

**Московский Физико-Технический Институт  
(Государственный Университет)**

**На правах рукописи**

**Слынько Юрий Вячеславович**

**РАЗРАБОТКА И ИССЛЕДОВАНИЕ АЛГОРИТМОВ ОПРЕДЕЛЕНИЯ  
ГЕОМЕТРИЧЕСКИХ ПРЕОБРАЗОВАНИЙ КАДРОВ  
ВИДЕОПОСЛЕДОВАТЕЛЬНОСТИ И ИХ ПРИМЕНЕНИЕ К ЗАДАЧАМ  
СТАБИЛИЗАЦИИ, СОПРОВОЖДЕНИЯ И СЕЛЕКЦИИ ДВИЖУЩИХСЯ  
ОБЪЕКТОВ**

**Специальность 05.13.17 – «Теоретические основы информатики»**

**Диссертация на соискание ученой степени кандидата  
физико-математических наук**

**Научный руководитель:  
к.т.н., доцент В. Н. Лагуткин**

**Москва - 2008**

## Оглавление

2.4.2	Детектирование особых случаев шума.....	73
2.4.3	Детектирование чересстрочной развертки.....	74
2.4.4	Отсеивание смазанных кадров .....	75
2.4.5	Анализ и результаты.....	76
2.5	Построение панорамного кадра.....	78
2.6	Выводы.....	81
3	Сопровождение движущихся объектов .....	82
3.1	Введение .....	82
3.2	Базовый алгоритм сопровождения.....	83
3.3	Алгоритмы оконтуривания .....	85
3.3.1	Оконтуривание по цвету .....	86
3.3.2	Оконтуривание по контрастности.....	86
3.3.3	Оконтуривание методом выделения границ.....	87
3.3.4	Временная фильтрация.....	88
3.3.5	Постобработка контура .....	89
3.3.6	Алгоритмы обнаружения и сопровождения на основе контуров объектов.....	90
3.4	Алгоритм сопровождения методом максимального правдоподобия.....	91
3.4.1	Оценка функции максимального правдоподобия .....	92
3.4.2	Поиск максимума правдоподобия .....	94
3.4.3	Фильтрация.....	96
3.4.4	Слабообусловленная задача.....	97
3.4.5	Возможные модификации.....	98
3.5	Алгоритм оценки шума .....	98
3.6	Алгоритм управления приводом камеры .....	99
3.7	Анализ и результаты.....	100
3.8	Выводы.....	102
4	Селекция движущихся целей.....	103

4.1	Введение .....	103
4.2	Алгоритм авторегрессии .....	104
4.3	Сигнальная компенсация .....	105
4.4	Анализ и результаты.....	107
4.5	Границы применимости .....	109
4.6	Выводы.....	113
5	Экспериментальная установка и программная реализация.....	115
5.1	Введение .....	115
5.2	Описание экспериментальной установки.....	115
5.3	Программная реализация .....	119
5.4	Функциональные возможности .....	123
5.5	Выводы.....	124
	Выводы .....	126
	Литература .....	128

## Введение

**Актуальность темы.** Задачи автоматической обработки видеоизображений принадлежат к числу наиболее востребованных задач современной математики и кибернетики. К ним относятся задачи сжатия и улучшения качества изображений, обнаружения и сопровождения объектов, построения трехмерных моделей наблюдаемых картин, распознавания образов и принятия решений на основе полученной информации.

Такие задачи решаются в системах видеонаблюдения, системах мониторинга и обнаружения опасных ситуаций, робототехнике - начиная от бытовых приложений и заканчивая космической техникой. Методы компьютерного зрения используются в протоколах передачи и стандартах хранения видеинформации. В последнее время алгоритмы компьютерного зрения активно применяются в медицине.

Видеопоток характеризуется большим количеством информации при сравнительно низкой стоимости оборудования. Этим обусловлена возрастающая популярность видеокамер как универсальных измерительных инструментов для эффективного решения многих практических задач.

При современном уровне развития компьютерной и видеотехники становится возможным создание систем, обрабатывающих видеопоследовательности в реальном времени на таком уровне качества, при котором становится целесообразно уделять им большое внимание. В частности, пропускная способность современных каналов сети Интернет позволяет передавать по ним видео в реальном времени. Однако для такой передачи оно должно быть сжато весьма сложно устроенным алгоритмом сжатия. В задачах видеонаблюдения и контроля программные средства позволяют в автоматическом режиме оценивать ситуацию и выдавать предупреждения оператору обо всех подозрительных ситуациях. В задачах робототехники возможно создание роботов и автоматов с очень сложным поведением, вплоть до стыковки космических аппаратов.

