МПИ,  $E_{DoG}$  – количество байт, требующихся для хранения структуры данкраев,  $E_{sift}$  – количество байт, требующихся для хранения структуры дан-

ных, описывающей характерную точку (включает координаты точки, размер окрестности, доминирующую ориентацию, тип экстремума, индекс октавы),  $E_{idx}$  – количество байт, требующихся для хранения единичного элемента матрицы индексированных экстремумов (см. рис. 3.4.3).

В разрабатываемой реализации уровни МПИ и уровни многомасштабной карты краев представляются матрицами данных типа float (4 байта на единичный элемент матрицы). Структура данных, описывающая характерную точку, занимает 24 байта и имеет следующую нотацию на языке Си (с использованием CUDA- директив)

```
struct SSIFTPoint {
public:
               //!< субпиксельные координаты
 float x ;
 float y ;
 float scale ; //!< субпиксельный масштаб/размер окрестности
 float angle ; //!< угол ориентации
 int octave_index ; //!< индекс октавы
 int extrema type ; //!< тип экстремума {-1, +1}
public:
 host device SSIFTPoint() : x(0.0f),
                                     y(0.0f),
                                    scale(0.0f),
                                    angle(0.0f),
                                    octave index(0),
                                    extrema type (EXTREMA NO) { };
      } ;
```

Таким образом, при обработке уровня МПИ (см. рис. 3.4.1) минимально допустимый размер буфера определяется, как  $M_{buf} = 36 \cdot W \cdot H$  байт (для хранения элементов матрицы индексированных экстремумов применяется тип данных int, т.е.  $E_{idx} = 4$  байта, также в результате формирования матрицы характерных точек потребность в одном из уровней карты краев отпадает, т.е расчет  $M_{buf} = (2 \cdot 4 + 24 + 4) \cdot W \cdot H)$ .

На рис. 3.4.4 показана общая <u>схема операции вычисления ориента-</u> ции. При выполнении этой операции предусмотрена возможность указания максимально допустимого количества дополнительных ориентаций у характерной точки.

Вследствие того, что для характерной точки может быть назначено несколько ориентаций, то возникает необходимость в их перегруппировке, так, чтобы одному элементу соответствовало одно значение ориентации (формирование нового массива характерных точек). Для этого предварительно формируется интегральная сумма, на основе массива количества ориентаций с использованием CPU, а также определяется размер нового массива. После чего выполняется переиндексация элементов массива и формируется новый массив характерных точек (рис. 3.4.5).



Рис. 3.4.4. Схема операции вычисления ориентации характерных точек.



Рис. 3.4.5. Схема операции вычисления ориентации характерных точек.

<u>Формирование дескрипторов</u> выполняется параллельно, при этом каждая гистограмма дескриптора формируется своей CUDA нитью, после чего производится нормирование дескрипторов (см. рис. 3.4.6).



Рис. 3.4.6. Схема операции вычисления дескрипторов характерных точек.



Рис. 3.4.7. Схема операции сопоставления дескрипторов на основе метрики ближайшего соседа.

Параллельная схема <u>операции сопоставления дескрипторов</u> на основе метрики ближайшего соседа показана на рис. 3.4.7. Как и в задаче сопоставления аэрофотоснимков (п. 3.3) важным фактором, при сопоставлении снимков с БПЛА, является возможность учета физических ограничений:

- Сначала выполняется вычисление матрицы евклидовых расстояний по схеме на основе полного перебора.
- Поиск наилучших пар вдоль строк (такое сопоставление не обладает свойством симметричности метрики).
- 3) Выбор соответствий в пределах заданного порога.

## 3.5. Сравнение и оценки производительности разработанной реализации SIFT-детектора

Временные оценки предложенной реализации SIFT-детектора приведены в таблице 3.5.1. Характеристики приведены для обработки изображения 800х600 пикселей на ПК с центральным процессором Intel Core Duo 6600 и GPU GeForce GTX 260, с параметрами: 4 октавы, 5 слоев, не более двух ориентаций у характерной точки, 128 элементов в дескрипторе (16 гистограмм, по 8 ориентаций в каждой). На рисунке 3.5.1 показано входное изображение и найденные характерные точки (красные – минимумы, синие – максимумы).

Поиск характерных точек (построение МПИ, вычисление матрицы характерных точек и формирование линейного массива)	37.55 мс (974 признака)				
Назначение ориентаций и	5.15 мс (1139 признака)				
перегруппировка					
Формирование дескрипторов	43.89 мс				
Полное время	87.2 мс				
Извлечение точечных особенностей					
Октава	Октава 0	Октава 1	Октава 2	Октава 3	
Построение МПИ	13.25 мс	3.13 мс	1.18 мс	0.68 мс	
Переход от матрицы к линейному масси-	2.36 мс	1.08 мс	0.67 мс	0.475 мс	
ву (индексирование)					
Поиск характерных точек	10.47 мс	4.48 мс	2.71 мс	2.32 мс	
Количество найденных признаков	612	256	89	17	

Таблица 3.5.1. Временные оценки реализации алгоритма Д. Лова

Полученные временные оценки уступают реализации, представленной в работе [79], обработка тестового входного изображения (рис. 3.5.1) потребовала 53.9 мс/1052 характерных точек. Однако представленный подход был рассчитан на задачу обработки снимков с БПЛА, что потребовало более экономичного использования памяти GPU. Например, обработка изображения размерами 4016х2672 пикселов заняла около 2 секунд (~16.000 признаков), тогда как посредством [79] не возможно обработать такое изображение без уменьшения его размеров<sup>1</sup>.



Рис. 3.5.1. Пример обработки изображения 800х600 (изображение из [79] применяется для сравнения с опубликованными реализациями).

Наиболее существенные отличия представленного подхода от реализации из работы [79]:

- 1) не используется текстурная память;
- 2) обработка октав и слоев реализовано последовательно, для экономичного использования памяти GPU;
- переход от матричного представления данных к линейному, а также процедура перегруппировки (при вычислении ориентации), производится с использованием CPU;
- в качестве параметров характерной точки выступают тип экстремума и индекс октавы;

<sup>&</sup>lt;sup>1</sup> Для сравнения реализаций было загружено приложение с сайта [78].

5) предусмотрены настройки SIFT-детектора: индекс начальной октавы (возможно увеличение и уменьшение размеров входного изображения), количество октав и количество слоев в октаве, максимальное количество ориентаций у признака, размерность SIFTдескриптора (количество гистограмм и их размер).



Summary Plot

Рис. 3.5.2. Результаты анализа работы SIFT-детектора, полученные с помощью профилировщика CUDA Visual Profiler. Синие столбцы – затраты на формирование дескрипторов, голубой столбец – затраты на построение МПИ, желтые столбцы – затраты на поиск характерных точек, фиолетовый столбец – затраты на вычисление доминирующей ориентации, красные столбцы – затраты на копирование данных, зеленые столбцы – вспомогательные функции.

На рис. 3.5.2 представлен анализ работы реализации SIFT-детектора с помощью профилировщика CUDA Visual Profiler. На гистограмме показано, что наибольшее время потрачено на формирование SIFT-дескрипторов, построение МПИ и поиск характерных точек, а суммарный вклад на операции копирования данных между CPU $\leftrightarrow$ GPU и GPU $\leftrightarrow$ GPU составляет менее 5%.

#### 3.6. Оценки производительности сопоставления снимков с БПЛА

Оценки производительности реализованного алгоритма сопоставления снимков с БПЛА (рис. 3.6.1-3.6.2) для различных архитектур GPU приведены в таблице 3.6.1. Применение более высокопроизводительных GPU на основе архитектуры Fermi [69] позволяет ускорить поиск признаков и формирование дескрипторов более, чем в 2 раза, а сопоставление дескрипторов на основе полного перебора – в 4 раз. В свою очередь, учет физических ограничений, позволяет достичь двукратного ускорения обработки.