Но, несмотря на такой прогресс, количество нерешенных задач намного превышает количество решенных. Например, подавляющее большинство интеллектуальных систем видеонаблюдения использует стационарные камеры, алгоритмы сжатия изображений дают значительное снижение качества видео, а системы автоматического управления на автомобилях или самолетах до сих пор лишь помогают оператору управлять, но не берут управление на себя. Ну, а уровень обработки изображений человеческим глазом до сих пор остается не досягаем для современных систем компьютерного зрения.

Зачастую съемка производится подвижным носителем (например, камера установлена на автомобиле, вертолете или другой нестационарной платформе). В таких ситуациях необходима стабилизация изображения. Она целесообразна как для улучшения восприятия изображения оператором, так и в качестве предварительного шага для большей части алгоритмов компьютерного зрения, рассчитанных на работу со стационарным кадром.

Другой важной задачей обработки изображений, тесно связанной с задачей стабилизации, является задача сопровождения выделенных объектов. Эта задача актуальна для систем видеонаблюдения и мониторинга, применяемых для контроля периметра, в милицейских системах наблюдения и т.д.

Еще одна область применения алгоритмов компьютерного зрения - селекция движущихся целей. Она актуальна в космической и авиационной отраслях для наблюдения и контроля подстилающей поверхности.

**Целью диссертационной работы** является создание и исследование алгоритмов определения геометрических трансформаций кадров видеопоследовательности и их применение к задачам стабилизации и построения мозаики, а также алгоритмов сопровождения объектов и селекции движущихся целей.

**Задачами диссертационного исследования являются:**

1. Разработка, реализация и исследование методов реального времени совместного сопровождения и оконтуривания объектов в условиях существенной априорной неопределенности.
2. Создание, реализация и анализ алгоритма реального времени оценки геометрических искажений кадров видеопоследовательности в применении к задачам стабилизации и построения панорамных изображений.
3. Создание, реализация и исследование алгоритма реального времени определения качества видеопоследовательности.
4. Создание и испытание экспериментального прототипа комплекса видеонаблюдения и программного обеспечения для него.
5. Разработка и реализация алгоритма селекции движущихся целей при авиационном или космическом наблюдении, а также анализ точности и границ его применимости.

**Методы исследования.** Для достижения поставленной цели в диссертационной работе используются методы теории вероятностей, теории случайных процессов, теории принятия решений, вычислительной математики, а также имитационного моделирования и физического эксперимента.

**Основные положения, выносимые на защиту:**

1. Решающее правило для задачи совместного сопровождения и оконтуривания объектов в условиях существенной априорной неопределенности на основе критерия максимального правдоподобия.
2. Алгоритм оценки геометрических искажений кадров видеопоследовательности на основе объединения трех основных существующих подходов с оценкой качества (доверительной области) решения на каждом шаге алгоритма и его применение к решению задач стабилизации и построения панорамных кадров.
3. Критерий качества видеопоследовательности на основе разработанных метрик с классификацией методами нейронных сетей.

4. Алгоритм селекции движущихся целей при авиационном или космическом наблюдении, а также характеристики его применимости и оценка качества.

**Научная новизна:**

1. Введен аналитический вид функции правдоподобия для задачи одновременного сопровождения и оконтуривания.
2. Разработана численная схема поиска максимума функции правдоподобия в многомерном пространстве.
3. Выведена зависимость качества (доверительной области) принятия решения при оценке сдвига изображения корреляционным методом.
4. Выведены аналитические зависимости точности оценок параметров геометрических преобразований при различных преобразованиях изображений.
5. Введены новые метрики для изображений, позволяющие осуществлять оценку качества видеопоследовательности.
6. Разработан метод селекции движущихся целей при авиационном и космическом наблюдении и найдены границы его применимости.

**Практическая ценность и реализация результатов.** Результаты работы внедрены и используются в ОАО «МАК «Вымпел» при разработке моделирующего комплекса в рамках ОКР «Фронтон», НИР «Палаццо» и при выполнении работ по проекту «ИРИНА», о чем свидетельствует соответствующие акты внедрения.

Результаты работы реализованы в «Программно - алгоритмическом комплексе стабилизации и улучшения качества видеоизображений, получаемых информационными средствами космического базирования», зарегистрированном в реестре программ для ЭВМ №2007614534 от 27 октября 2007г.

Кроме того, результаты диссертационной работы использовались в учебном

процесса на Кафедре информационных систем МФТИ (ГУ) при ОАО «МАК «Вымпел».

**Апробация результатов работы.** Основные результаты диссертации докладывались и обсуждались на:

- Международных конференциях «Цифровая обработка сигналов и её применение» (2006г., 2008г., Москва).
- Научно-техническом семинаре молодых ученых «Проблемы обнаружения и сопровождения космических объектов в околоземном космическом пространстве» (2007г., Москва).
- Военно-научной конференции (2007г., Тверь).
- Выставках «Высокие технологии - XXI» (2007 – 2008гг., Москва).
- Выставке «МАКС» (2007г., Москва).
- Научных конференциях МФТИ (2002 - 2007гг., Долгопрудный).

**Публикации.** По теме диссертации опубликовано 16 научных работ. Три из них [1 - 3] в рецензируемых журналах, утвержденных в перечне ВАК.

По теме диссертационной работы опубликованы следующие работы:

1. *Ю.В. Слынько, В.Н. Лагуткин, А.П. Лукьянов.* Разработка и исследование робастных алгоритмов реального времени оценивания параметров геометрических преобразований кадров видеопоследовательностей. // Радиотехника и электроника, 2007, том 52, №3, С. 351-357.
2. *А.Е.Колесса, В.Н.Лагуткин, А.П.Лукьянов, Ю.В. Слынько.* Комплекс алгоритмов оценки и компенсации геометрических линейных искажений входной информации, полученной при неизвестных динамических изменениях системы координат наблюдателя. // Вопросы радиоэлектроники, сер РТЛ, 2004, вып. 1. С. 58 – 72.
3. *Ю.В. Слынько.* Селекция движущихся целей при авиационном наблюдении с построением трехмерной карты облачного слоя. // Вопросы радиоэлектроники, сер. РТЛ, 2006, вып. 1. С. 107 - 122.

4. *Yuriy Slyntko*. Maximum likelihood approach to the problem of simultaneous contouring and tracking. // Proceedings of ELMAR-2008, in press.
5. *Ю.В. Слынько, А.П. Лукьянов, В.Н. Лагуткин*. Гибридный подход к решению задачи оценивания геометрических искажений кадров видеопоследовательностей. // Доклады конф. Цифровая обработка сигналов и ее применение, Вып. VIII – 2, С. 437-440.
6. *Ю.В. Слынько* Определение смены сцены и шумоподобности кадров видеопоследовательности. // Доклады конф. Цифровая обработка сигналов и ее применение, Вып. X – 2, С. 422-425.
7. *А.Е. Колесса, Ю.В. Слынько, В.Н. Лагуткин, А.П. Лукьянов* Программно-алгоритмический комплекс стабилизации и улучшения качества видеоизображений, получаемых информационными средствами космического базирования: свидетельство об официальной регистрации программ для ЭВМ №2007614534 от 26 октября 2007г.
8. Технический отчет по проекту “ИРИНА”. / 2007 г., ОАО «МАК «Вымпел».
9. *Ю.В. Слынько, Е.Н. Подкорытов*. Алгоритм компенсации нестационарного мешающего фона с целью обнаружения движущихся точечных объектов. // Труды XLV научной конференции МФТИ “Современные проблемы фундаментальных и прикладных наук”, 2002, Ч. 1. С. 49.
10. *Ю.В. Слынько*. Синтез алгоритма фильтрации нестационарного фона при авиационном наблюдении. // Труды XLVI научной конференции МФТИ “Современные проблемы фундаментальных и прикладных наук”, 2003, Ч.1. С. 67.
11. *Ю.В. Слынько*. Алгоритм оценки и компенсации геометрических линейных искажений входных видео изображений. // Труды XLVII

- научной конференции МФТИ “Современные проблемы фундаментальных и прикладных наук”, 2004, Ч.1, С. 61-63.
- 12.Ю.В. Слынько. Обзор принципов первичной обработки изображений человеческим глазом применительно к вопросам компьютерного зрения. // Труды XLVIII научной конференции МФТИ “Современные проблемы фундаментальных и прикладных наук”, 2005, Ч. 1, С. 87-88.
- 13.Ю.В. Слынько. Методы построения алгоритма классификации и оценки качества видеопоследовательности. // Труды XLIX научной конференции МФТИ “Современные проблемы фундаментальных и прикладных наук”, 2006, Ч. 1, С. 106-107.
- 14.Ю.В. Слынько. Построение 3D картины по движению и селекция движущихся целей при авиационном наблюдении. // Труды 32 Военно-научной конференции, 2007, ч.2, кн. 2. И nv. 51470. ФГУ «2 ЦНИИ МО РФ».
- 15.Ю.В. Слынько Исследование влияния предобработки видеопоследовательности на эффективность алгоритмов оценивания. // Труды XLX научной конференции МФТИ “Современные проблемы фундаментальных и прикладных наук”, 2007, Ч. 1., С. 130 - 132.
- 16.Ю.В. Слынько, А.Е. Колесса Селекция движущихся целей на фоне движущейся подстилающей поверхности: ТД 27404, 2008, ОАО «МАК «Вымпел», с. 50-54.

**Структура и объем диссертационной работы.** Диссертация состоит из введения, пяти глав, заключения, списка литературы, включающего 55 наименований. Работа изложена на 133 страницах и содержит 47 рисунков и 10 таблиц.

Алгоритм определения геометрических преобразований кадров изложен в главе 1. Его применение в задаче стабилизации и построения панорамных кадров описано в главе 2. Там же изложен алгоритм определения качества видеопоследовательности как необходимая часть алгоритма стабилизации.