**Таблица 3.6.1** Временные оценки сопоставления дескрипторов для архитектур GPU NVidia GT200/Fermi

GPU	Поиск характерн	Сопоставление де-	
	Снимок 1	Снимок 2	скрипторов
GF GTX 260(GT200)	539 мс/11058	539 мс/9258	5.47c/865
GF GTX 480(Fermi)	222 мс/11063	196 мс/9262	1.38c /866
GF GTX 480(Fermi) и учет	188 мс/9556	170 мс/7965	0.725c/767
ограничений			



Рис. 3.6.1. Пример снимков с БПЛА, обработанных SIFT-детектором (маркерами показаны характерные точки найденные на изображениях, синим цветом - локальные максимумы, красным - локальные минимумы)



Рис. 3.6.2. Пример сопоставления снимков с БПЛА (маркерами показаны соответствия найденные на изображениях, синим цветом - локальные максимумы, красным - локальные минимумы)



Рисунок 3.6.3. Сопоставление двух массивов дескрипторов (по ~10.000 каждый) на основе метрики ближайшего соседа. Высокие столбцы – 128-мерный дескриптор, низкие столбцы – 32-мерный дескриптор. Синие столбцы – полный перебор, красные - добавлено ограничение отсутствия вращения, желтые – добавлено ограничение отсутствия вращения, желтые – добавлено ограничение одной и той же местности, зеленые – добавлено ограничение одной и той же удаленности от объекта наблюдения.

На рис. 3.6.3 показаны результаты <sup>2</sup> сопоставления SIFTдескрипторов для снимков с БПЛА (2008х1336 пикселей), при учете физических ограничений, количество дескрипторов было ~10.000 для каждого из снимков (размер дескрипторов: 128 – для столбцов большей высоты и 32 – для столбцов меньшей высоты. Синие столбцы на рисунке соответст-

<sup>&</sup>lt;sup>2</sup> При сопоставлении использовалось оборудование – СРU Intel Core Duo 6600 с графическим ускорителем GPU GFGTX 260.
вуют сопоставлению без учета ограничений, красные – сопоставление без учета поворота (учитывается только сдвиг изображений), желтые – сопоставление дескрипторов с одинаковым типом экстремума и с учетом сдвига изображений, зеленые – сопоставление дескрипторов с одинаковым индексом октав и с учетом одинаковых типов экстремумов и сдвига изображений.

Таким образом, для сопоставление снимков с БПЛА применение графических процессоров позволяет выполнить задачу сопоставление путем полного перебора, при этом применение очевидных критериев целесообразности сопоставления соответствующих точечных особенностей позволяет ускорить обработку до 8 раз (на GF GTX 260).

#### 3.7. Выводы

Применение GPU для обработки изображений позволяет существенно увеличить скорость обработки на базе доступного оборудования. Возможности технологии программирования CUDA позволяют выполнить эффективную разработку и отладку программного обеспечения, оптимизированного под GPU. Для современных высокопроизводительных архитектур GPU NVidia Fermi/Kepler могут быть использованы эффективные средства управления кэшем при меньшем энергопотреблении [69].

В данной главе рассмотрены новые параллельные реализации алгоритмов сопоставления аэрофотоснимков, оптимизированные для архитектуры NVidia GT200/Fermi. Эти реализации позволяют обрабатывать изображения большего размера по сравнению с известными доступными аналогами.

### Глава 4. СИСТЕМЫ КОМПЬЮТЕРНОГО ВИДЕНИЯ С НЕСКОЛЬ-КИМИ ПОЛЯМИ ЗРЕНИЯ В ПРИКЛАДНЫХ ЗАДАЧАХ

В данной главе описываются системы компьютерного видения с несколькими полями зрения для решения четырех прикладных задач, рассматривавшихся в первой главе. Для каждой системы будут рассмотрены следующие основные вопросы:

- выбор или разработка технологии обработки данных, получаемых с помощью систем с несколькими полями зрения;
- выбор аппаратных средств для разрабатываемых систем компьютерного видения;
- разработка алгоритмов, адаптированных для рассматриваемых систем с несколькими полями зрения и оптимизированных в расчете на выбранное аппаратное обеспечение.
- применение спроектированных программных компонент и разработанных параллельных реализаций;
- примеры работы и оценки результатов работы системы.

### 4.1. Система для автоматического построения фотосхемы местности по данным аэрофотосъемки много-объективной камеры бокового обзора

По результатам анализа существующих фотограмметрических пакетов (п. 1.5), а также с учетом особенностей много объективной камеры (п. 1.1), была принята следующая технология обработки данных аэрофотосъемки:

- Подготовка файла-проекта с описанием исходных данных аэрофотосъемки.
- Формирование единых узкополосных снимков (осуществляется программными средствами производителя камеры).

- 3) Сопоставление узкополосных снимков: поиск связующих точек.
- 4) Фототриангуляция с учетом данных GPS-датчика.
- Геометрическое преобразование снимков и построение фотосхемы местности.

Для построения фотосхем (с минимальной требуемой производительностью 100 кадров в час) требовалось разработать программноаппаратный комплекс, при этом для автоматического сопоставления узкополосных снимков необходимо было необходимо разработать алгоритм, адаптированный для данных, получаемых в результате аэрофотосъемки много-объективной камерой бокового обзора.



Рис. 4.1.1. Структурная схема аппаратной части комплекса обработки данных аэрофотосъемки

В качестве аппаратного обеспечения комплекса была сконфигурирована система (конец 2008 г.), состоящая из двух модулей (см. рис. 4.1.1):

- модуль проведения расчетов;
- рабочее место оператора (подготовка проектов обработки данных и вывод фотосхем на печать).

Модуль проведения расчетов включал в себя Tesla S1070 (4 устройства обработки), двухядерный процессор Intel Xeon Quad-Core X7350 (поддерживает 8 потоков обработки данных), 16Гб ОЗУ, дисковое хранилище 7.5 Тб (RAID5). В качестве операционной системы для функционирования комплекса использовались ОС Windows XP на APM оператора и Windows Server 2003 64 bit (или LINUX RedHat 64 bit) на ЭВМ проведения расчетов.

Для обработки снимков была построена следующая схема параллельной обработки:

- алгоритмы обработки снимков были оптимизированы для GPU архитектуры GT200 (п. 3.2);
- обрабатываемые данные равномерно распределяются по четырем устройствам обработки GPU.

С целью автоматического построения фотосхем был разработан новый алгоритм сопоставления формируемых узкополосных снимков, рассчитанный на GPU архитектуры GT200/Fermi. Алгоритм реализует схему грубо-точного (coarse-to-fine) сопоставления и включает в себя два этапа:

- на первом этапе определяются параметры геометрического преобразования между двумя снимками;
- на втором этапе производится "грубый" поиск соответствий на области перекрытия двух снимков, после чего найденные соответствия локально уточняются.

Высота полета самолета при аэрофотосъемке с использованием рассматриваемой много объективной камеры составляла 7-10 км. Размеры участка местности, покрываемого одним кадром, составляли 30-40 км (рис. 1.1.1). В качестве геометрической модели преобразования между смежными снимками использовалась модель гомографии, для этого вводилось допущение о плоском участке снимаемой местности. При перепаде высот в несколько сотен метров наблюдаемую местность можно аппроксимировать плоскостью, поскольку этот перепад незначителен по сравнению с удаленностью центра фотографирования от наблюдаемой сцены.



Рис. 4.1.2. Алгоритм сопоставления узкополосных снимков (зеленым цветом выделены этапы, реализованные на GPU, либо данные хранятся на GPU).

С целью построения безразрывных фотосхем на основе узкополосных снимков требуется выполнять поиск связующих точек по возможности равномерно по перекрывающимся областям изображений. Поэтому в разработанном алгоритме на первом этапе оценка параметров преобразования между сопоставляемыми снимками выполнялась и оценивалась на основе распределенного вдоль изображения всего набора соответствий (пар соответствующих признаковых точек). Для достижения этой цели был принят критерий оценки распределения набора соответствий на снимке.

Рассмотрим используемый критерий пространственного распределения соответствий на плоскости снимка. Допустим, имеется два множество векторов  $Q = \{\vec{q}\}$  и  $P = \{\vec{p}\}$ , определяющих связующие точки на снимках, т.е. |Q| = |P|, тогда для оценки распределения, использовалось следующее соотношение

$$\begin{cases} \left| x_{\max}^{Q} - x_{\min}^{Q} \right| > \Delta \\ \left| x_{\max}^{P} - x_{\max}^{P} \right| > \Delta \end{cases}$$
(4.1.1),

где  $x_{\{\min,\max\}}^G = \{\min,\max_G\}g_x$  — минимальная/максимальная координата x на некотором множестве векторов  $G = \{\vec{g}\}, \Delta$  — минимально допустимое распределение (удаленность) координат найденных соответствий на снимке.

Рассмотрим первый этап предложенного алгоритма сопоставления узкополосных снимков (на рис. 4.1.2 зеленым цветом отмечены этапы, задействующие GPU, исходные растровые данные также загружены в память GPU):

- Во-первых, аппаратно выполняется формирование уменьшенных копий растровых изображений и сопоставление этих копий с использованием параллельного алгоритма, описанного в п. 3.2.
- 2) На основе критерия (4.1.1) выполняется оценка распределения найденных соответствий вдоль узкополосного снимка. Если это

условие выполняется, то производится коррекция параметров поиска характерных точек на изображении (порог карты откликов детектора Харриса), параметров уровня пирамиды (переход на уровень с большей детализацией) и переход к шагу 1.

- Выполняется фильтрация выбросов и оценка параметров гомографии на основе алгоритма RANSAC [58].
- 4) Для соответствий, оставшихся после фильтрации "выбросов" выполняется шаг 2.

В результате выполнения первого этапа определяется подходящий уровень детализации пирамиды и порог детектора Харриса (или радиус поиска локального максимума), для которых рассчитывается преобразование между снимками, удовлетворяющее условию (4.1.1). В случае, если условие (4.1.1) не выполняется, выбирается то преобразование, для которого найдено распределение, удовлетворяющее условию

$$\begin{cases} \left\| x_{\max}^{Q} - x_{\min}^{Q} \right\| - \Delta \to \min \\ \left\| x_{\max}^{P} - x_{\max}^{P} \right\| - \Delta \to \min \end{cases}$$
(4.1.2).

На втором этапе для обработки использовались исходные растровые снимки, загруженные в память GPU:

 На основе грубого преобразования определяется область перекрытия снимков. Для заданного количества искомых точек выполняется разбиение на блоки, с условием, что один блок содержит одну точку. Допустим {N<sub>x</sub>, N<sub>y</sub>} – количество искомых блоков по ширине и высоте области перекрытия. Введем предположение, что

$$k = \frac{H}{W} \approx \frac{N_y}{N_x} \tag{4.1.3},$$

- -

где k – коэффициент соотношения между количеством блоков по высоте/ширине области перекрытия – {H,W}. Тогда, если N – количество искомых связующих точек и  $N = N_x \cdot N_y$ , то

$$N_v = \sqrt{k \cdot N} \tag{4.1.4}.$$

Далее  $\{N_x, N_y\}$  округляются до целых чисел таким образом, что

$$N \le N_x \cdot N_y \tag{4.1.5}.$$

- 2) Выполняется последовательный обход блоков, при этом в каждом блоке случайным образом выбирается точка и на основе гомографии рассчитывается ее приблизительное положение на смежном снимке. Далее выполняется поиск локального максимума на некоторой окрестности рассчитанного положения, с применением метрики кросс-корреляции (расчет реализован аппаратно, п. 3.2).
- В результате поиска формируется массив соответствий, для которых выполняется фильтрация выбросов на основе модели фундаментальной матрицы (см. п. 3.2).

Для оптимизации обработки *маршрутов*<sup>3</sup> в программном модуле был реализован перенос рассчитанных связующих точек между снимками, имеющими более однократного перекрытия. UML-диаграмма модуля для сопоставления узкополосных снимков показана на рис. 4.1.3. На этой диаграмме отмечены классы спроектированной программной компоненты для сопоставления изображений, на основе которых выполняется расширение компоненты для решения данной прикладной задачи. Новые классы программного модуля:

• CAirCorrelator – класс для поиска связующих точек на наборе узкополосных снимков;

<sup>&</sup>lt;sup>33</sup> Под маршрутом понимаются данные аэрофотосъемки необходимые для построения фотосхем местности

• CFineCorrelator – класс обеспечивает уточняющий поиск, и реализован на основе переопределения методов абстрактного класса I\_Correlator;



Рис. 4.1.3. UML-диаграмма программного модуля для сопоставления узкополосных снимков

Алгоритмы фототриангуляции, трансформирования снимков и построения фотосхем, а также особенности их реализации в рассматриваемом программно-аппаратном комплексе описываются в работах [4,9].

На рис. 4.1.4 показан результат автоматической обработки ~1600 кадров, а также увеличенные фрагменты отдельных типов снимаемой поверхности (прибрежная местность и степь). Время работы алгоритма сопоставления при обработке маршрута занимало порядка 5-15 секунд на снимок, в зависимости от типа снимаемой поверхности, наличия легко обнаруживаемых/идентифицируемых объектов, типа объектов городской застройки и д.р. Время построения фотосхемы заняло около 4 часов. Размер

результирующего изображения составил ~68000x267000 пикселей, объем ~55 Гб.



Рис. 4.1.4. Построение фотосхемы, при наклонной съемке. Время обработки ~4 ч., ~1600 кадров, размеры 68000x267000 пикселей.



Рис. 4.1.5. Построение фотосхемы, при горизонтальной съемке. Время обработки ~2 ч., ~1300 кадров, размеры ~353000х47000 пикселей.

Еще один пример обработки 1300 кадров, при горизонтальной съемке с использованием много объективной камеры, показан на рис. 4.1.5. Размер каждого кадра составляет 52000х1400 пикселей, формат RGB, 8 бит на цветовой канал. Время обработки заняло 2 часа, результирующий растр – 353000х47000 пикселей, объем ~45 Гб.

# 4.2. Система для автоматического построения фотосхем местности по снимкам с двух объективной камеры БПЛА

Распространенным подходом к построению фотосхем по снимкам с некалиброванных камер является восстановление внешних и внутренних параметров камеры с использованием самокалибровки (п. 2.3), при этом, например, в фотограмметрическом пакете Photomod [16] технологическая цепочка адаптирована для камеральной автоматизированной обработки (без возможности интеграции в комплексы реального времени).





Рис. 4.2.1. Внешний вид двух объективной камеры и конструктивные особенности системы с несколькими полями зрения.

Поскольку в данной задаче (п. 1.2) требовалось оперативное построение фотосхемы по набору снимков не более 5 пар кадров, то был разработан и реализован алгоритм, оптимизированный для GPU архитектуры GT200/Fermi. В качестве аппаратного обеспечения системы применялся персональный компьютер: центральный процессор Intel Core i7 930 2.8 ГГц, оперативная память 8Гб RAM, графический процессор NVidia GF GTX 480. Программное обеспечение функционировало на OC Windows XP.