В главе 3 приведены алгоритмы сопровождения и оконтуривания выделенных объектов.

Предложенный алгоритм определения геометрических искажений использован в решении задачи селекции движущихся целей при авиационном или космическом наблюдении. В главе 4 описаны специфика данного алгоритма, а также границы его применимости.

Состав и структура экспериментального прототипа комплекса реального времени обработки видеоизображений приведены в главе 5. Там же описаны детали программной реализации.

# 1 Оценка геометрических преобразований

## 1.1 Введение

Одной из базовых проблем в обработке видеопоследовательностей является задача определения параметров геометрических преобразований кадров. К решению этой задачи сводятся такие задачи, как стабилизация изображений, сопровождение движущихся объектов и оценка параметров их движения, создание мозаики.

Более того, многие алгоритмы обработки изображений, такие как выделение переднего плана, распознавание образов требуют предварительной стабилизации изображений.

В данной работе рассматриваются алгоритмы определения геометрических преобразований кадров, удовлетворяющие следующим условиям:

- алгоритмы должны работать устойчиво с видеопоследовательностями различного типа, включая снятые в сложных условиях и имеющие низкое качество, т.е. обладать рабочестью.
- данные алгоритмы должны обеспечивать обработку видеопотоков в реальном времени на современных неспециализированных вычислительных средствах.

Алгоритм определения геометрических преобразований описан в разделе 1.2. В разделе 1.3 представлены некоторые способы учета различных особых явлений на изображении. Раздел 1.4 посвящен теоретическому исследованию качества представленных алгоритмов и характеристик изображений, влияющих на это качество. И, наконец, завершают эту главу выводы – раздел 1.5.

## 1.2 Оценка геометрических преобразований

Существуют различные методы оценки параметров деформации кадра: на основе оптического потока внутри кадра, на основе выделения характерных точек изображения, на основе непосредственного использования яркости точек входного изображения и с использованием различных интегральных преобразований.

Методы на основе оптического потока сводятся к вычислению сдвига каждого пикселя на изображении с последующим вычислением полного преобразования кадра. Два наиболее известных алгоритма определения оптического потока представлены в [2, 3]. Их развитие можно найти в работах [1, 4]. Среди достоинств данного метода следует отметить возможность применения в случаях различных трансформаций изображений и фильтрации мешающих движущихся объектов. Основным недостатком является малая максимально определяемая величина трансформации и «проблема апертуры», на которую впервые было указано еще в 1911 году [5].

Другой метод заключается в поиске на изображении особых точек и соотнесении этих точек друг с другом на разных изображениях. Одна из первых работ на эту тему – [8]. Одним из самых известных алгоритмов этого класса является SIFT [10]. Среди достоинств метода можно отметить возможность его использования для сложных трехмерных сцен [6, 7]. Основным недостатком является низкое качество работы на размытых сценах с малым количеством характерных особенностей и большая вычислительная сложность.

Прямые методы определения сдвига основываются, как правило, на вычислении функции корреляции или невязки. Один из первых вариантов таких алгоритмов приведен в [9]. Другие примеры могут быть найдены в [11].

Существует много способов стабилизации изображений с использованием различных интегральных преобразований. Это может быть преобразование Фурье, разложение по вейвлетам [19] и др. Первым недостатком таких алгоритмов является малый максимальный определяемый

сдвиг. Как правило, они совмещают в себе, помимо стабилизации, еще и алгоритмы повышения разрешения [17], что ведет к высоким требованиям на отношение сигнал шум. Помимо этого, такие алгоритмы, как правило, работают только со сдвигом, и не могут определять более сложные преобразования.

Большинство вышеперечисленных методов не рассчитаны на работу в реальном времени для крупных кадров и/или на неспециализированных вычислительных средствах. Они работают устойчиво, если входные изображения удовлетворяют выбранной математической модели и часто становятся неустойчивыми при изменении этой модели или несоответствии ей входных изображений.

Как было сказано выше, в представленных алгоритмах сделан упор на скорость работы и устойчивость (робастность) по отношению к изменениям условий применения.

В качестве ключевых особенностей предложенного алгоритма можно выделить две:

1. Алгоритм основан на объединении трех основных подходов к решению данной задачи, что позволяет компенсировать их недостатки и подчеркнуть достоинства.
2. В алгоритмах определения сдвига используется детальный анализ функции корреляции, что позволяет оценивать точность и надежность принятия (доверительную область) решения и ограничить вычисления тогда, когда эти показатели достигнут заданных значений.