Рис. 4.2.2. Алгоритм построения фотосхемы местности по снимкам с двух объективной камеры БПЛА (зеленым цветом отмечены этапы реализованные на GPU).

Алгоритм построения фотосхемы включает два этапа: этап оценки параметров преобразования между смежными кадрами и этап трансформирования снимков и построения фотосхемы [63].

В качестве геометрического преобразования между смежными кадрами разных (соседних) пар будем использовать – гомографию: допущение о съемке плоской местности (перепад высот на местности незначителен по сравнению с удаленностью центра фотографирования от наблюдаемой сцены). Преобразование между смежными кадрами одной пары также будем рассматривать в виде гомографии, поскольку можно ввести допущение о совпадение центров фотографирования камер в связи с конструктивными особенностями взаимной ориентации объективов (рис. 4.2.1).

Этапы оценки параметров преобразования показан на рисунке 4.2.2 и включает следующие действия:

- Все растровые данные считываются в оперативную память СРU (не более 10 кадров размером 2008х1336 пикселей в цветовом пространстве ARGB, 4 байта на пиксель занимают ~100Мб).
- Поиск характерных точек и вычисление дескрипторов для каждого кадра обрабатываемого набора;
- 3) Сопоставление дескрипторов смежных кадров. Для этого были реализованы следующие последовательности обработки кадров:
  - сопоставление смежных кадров каждой пары;
  - сопоставление кадров смежных пар, обозначенных индексом "0" на рис. 4.2.2;
  - сопоставление кадров смежных пар, обозначенных индексом "1" на рис. 4.2.2.
- 4) Оценка параметров гомографии между смежными кадрами.
- 5) Оценка количества найденных связующих точек между смежными кадрами. При недостаточном количестве точек, обработка вы-

полняется только тех пар кадров, между которыми найдены "сильные" связи.

 Определение преобразование каждого снимка в СК базового снимка (базовый снимок не подвергается коррекции, этот снимок задает СК фотосхемы).

Сопоставление снимков осуществлялось на основе разработанной параллельной реализации алгоритма SIFT. Программная компонента для сопоставления изображений (п. 2.8) использовалась в качестве высокоуровневой оболочки. С целью ускорения обработки выполнялось сопоставление дескрипторов половины кадра (актуальной), а также учитывались физические ограничения, описанные в п. 3.6.

Второй этап, включающий трансформирование снимков, выравнивание яркости и создание бесшовных единых кадров был реализован на GPU.



Рис. 4.2.3. Радиальная взвешивающая функция для выравнивания яркости изображения.

Для выравнивания яркости изображений применялась радиальная взвешивающая функция (рис. 4.2.3). Построение бесшовных единых кадров было реализовано на основе применения линейного взвешивания яркостей изображений на области перекрытия кадров [82]:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \omega \cdot \begin{bmatrix} R_1 \\ G_1 \\ B_1 \end{bmatrix} + (1 - \omega) \cdot \begin{bmatrix} R_2 \\ G_2 \\ B_2 \end{bmatrix}$$
(4..1),

где R, G, B — обрабатываемые каналы изображения,  $\omega = [0...1]$  — коэффициент влияния каналов первого изображения в зависимости от удаленности от границы второго изображения и приближенности к границам первого (см. рис. 4.2.4).



Рис. 4.2.4. Смешивание изображений с линейным взвешиванием яркостей, при построении бесшовных изображений

Результаты обработки трех пар снимков с применением и без применения цветовой коррекции показаны на рис. 4.2.5. Обработка трех пар кадров занимает порядка 10-15 секунд.



а) без применения цветовой коррекции



б) с применением цветовой коррекции

Рис. 4.2.5. Пример построения фотосхемы по снимкам с камеры БПЛА, размеры фотосхемы 4200х2600 пикселей, время формирования фотосхемы – 12 секунд.

### 4.3. Приложение для калибровки систем компьютерного видения с несколькими полями зрения

По теме алгоритмического и программного обеспечения калибровки видеокамер можно опубликован ряд печатных работ [27,34,30,40,42], а также имеются доступные программные средства [83,84], в том числе в библиотеке OpenCV [25]. Во многих из этих работ рассматриваются вопросы автоматизации процесса поиска тестового объекта, методы оценки начального приближения и различных условий применения камер в прикладных задачах. В имеющихся доступных реализациях [83,84] можно отметить отсутствие некоторых функциональных возможностей, существенных для решения рассматриваемых прикладных задач:

- коррекция местоположения меток тест-объекта (точек калибровочного стенда) в случае, если автоматический поиск завершился неудачей;
- 2) выбор способа учета и модели дисторсии.



Рис. 4.3.1. Главное окно приложения для калибровки видеокамер.

При решении прикладных задач требуется провести корректную оценку параметров камеры, при этом модель камеры следует выбирать в соответствии с требованиями прикладной задачи.

Для решения перечисленных проблем было разработано приложение для калибровки видеокамер, а также приложение для определения взаимного ориентирования видеокамер в системах компьютерного видения с несколькими полями зрения.

На рис. 4.3.1 показано главное окно приложения для калибровки видеокамер. Приложение позволяет реализовать следующую технологию калибровки видеокамер:

- 1) Чтение видеоданных с образами тестового объекта.
- 2) Маркирование тестового объекта в двух режимах:
  - в ручном режиме (приложение позволяет добавлять, удалять, редактировать положение, индекс маркера, а также его положение в СК тестового объекта);
  - в автоматическом режиме тестовых объектов в виде набора шахматных клеток (используется реализация OpenCV).
- Задание параметров камеры (возможно задание параметров поумолчанию):
  - указание типа камеры (аналоговая/цифровая);
  - указание данных спецификации датчика изображений: размеры пикселя, фокусное расстояние;
  - выбор способа учета дисторсии (подходы (2.1.3), (2.1.4)) для модели дисторсии Брауна (2.1.5).
- Вычисление параметров камеры при этом формируется список точек с указанием величины ошибки перепроецирования.

Для определения параметров дисторсии независимо от задачи минимизации ошибки перепроецирования был реализован подход на основе метода отвесных линий [34]. Этот метод применим только при учете дисторсионных искажений с помощью восстановления идеальной проекции (2.1.4). Он заключается в фотографировании плоского тестового объекта, на котором можно явно идентифицировать прямые линии и множество точек лежащих на тестовом объекте. Далее для множества наборов точек, принадлежащих различным линиям, формируется система нелинейных уравнений, параметры прямых являются неизвестными

$$\begin{bmatrix} \cos\theta & \sin\theta & -\rho \end{bmatrix}^T \cdot \begin{bmatrix} \vec{D}(\vec{p} - \vec{p}_0) \\ 1 \end{bmatrix} = 0$$
(4.3.1)

где  $\rho \ge 0$  – расстояние от начала координат до прямой;  $\theta$  – угол, задающий вектор нормали;  $\vec{p}$  – координаты точки на плоскости изображения,  $\vec{p}_0$  – координаты главной точки,  $\vec{D}$  – вектор-функция, определяющая модель дисторсии (2.1.5).



Рис. 4.3.2. UML-диаграмма программного модуля калибровки видеокамеры.

При реализации приложения были разработаны два модуля: модуль для отображения и редактирования маркеров и модуль для определения

параметров камеры (он построен на базе программной компоненты для определения параметров камеры, п. 2.4). На рис. 4.3.2 показан класс CDistortionCalibration, унаследованный от класса I\_DiffVecFunction (дифференцируемая вектор-функция). Класс CDistortionCalibration отвечает за формирование системы нелинейных уравнений (4.3.1), при этом используется модель дисторсии Брауна, реализованная в классе CBrownDistortion.

Для решения задачи минимизации ошибки перепроецирования с учетом модели дисторсии, обеспечивающей искажение идеальной проекции, использовалась реализация OpenCV.



Рис. 4.3.3. Пример последовательности изображений для калибровки цифровой камеры БПЛА.

Для разработанных реализаций было выполнено сравнение результатов определения внутренних параметров камеры. На рис. 4.3.3 показаны снимки, для которых выполнялось сравнение. Рассматривались следующие реализации подходов к определению параметров камеры:

- минимизация ошибки перепроецирования для модели камеры, учет дисторсионных искажений (модель Брауна) которой выполняется на основе восстановления идеальной проекции (графики показаны красным цветом);
- минимизация ошибки перепроецирования для модели камеры, учет дисторсионных искажений (модель Брауна) которой выполняется на основе искажения идеальной проекции (OpenCV). Модель камеры задается в OpenCV следующим образом (дисторсия применяется только к относительным значениям проецирующих векторов; графики показаны синим цветом);

$$\vec{p} - \vec{p}_0 = -f \cdot \vec{D} \left( \begin{bmatrix} x' & y' \\ z' & z' \end{bmatrix}^T \right)$$
(4.3.2)

 метод отвесных линий для модели камеры, учет дисторсионных искажений (модель Брауна) которой выполняется на основе восстановления идеальной проекции (графики показаны зеленым цветом).



Рис. 4.3.4. Оценка фокусного расстояния, при калибровке камеры двумя методами.



Рис. 4.3.5. Оценка положения главной точки (векторы показывают тренды для каждого метода, при увеличении количества кадров).



Рис. 4.3.6. Оценка параметров радиальной дисторсии (OpenCV).



Рис. 4.3.7. Оценка параметров тангенциальной дисторсии (OpenCV).



Рис. 4.3.8. Оценка параметров радиальной дисторсии на основе калибровки и метода отвесных линий.

95



Рис. 4.3.9. Оценка параметров тангенциальной дисторсии на основе калибровки и метода отвесных линий.

Результаты сравнения приведены на рис. 4.3.4-4.3.9, при этом следует учитывать, что маркеры были указаны с точностью 1-2 пикселя в полуавтоматическом режиме, а ошибка обратного перепроецирования для обоих методов калибровки варьировалась в диапазоне 0.5-2.5 пикселя (при рассмотрении 12 снимков). На основе графиков можно сделать следующие выводы:

- Для различных способов учета влияния дисторсии, на основе минимизации ошибки перепроецирования найденные величины фокусного расстояния и координаты главной точки отличаются на величину порядка 1 пиксела (рис. 4.3.4, 4.3.5), что соизмеримо с точностью указания маркеров.
- 2) Сравнение параметров дисторсии реализации OpenCV и разработанной реализации (на основе минимизации ошибки перепроецирования для модели камеры, учет дисторсионных искажений которой выполняется на основе восстановления идеальной проек-

ции) нецелесообразно вследствие применения различных математических моделей (рис. 4.3.6- 4.3.9).

- Координаты главной точки снимка и параметров тангенциальной дисторсии, определяемые на основе метода отвесных линий, не применимы для независимой оценки (рис. 4.3.5, 4.3.9).
- Параметры радиальной дисторсии, определяемые на основе метода отвесных линий, применимы для независимой оценки, а расхождения результатов не превышали 5% (рис. 4.3.8).

Mutual calibr	ation	
Left camera(0)		
Camera model	D:/work/dissertation/stereo_system/21_8_12/cam0.txt	Browse
Test-field	D:/work/dissertation/stereo_system/21_8_12/tf_mutual0.xml	Browse
Right camera(1	)	
Camera model	D:/work/dissertation/stereo_system/21_8_12/cam1.txt	Browse
Test-field	D:/work/dissertation/stereo_system/21_8_12/tf_mutual1.xml	Browse
	Op	enCV 💌 Start
Rotation Matrix:		
Basis (m.): [0.22 Angles (deg.): [	0.999732 0.0155484 -0.0171238 -0.0150207 0.999421 0.030531 0.0175886 -0.0302657 0.999387 [5702 -0.0156879 0.0141043] 1.73463 1.0078 0.860786]	
Basis (m.): [0.22 Angles (deg.): [ Reprojection err	0.999732 0.0155484 -0.0171238 -0.0150207 0.999421 0.030531 0.0175886 -0.0302657 0.999387 5702 -0.0156879 0.0141043] 1.73463 1.0078 0.860786] ors (pix.):	

Рис. 4.3.10. Главное окно программы для взаимной калибровки камер стереосистемы.

С целью калибровки систем компьютерного видения с несколькими полями зрения применялась следующая технология:

1) Калибровка внутренних параметров каждой камеры.

- Внешняя калибровка каждой камеры относительно наблюдаемой статической сцены (либо в случае синхронизированных камер возможна внешняя калибровка относительно подвижного тестового объекта).
- Определение матрицы перехода от СК системы камер к СК каждой камеры.

На рис. 4.3.10 показано главное окно приложения для взаимной калибровки камер стереосистемы. Для маркирования тестового объекта применялось рассмотренное выше приложение (рис. 4.3.1). Это приложение обеспечивает возможность записи результатов маркирования в файл.

В заключении следует отметить, что с помощью рассмотренных в параграфе приложений была выполнена калибровка цифровых и аналоговых широкоугольных камер общего назначения, а также определены параметры взаимного ориентирования стереосистем для измерений. Ручная корректировка маркеров, найденных с использованием автоматических методов, требовалась в 90% случаев. При этом для аналоговых широкоугольных камер маркирование в автоматическим режиме успешно выполнялось только для половины обрабатываемых снимков.

# 4.4. Система компьютерного видения для оперативного картографирования

Технологическая схема оперативного картографирования (п. 1.3) состоит из этапов подготовки, включающих калибровку стереосистемы и задание связи БИНС и СК стереосистемы, а также непосредственного этапа обработки – картографирования в режимах on/off-line. Для камеральной обработки данных (off-line режим) использовалась следующая технология:

1) формирование файла-проекта для обработки в режиме off-line;

- нанесение маркеров, определение их координат и выполнение измерений (рис. 4.4.1);
- 3) запись результатов в файл.



Рис. 4.4.1. Главное окно программы для выполнения стерео измерений off-line.

На этапе подготовки для калибровки стереосистем применялось приложение, описанное в п. 4.3. Для стереоизмерений было разработано новое приложение, главное окно которого показано на рис. 4.4.1.

Модуль стерео измерений обеспечивает вычисление координат объектов сцены и применяется в приложениях on-line и off-line обработки

$$\vec{R} = \frac{1}{2} \cdot (\mathbf{K} \cdot (\mathbf{G}^{\mathsf{T}} \cdot \mathbf{G})^{-1} \cdot \mathbf{G}^{\mathsf{T}} + \mathbf{E}) \cdot \vec{B}$$
 (4.4.1), где

где  $\mathbf{K} = [\vec{n}_1 \quad \mathbf{A}^T \vec{n}_2]$  и  $\mathbf{G} = [\vec{n}_1 \quad -\mathbf{A}^T \vec{n}_2]$  – вспомогательные матрицы образованные на основе проецирующих векторов  $\vec{n}_1$  и  $\vec{n}_2$  от двух полей зрения;  $\mathbf{E}$  – единичная матрица;  $\mathbf{A}$  – матрица поворота из СК камеры 1 в СК камеры 2;  $\vec{B}$  – базис фотографирования в СК камеры 1;  $\vec{R}$  – координаты наблюдаемой точки сцены, для которой определены проецирующие вектора  $\vec{n}_1$  и  $\vec{n}_2$  двух соответствующих полей зрения на основе указанных пиксельных координат.