### 1.2.1 Общая структура алгоритма

Рассматриваемая задача формулируется следующим образом. Для каждого кадра видеопоследовательности необходимо оценить его геометрическую трансформацию относительно предыдущего, полагая при этом, что кадры являются изображением одного и того же стационарного фона. Под геометрической трансформацией подразумевается произвольное непрерывно

дифференцируемое взаимно однозначное преобразование координат. Наиболее употребительными являются следующие трансформации:

аффинное преобразование –

$$\begin{pmatrix} x(t+1) \\ y(t+1) \end{pmatrix} = \begin{pmatrix} a_0 & a_1 \\ a_2 & a_3 \end{pmatrix} \cdot \begin{pmatrix} x(t) \\ y(t) \end{pmatrix} + \begin{pmatrix} a_4 \\ a_5 \end{pmatrix}, \quad (1.1)$$

проективное –

$$\begin{pmatrix} x(t+1) \\ y(t+1) \end{pmatrix} = \frac{1}{a_6x(t) + a_7y(t)} \begin{pmatrix} a_0 & a_1 \\ a_2 & a_3 \end{pmatrix} \cdot \begin{pmatrix} x(t) \\ y(t) \end{pmatrix} + \begin{pmatrix} a_4 \\ a_5 \end{pmatrix}, \quad (1.2)$$

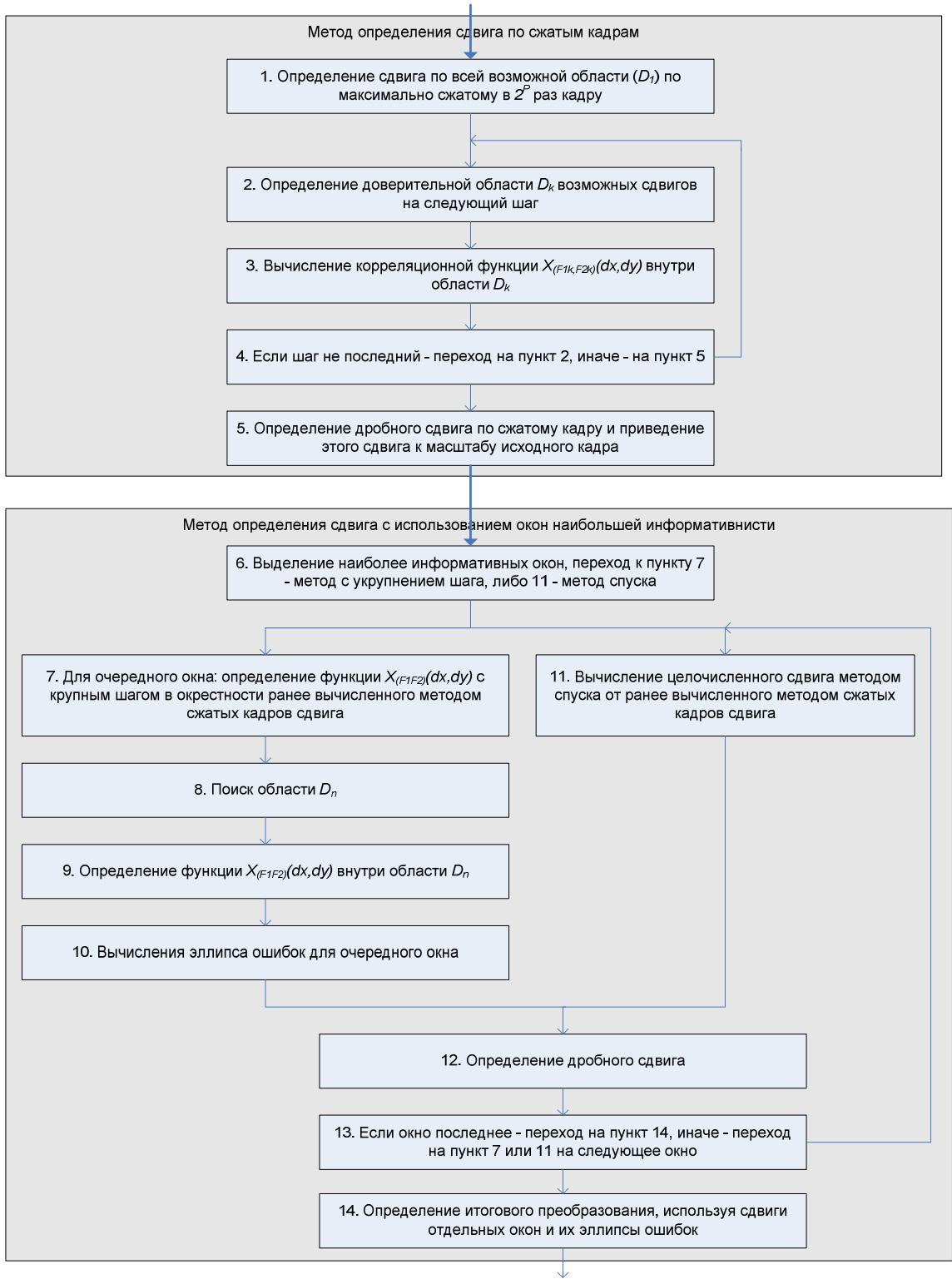
и квадратичное –

$$\begin{pmatrix} x(t+1) \\ y(t+1) \end{pmatrix} = \begin{pmatrix} a_0 + a_1x(t) + a_2y(t) + a_6x^2(t) + a_7x(t)y(t) \\ a_3 + a_4x(t) + a_5y(t) + a_6x(t)y(t) + a_7y^2(t) \end{pmatrix}. \quad (1.3)$$

Аффинное преобразование хорошо описывает произвольные небольшие геометрические трансформации кадров и является вполне достаточным для задачи стабилизации.

Проективное преобразование точно описывает изменение кадра, если сценой является плоскость. Это актуально для задач авиационного наблюдения, когда наблюдаемую поверхность Земли во многих случаях можно считать плоской. Недостатком этого преобразования является его нелинейность относительно параметров преобразования, что ведет к необходимости использовать нелинейные методы оптимизации для его поиска.

Квадратичное преобразование можно рассматривать как линейное относительно параметров приближение проективного преобразования. Основным его недостатком является несохранение свойства квадратичности при многократном применении (например, если от кадра  $F(t)$  к  $F(t+1)$  и от кадра  $F(t+1)$  к  $F(t+2)$  преобразования квадратичные, то от кадра  $F(t)$  к  $F(t+2)$  преобразование будет иметь четвертую степень).



**Рис. 1.1. Блок схема алгоритма определения трансформации кадров**  
 Принципиальная схема алгоритма представлена на Рис. 1.1.

Данный алгоритм состоит из двух частей: определение сдвига по сжатым кадрам и определение всех параметров трансформации с использованием окон наибольшей информативности.

Алгоритмом вычисляется только преобразование между кадрами, соседними по времени. Преобразование между кадрами, отстоящими далеко по времени рассчитывается как произведение преобразований для единичных моментов времени (как было сказано выше, такой подход невозможен для квадратичного преобразования).

Ключевым элементом обоих частей является алгоритм определения сдвига с оценкой качества принятия решения – доверительной области и ковариационной матрицы ошибок оценивания.

### 1.2.2 Определение сдвига с оценкой качества

Алгоритм определения преобразования одного кадра относительно другого всегда можно свести к следующей задаче оптимизации:

$$\hat{Q} = \arg \min_Q X(\|F(t-1)_{ij}\|, \|T_Q(F(t))_{ij}\|), \quad (1.4)$$

где  $\hat{Q}$  - оценка вектора параметров межкадрового преобразования изображения  $Q = (a_1, a_2 \dots)$ ,  $X$  - функция "непохожести", которая должна быть тем меньше, чем сильнее совпадают ее первый и второй аргументы,  $F(t)_{ij}$  - кадр, относящийся к моменту времени  $t$ ,  $T_Q$  - преобразование кадра с заданным вектором параметров преобразования.

Для достижения требуемых качеств - надежности и скорости - необходимо, во-первых, минимизировать функцию  $X(F_1, F_2)$  и, во-вторых, оценить ее поведение, которое характеризует качество принятия решения. Это особенно важно, если функция  $X(F_1, F_2)$  вычисляется за несколько шагов.

Информация о поведении функции  $X(F_1, F_2)$  позволяет достичь следующих желаемых результатов:

- ограничить объем вычислений без потери качества и надежности.

Зная ошибку определения искомых параметров, можно прервать вычисления тогда, когда эта ошибка станет приемлемой;