В качестве модели камеры была выбрана модель, на основе восстановления идеальной проекции (2.1.4), обеспечивающая возможность стерео измерений без дополнительной коррекции растрового изображения.



Рис. 4.4.2. Ошибка измерения размеров объектов.



Рис. 4.4.3. Измеренное расстояние до объекта интереса.



Рис. 4.4.4. Ошибка обратного перепроецирования.



Рис. 4.4.5. Размеры объекта на изображении.

Результаты стерео измерений объекта размером ~2.5 м, при движении к нему транспортного средства с установленной бортовой стереосистемой, показаны на рис. 4.4.2-4.4.5. По этим данным об ошибке измерения размеров объекта можно заключить, что точность измерения ~10% от размера объекта достигается на расстояниях до 30-40м. При низком разрешении изображения (720х576 пикселей), на расстояниях более 30м оператору довольно трудно идентифицировать границы объекта, что приводит к существенным ошибкам измерений. При этом указание соответствующих точек на изображениях выполняется достаточно стабильно – средняя ошибка перепроецирования составляет ~1.5 пикс. (рис. 4.4.4).

# 4.5. Система для автоматизации процесса регистрации трещин в хрупких тензочувствительных покрытиях

Совместно с ИМАШ им. А.А. Благонравова РАН была разработана методика автоматизации процесса регистрации трещин в хрупких тензочувствительных покрытиях. Автором было реализовано необходимое программное обеспечение для апробации методики.

Методика включает подготовительный этап, заключающийся в калибровке камеры общего назначения с использованием приложения, описанного в п. 4.3, и этап автоматизации процесса регистрации трещин. Этап автоматизации состоит из следующих шагов обработки:

- Получение изображений трещин при направлении оптической оси камеры под углом 30-60° к наблюдаемой поверхности.
- 2) Формирование 3D-модели исследуемого элемента конструкции.
- 3) Выбор точек модели на снимке.
- Внешняя калибровка камеры относительно наблюдаемого объекта.
- 5) Ортографическое трансформирование снимка на основе 3Dмодели.
- 6) Визуализация результатов обработки.

Рассмотрим применение разработанной методики на примере картин трещин (в оксидном тензочувствительном покрытии), полученных при испытаниях образца авиационной панели на растяжении (рис. 4.5.1).



Рис. 4.5.1. Пример трещин в хрупких тензочувствительных покрытиях, полученных при испытании образца авиационной панели на растяжение.

Каркасная модель выделенного элемента панели была сформирована на основании прямых замеров с использованием графического комплекса 3DStudio Max 8 (рис. 4.5.2). Далее модель экспортировалась в текстовый ASCII формат (\*.ASE) и с помощью программы 3D Object Converter v4.0 конвертировалась в триангуляционную модель (\*.tin), изображение которой приведено на рисунке 4.5.3.



Рис. 4.5.2. Проекция каркасной модели образца авиационной панели в 3D Studio Max 8.



Рис. 4.5.3. Триангуляционная модель проекции авиационной панели.



Рис. 4.5.4. Стрелками отмечены точки, используемые для внешней калибровки камеры относительно образца.

На рис. 4.5.4 стрелками показаны выбранные опорные точки, на основе которых было рассчитаны линейные и угловые параметры камеры относительно образца. Далее на основе имеющейся триангуляционной модели было выполнено ортографическое трансформирование снимка [24] (рис. 4.5.5).



Рис. 4.5.5. Ортографическая трансформация снимка на основе спроектированной модели.



Рис. 4.5.6. Визуализация результатов обработки данных. Оператором найдено 73 трещины, размеры трещин 1.8-6.7°см.

Результаты обработки показан на рис. 4.5.6. Для визуализации и расчетов было разработано специальное приложение (рис. 4.5.3). Рендеринг и текстурирование модели были выполнены на GPU средствами OpenGL [85]. Время обработки в рассмотренном примере составило порядка ~15 минут (на ПК с процессором Intel Core Duo 6600 и графическим процессором NVidia GeForce 260 GTX), ошибки измерения линейных и радиальных размеров объекта менее 5%.

#### 4.6. Выводы

В главе описаны системы компьютерного видения с несколькими полями зрения для решения различных прикладных задач, имеющих важное научно-практическое значение.

В рассмотренных системах используются унифицированные программные компоненты и интенсивно применяются параллельные реализации алгоритмов обработки. Для новых специализированных многообъективных камер предложены новые алгоритмы, оптимизированные для графических процессоров общего назначения.

Для каждой рассмотренной системы показаны практические результаты применимости, а также оценки временных затрат и оценки точности измерений.

#### ЗАКЛЮЧЕНИЕ

Основные результаты работы состоят в следующем:

- Разработана новая архитектура программных компонент для определения параметров камеры и для сопоставления изображений. На основе этой архитектуры реализованы унифицированные повторно используемые программные компоненты, использовавшиеся в программном обеспечении прикладных систем компьютерного видения с несколькими полями зрения.
- Разработаны новые параллельные реализации алгоритмов сопоставления аэрофотоснимков оптимизированные для архитектуры GPU семейств NVidia GT200/Fermi и обеспечивающие обработку изображений большего размера, по сравнению с существующими аналогами.
- 3. Разработаны новые алгоритмы для новых специализированных камер, оптимизированные для архитектуры GPU семейств NVidia GT200/Fermi:
  - алгоритм для сопоставление узкополосных снимков, полученных при аэрофотосъемке с помощью много-объективной камеры бокового обзора;
  - алгоритм для построение обзорных изображений по снимкам, получаемым с двух-объективной камеры БПЛА.
- Спроектированы и реализованы системы компьютерного видения для решения четырех прикладных задач обработки зрительных данных:

- построение фотосхемы местности по данным аэрофотосъемки с много-объективной камеры бокового обзора;
- программно-технический комплекс для мониторинга местности в реальном времени с использованием БПЛА;
- оперативное картографирование местности на основе системы камер и БИНС, установленных на ТС;
- автоматизация процесса регистрации трещин в хрупком тензопокрытии, применяемом для анализа НДС исследуемой конструкции.

### БЛАГОДАРНОСТИ

Автор выражает благодарность научному руководителю А.А.Богуславскому, проф. С.М.Соколову из ИПМ им. М.В.Келдыша РАН, проф. В.Ф.Чекалину и В.А.Мышляеву из ОАО НИИ ТП, а также коллеге и соавтору ряда публикаций – А.В.Крылову из ОАО НИИ ТП за плодотворные совместные обсуждения идей диссертации.
## ЛИТЕРАТУРА

- Платонов А.К., Соколов С.М., Сазонов В.В., Богуславский А.А., Трифонов О.В., Васильев А.И., Куфтин Ф.А., Моксин К.А. Программно-аппаратный комплекс средств навигации мобильных систем. // Вопросы оборонной техники. Серия 9. Специальные системы управления, следящие приводы и их элементы. Вып. 1(242)-2(243), 2010. с. 47-55.
- Махутов Н.А., Васильев И.Е., Богуславский А.А., Васильев А.И. Автоматизация процесса регистрации трещин в хрупких тензочуствительных покрытиях. // Заводская лаборатория. Диагностика материалов, №11, 2010, с. 45-50.
- Соколов С.М., Богуславский А.А., Васильев А.И., Трифонов О.В., Назаров В.Г., Фролов Р.С. Методические и практические аспекты разработки систем компьютерного видения для навигации и измерений // Известия ЮФУ. Технические науки, № 3, март 2012, ISSN 1999-9429, с.76-85.
- Васильев А.И., Крылов А.В., Вахонин А.В. Применение графических процессоров NVIDIA в задачах фотограмметрической обработки результатов ДЗЗ. // Сборник материалов Юбилейной конференции, посвященной 50-летию полета в космос Ю.А.Гагарина, НИИ ТП, 20-21 апреля, 2011, Москва, с. 114-117.
- 5. Васильев А.И., Крылов А.В., Судоргин А.С. Разработка программноматематического обеспечения для поиска совпадающих точек на аэрофотоснимках. Применение технологии NVIDIA CUDA в задаче поиска совпадающих точек. // Сборник материалов Юбилейной конференции, посвященной 50-летию полета в космос Ю.А.Гагарина, НИИ ТП, 20-21 апреля, 2011, Москва, с. 118-121
- Васильев А.И., Крылов А.В., Логутко Д.А., Павленко Г.П. Разработка программно-технического комплекса для проведения аэрофотосъемки и наземной обработки результатов съемки на примере БПЛА

"Типчак". // Сборник материалов Юбилейной конференции, посвященной 50-летию полета в космос Ю.А.Гагарина, НИИ ТП, 20-21 апреля, 2011, Москва, с. 86-90.

- Васильев А.И., Судоргин А.А. Построение обзорных изображений местности по снимкам БПЛА. // Сборник материалов молодежной конференция "Новые материалы и технологии в ракетнокосмической технике", 22-24 июня, 2011, Звездный городок, Т.2 с. 226-229.
- Vasilyev A.I., Boguslavskiy A.A., Sokolov S.M. Parallel SIFT-detector implementation for images matching. // Proc. of the 21st Conference on Computer Graphics and Vision, GraphiCon'2011, September 26-30, 2011, Moscow, pp. 173-176.
- Vasilyev A. I., Krylov A.V., Sudorgin A.A. Application of NVIDIA GPU for the mosaic formation according to the aerial photography data. // Proc. of the 11th International Scientific and Technical Conference "From Imagery to Map: Digital Photogrammetric Technologies", 2011, September 19-22, 2011, Spain, Tossa de Mar, pp. 43-44.
- 10.Makhutov. N.A., Vasil'ev I.E., Boguslavskii A.A., Vasil'ev A.I. Automation of crack recording in brittle tensosensitive coatings. // Inorganic Materials, № 15, vol. 47, 2011 p. 1707-1712.
- 11.Соколов С.М., Богуславский А.А., Трифонов О.В, Васильев А.И. Методические и практические аспекты разработки СТЗ для оперативного мониторинга объектов инфраструктуры и подвижного состава на железнодорожном транспорте. // Труды 3-ей Российская конференция с международным участием "Технические и программные средства систем управления, контроля и измерения" (УКИ-12) 16 19 апреля 2012 года в ИПУ им. В. А. Трапезникова РАН. с. 972-983.

- 12.Hinz A., Dörstel Ch., Heier H. DMC The Digital Sensor Technology of Z/I Imaging. // Photogrammetric Week, D. Fritsch, R.Spiller Ed. Wichmann-Verlag, Heidelberg. pp. 93-104., 2001
- Kröpfl M., Gruber M. UltraCam: Calibration Geometry and Radiometry Analysis, http://www.gtbi.net/export/sites/default/GTBiWeb/productos/descargas/V excelUltracamD-Calibracion-en.pdf (2012)
- 14.Kheiri M. Digital Aerial Cameras. // International Archives of the Photogrammetry, Remote Sensing and Spatial Information Science, Volume XXXVI, Part 6, Tokyo Japan 2006, pp. 229-234.
- 15. Фотограмметрический пакет Erdas Imagine. <u>http://www.erdas.com</u> (2012)
- 16.Фотограмметрический пакет Photomod. <u>http://www.racurs.ru</u> (2012)
- 17. Формат хранения растровых данных TIFF. <u>http://www.libtiff.org/</u> (2012)
- 18.Формат хранения растровых данных JPEG. <u>http://www.jpeg.org/</u> (2012)
- 19.БПЛА "Типчак". <u>http://ru.wikipedia.org/wiki/Типчак (БПЛА)</u> (2012)
- 20.Картографическая система координат WGS-84. http://earthinfo.nga.mil/GandG/publications/tr8350.2/tr8350\_2.html (2012)
- 21. Пригоровский Н.И., Панских В.К. Метод хрупких тензочувствительных покрытий. – М.: Наука, 1978. – 184 с.
- 22.НПО "Регион". <u>http://www.nporegion.ru/</u> (2012)
- 23.Лобанов А.Н. Фотограмметрия: Учебник для вузов. 2-е изд., перераб.
  и доп. М.: "Недра", 1984. 552 с.
- 24.Назаров А.С. Фотограмметрия: учеб. пособие для студентов вузов. — Мн.: ТетраСистемс, 2006. — 368 с.: ил.
- 25.Open Computer Vision Library. <u>http://opencv.org/</u> (2012)
- 26.Bradski G., Kaehler A., Learning OpenCV. // OReilly. ISBN: 978-0-596-51613-0, 2008

- 27.Zhang Z. A flexible new technique for camera calibration. // IEEE Trans. on PAMI, 2000, Vol. 22(11), pp. 1330-1334
- 28.Intel Integrated Performance Primitives. <u>http://software.intel.com/en-us/articles/intel-ipp</u> (2012)
- 29.Hartley R., Zisserman A. Multiple View Geometry in computer vision. // Cambridge University Press. ISBN 0-521-54051-8, 2003
- 30.Wöhler C. 3D Computer Vision: Efficient Methods and Applications. // X.media.publishing, 2009
- 31.Forsyth D.A., Ponce J. Computer Vision, A Modern Approach. // Prentice Hall. ISBN 0-12-379777-2, 2003
- 32.Шредер Г., Трайбер Х. Техническая оптика. М.: Техносфера, 2006г. 424.с. ISBN 5-94836-075-Х
- 33.Русинов М.М. Техническая оптика: учеб. пособие для вузов. Л.: Машиностроение. Ленинградское отд., 1979 г. — 488 с.
- 34.Brown D. C. Close-Range Camera Calibration. // Photogrammetric Engineering vol. XXXVII, no. 8, 1971, pp. 855-866
- 35.Бард Й. Нелинейное оценивание параметров. / Пер. с англ. В.С. Дуженко и Е.С.Фоминой; Под ред. и. с предисл. В.Г. Горского — М.: Статистика, 1979 — 349с.
- 36.Поляк В.Т. Введение в оптимизацию. М.: Наука, 1983. 384 с.
- 37.Васильев Ф.П. Методы оптимизации. М.: Факториал Пресс, 2002.
   824 с.
- 38. Triggs B., McLauchlan P. F., Hartley R. I., Fitzgibbon A. W. Bundle adjustment - a modern synthesis. // In:ICCV '99: Proceedings of the International Workshop on Vision Algorithms, Springer-Verlag, London, UK, pp. 298–372., 2000
- 39. Distortion tester.

http://www.zeiss.com/C1257569004B5D2C/EmbedTitelIntern/Distortion Tester/\$File/Carl\_Zeiss\_Distortion\_Tester.pdf (2012)