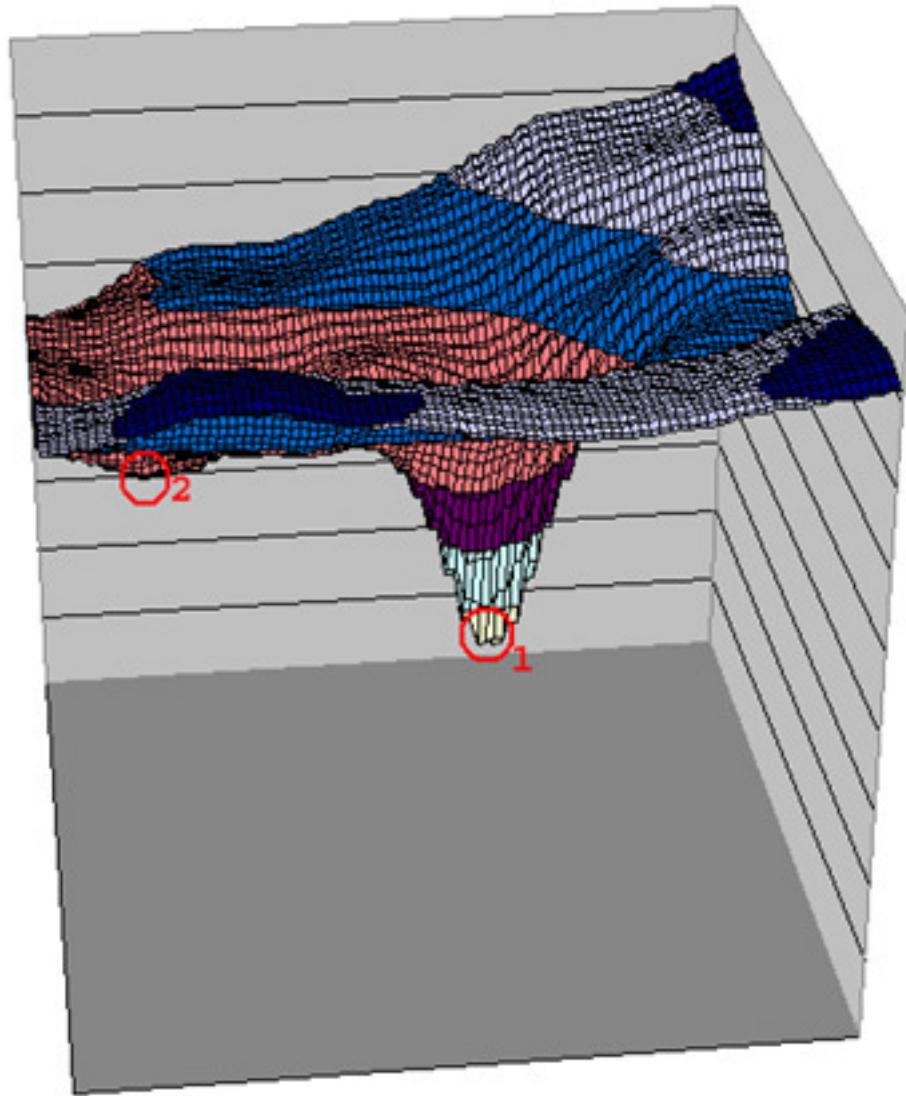
- повысить точность за счет взвешенного использования информации от различных точек. В предельном случае можно исключить использование заведомо ошибочной информации. Часто на кадре присутствуют участки или объекты, которые лишь вносят дополнительную ошибку (шумовую составляющую) в вычисляемые величины.
- прогнозировать ситуации, в которых точное определение сдвига затруднено или даже невозможно. В таких случаях вводится регуляризация решения и делается приближенная разумная оценка, либо вовсе исключается проблемный кадр из обработки.

Это позволяет существенно повысить точность, а, главное, надежность алгоритма и одновременно снизить вычислительные затраты, что увеличивает его скорость работы.

Все эти возможности более подробно описаны далее.

Базовым является алгоритм, использующий яркости точек входного изображения. Однако последовательное применение вышеизложенных принципов приводит к созданию алгоритма, представляющего собой синтез трех основных подходов.

Здесь под функцией  $X(F_1, F_2)$  подразумевается функция среднеквадратичной невязки или какие-либо ее обобщения. Вычисляется она достаточно быстро. При необходимости расчет проводиться параллельно на современных векторных процессорах. В качестве функции  $T_Q(F)$  выбрана комбинация линейных функций целочисленного сдвига. Ее вычисление занимает малое время по сравнению с остальными вычислениями.



**Рис. 1.2. Типичный вид функции корреляции. Видно два минимума 1 - главный и 2 - побочный.**

Функция невязки  $X(F_1, F_2)$  может быть представлена в виде<sup>1</sup>

$$X(F(t-1), T_Q(F(t))) = \frac{1}{S(M)} \sum_{(i,j) \in M} (F(t-1)_{ij} - F(t)_{i+dx, j+dy})^2 , \quad (1.5)$$

а само преобразование оценивается исходя из соотношения:

---

<sup>1</sup> Здесь и далее изображение считается черно-белым. Формула легко обобщается на случай цветных изображений – вместо квадрата разности яркости используется сумма квадратов цветовых компонент.

$$\hat{Q} = (d\hat{x}, d\hat{y}) = \arg \min_{(dx, dy) \in D} X(F(t-1), T_Q(F(t))), \quad (1.6)$$

где  $\hat{Q} = (d\hat{x}, d\hat{y})$  - сдвиги вдоль обеих осей,  $D$  - область возможных сдвигов, по которой проводится поиск.  $M(dx, dy)$  - область, по которой производится вычисление невязки между кадрами (например, область, для которой точки  $(i, j)$  и  $(i + dx, j + dy)$  принадлежат  $F(t-1)$  и  $F(t)$  соответственно, т.е. область пересечения кадров),  $S(M)$  - площадь области  $M(dx, dy)$  (число точек).