- 40.Tsai R. Y. A Versatile Camera Calibration Technique for High-Accuracy
  3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and
  Lenses. // IEEE J. of Robotics and Automation RA 3(4), pp. 323–344,
  1987
- 41.Clarke T.A., Fryer J.G. The Development of Camera Calibration Methods and Models. // Photogrammetric Record, 16(91): 51-66, April 1998
- 42.Михайлов А.П. и др. Применение цифровых неметрических камер и лазерных сканеров для решения задач фотограмметрии. <u>http://www.racurs.ru/www\_download/articles/Camaras\_digitales\_rus.pdf</u> (2012)
- 43.Abidi IM. A., Chandra T. A New Efficient and Direct Solution for Pose Estimation Using Quadrangular Targets: Algorithm and Evaluation. // IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol 17, No. 5, pp. 534-538, 1995.
- 44.Long Q., Zhong-Dan L. Linear N-Point Camera Pose Determination. // IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol.21, No.7, pp. 774-780, 1999.
- 45.Longuet-Higgins H.C. A Computer Algorithm for Reconstructing a Scene From Two Projections. // Nature, vol. 293, pp. 133–135, Sept 1981.
- 46.Nist'er D. An Efficient Solution to the Five-Point Relative Pose Problem.// CVPR, 195-202, 2003.
- 47.Schaffalitzky F., Zisserman A., Hartley R. I., Torr P.H.S. A Six Point Solution for Structure and Motion. // ECCV (1) 2000: 632-648
- 48.Szeliski R. Computer Vision: Algorithms and Applications. Springer, New York, 2010.
- 49.Конушин А.С. Автореферат диссертации. Алгоритмы построения трехмерных компьютерных моделей реальных объектов для систем виртуальной реальности, 2005,

http://www.keldysh.ru/council/1/konushin.pdf (2012)

- 50.Шапиро Л. Компьютерное зрение : Учебное пособие для вузов : Пер. с англ. / Л. Шапиро, Дж. Стокман ; пер. : А.А. Богуславский ; ред. пер. : С. М. Соколов. — М.: БИНОМ. Лаборатория знаний, 2006. — 752 с.
- 51.Harris C., Stephens M. A combined corner and edge detector. // Fourth Alvey Vision Conference, 1988
- 52.Corner Detection Algorithms. http://kiwi.cs.dal.ca/~dparks/CornerDetection/algorithms.htm (2012)
- 53.Hu M.K. Visual Pattern Recognition by Moment Invariants. // IRE Trans. Info. Theory, vol. IT-8, pp.179–187, 1962
- 54.Lowe D.G. Distinctive Image Features from Scale-Invariant Keypoints. // International Journal of Computer Vision, 60, 2, pp. 91-110, 2004.
- 55.Bay H., Ess A., Tuytelaars T., Gool L.V. SURF: Speeded Up Robust Features. // Computer Vision and Image Understanding (CVIU), Vol. 110, No. 3, pp. 346--359, 2008
- 56.Ke Y., Sukthankar R. PCA-SIFT: A More Distinctive Representation for Local Image Descriptors. // Computer Vision and Pattern Recognition, 2004.
- 57.Matas J., Chum O., Urban M., Pajdla T. Robust wide baseline stereo from maximally stable extremal regions. // Proc. of British Machine Vision Conference, pages 384-396, 2002.
- 58.Fischler M.A., Bolles R.C. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. // Comm. of the ACM 24 (6): 381–395. June 1981.
- 59.Choi, Kim, Yu. Performance Evaluation of RANSAC Family. // BMVC, 2009, doi:10.5244/C.23.81
- 60.Luong Q.T., Faugeras O.D. The Fundamental Matrix: Theory, Algorithms, and Stability Analysis. // International Journal of Computer Vision 17 (1): 43–75. doi:10.1007/BF00127818, 1996
- 61. Хартсхорн Р. Основы проективной геометрии. М.: Мир, 1970.

- 62.Grün A. Adaptive least squares correlation: a powerful image matching technique. // In: South African Journal of Photogrammetry, RS and Cartography, Vol. 14 (3), pp. 175-187, 1985
- 63.Zitova, B., Flusser, J. Image registration methods: a survey. In: Image and Vision Computing, Vol. 21, Issue 11, pp. 977-1000, 2003
- 64. Гонсалес Р., Вудс Р. Цифровая обработка изображений. М.: Техносфера, 2005
- 65.Rosenfeld A., Vanderbrug G.J. Coarse–fine template matching. // IEEE Transactions on Systems, Man and Cybernetics 7 (1977) 104–107.
- 66.Rodrigues R., Pellegrino S. An Experimental Evaluation of Algorithms for Aerial Image Matching. // 17th International Conference on Systems, Signals and Image Processing (IWSSIP). – 2010. – P.416 – 419.
- 67.Lindeberg T. Detecting salient blob-like image structures and their scales with a scale-space primal sketch: a method for focus-of-attention. // International Journal of Computer Vision, 11(3):283-318. 1993
- 68.Lindeberg T. Scale-space theory: A basic tool for analysing structures at different scales. // Journal of Applied Statistics, 21(2):224-270. 1994
- 69.NVIDIA, CUDA technology, http://www.nvidia.com/object/cuda\_home\_new.html (2012)
- 70.Боресков А.В., Харламов А.А. Основы работы с технологией CUDA.– Москва, ДМК Пресс, 2010
- 71.Sanders J., Kandrot E. CUDA by Example: An Introduction to General-Purpose GPU Programming. 2010. Addison-Wesley Professional. 312 p. ISBN: 978-0131387683, 2010
- 72.Harris detector source code (matlab), http://www.cs.illinois.edu/~slazebni/spring11/harris.m (2012)
- 73.Sinha S., Frahm J., Pollefeys M., Genc Y. Feature tracking and matching in video using programmable graphics hardware. // Machine Vision and Applications, 2007

- 74.Hosseini F., Fijany A., Fontaine J.-G. Highly Parallel Implementation of Harris Corner Detector on CSX SIMD Architecture. // the 16th International European Conference on Parallel and Distributed Computing (Euro-Par), August 31-September 3, 2010, Ischia - Naples, Italy
- 75.Claus C., Huitl R., Rausch J., Stechele W. Optimizing the SUSAN corner detection algorithm for a high speed FPGA implementation. // Field Programmable Logic and Applications, 2009.
- 76.Vedaldi A. Sift++ source code and documentation. http://www.vlfeat.org/ (2012)
- 77.Warn S., Emeneker W., Cothren J., Apon A. Accelerating SIFT on Parallel Architectures. // Cluster Computing and Workshops, 2009
- 78.Heymann S., Muller K., Smolic A., Frohlich B., Wiegand T. SIFT Implementation and Optimization for General-Purpose GPU. // in WSCG 2007
- 79.Wu C. SIFTGPU: A GPU implementation of scale invariant feature transform (SIFT). <u>http://www.cs.unc.edu/~ccwu/siftgpu/#lowesift</u> (2011)
- 80.Bjorkman M. A CUDA implementation of SIFT http://www.csc.kth.se/~celle/ (2011)
- 81.Kayombya G.-R. Implementation and Optimization of SIFT on an OpenCL GPU,

http://beowulf.csail.mit.edu/18.337/projects/reports/Kayombya\_report.pdf (2010)

- 82.Burt P.J., Adelson A.H. A multiresolution spline with application to image mosaics. // ACM Transactions on Graphics, 1983
- 83.V.Vezhnevets, A.Velizhev, N. Chetverikov, A. Yakubenko. GML C++ Camera Calibration Toolbox.

http://graphics.cs.msu.ru/ru/science/research/calibration/cpp (2012)

- 84.Multi-Camera Self-Calibration, <u>http://cmp.felk.cvut.cz/~svoboda/SelfCal/</u> (2012)
- 85.Open Graphic Library, <u>http://www.opengl.org/</u> (2012)