Такую функцию  $X(F_1, F_2)$  будем называть функцией корреляции двух кадров. Она зависит от двух аргументов:  $(dx, dy)$ , представляющих собой сдвиги по каждой из осей. В дальнейшем будем обозначать ее  $X_{(F_1 F_2)}(dx, dy)$ .

Важной особенностью здесь является многомодальность функции  $X_{(F_1 F_2)}(dx, dy)$  (Рис. 1.1). Для нахождения сдвига использовать методы спуска надо с большой осторожностью - необходим перебор всех возможных вариантов. За счет большой скорости вычисления функции корреляции способ перебора не приводит к снижению скорости алгоритма, однако сильно повышает надежность определения сдвига.

Метод корреляции для всего кадра слишком трудоемок (см. результаты в конце раздела). В предлагаемом алгоритме для уменьшения вычислительных затрат используются следующие методы:

- метод сжатия кадра;
- метод выделения некоторой области в кадре;
- метод уменьшения области возможных сдвигов.

При этом, в соответствии с принципами, изложенными выше, вычисляется доверительная область оценок на каждом шаге и проводится столько вычислений, сколько необходимо для достижения приемлемого качества.

Предлагается следующая схема. Пусть вычислены значения функции  $X_{(F_1 F_2)}(dx, dy)$  в некоторой области  $D$ . Искомые параметры соответствуют

минимуму функции, поэтому для оценки их качества необходимо исследовать поведение функции  $X_{(F_1F_2)}(dx, dy)$  в районе ее минимума.

Предположим, что шум в пикселях нормальный некоррелированный, при этом значения функции  $X_{(F_1F_2)}(dx, dy)$  имеют  $\chi^2$  - распределение, а ее значение в минимуме будем считать математическим ожиданием.

Тогда дисперсия шума в отдельных пикселях равна

$$\sigma_p = \sqrt{\frac{X_m}{N_p}}, \quad (1.7)$$

где  $X_m$  - значение функции  $X_{(F_1F_2)}(dx, dy)$  в минимуме,  $N_p$  - число точек в области  $M(dx, dy)$ , соответствующей минимуму.

Математическое ожидание и дисперсия значений функции  $X_{(F_1F_2)}(dx, dy)$  определяются соотношениями

$$m_i = \frac{X_m}{N_p} \cdot N_i, \quad \sigma_i = \frac{X_m}{N_p} \cdot \sqrt{2N_i}, \quad (1.8)$$

где  $m_i$  - математическое ожидание,  $\sigma_i$  - дисперсия,  $N_i$  - количество точек в области  $M(dx_i, dy_i)$ , по которой подсчитана функция  $X_{(F_1F_2)}(dx_i, dy_i)$ .

Доверительная область, в которой могут лежать значения функции  $X_{(F_1F_2)}(dx_i, dy_i)$ , обусловленные шумом, определяется следующим выражением:

$$\tilde{D} = \{dx_i, dy_i : X_{(F_1F_2)}(dx_i, dy_i) < m_i + B \cdot \sigma_i\}, \quad (1.9)$$

где  $B$  - порог, определяющий доверительную вероятность ошибки.

Выбор порога  $B$  позволяет определить доверительную область минимума даже для случаев негауссовых и коррелированных шумов.

Таким образом, для точки минимума функции  $X_{(F_1F_2)}(dx_i, dy_i)$  мы получили область, в которой с заданной вероятностью находится истинное решение. Эта область достаточно полно характеризует качество определения сдвига.

Однако такая оценка является нелинейной и не всегда удобна для анализа. Ее можно заменить более простой, ограничившись вторыми моментами распределения, т.е. использовать гауссовскую модель ошибок с ковариационной матрицей оценивания искомой величины (в данном случае - сдвига). Обратная ковариационная матрица ошибок оценивания сдвигов:

$$C^{-1} = \begin{vmatrix} A & B \\ B & E \end{vmatrix}, \quad (1.10)$$

имеет размерность 2 на 2, а ее элементы представляют собой оценки параметров эллипса, аппроксимирующего область  $D$  методом наименьших квадратов в виде

$$A(x - x_m)^2 + 2B(x - x_m)(y - y_m) + E(y - y_m)^2 = 0, \quad (1.11)$$

где  $(x_m, y_m)$  - точка минимума функции  $X_{(F_1 F_2)}(dx, dy)$ .

### 1.2.3 Определение сдвига по пирамиде сжатых кадров

Одним из способов повышения скорости (а, зачастую, и качества) работы алгоритма является процедура оценки сдвига по сжатым кадрам.

Предварительно следует сжать исходные кадры. Т.к. из одного кадра необходимо получить сразу последовательность кадров разного масштаба (Рис. 1.3), то удобнее всего использовать сжатие в 2 раза. Тогда:

$$F^{k+1} = \Sigma(F^k), \quad (1.12)$$

где  $F^k$  - кадр, сжатый в  $k$  раз (т.е. имеющий в  $k$  раз меньший линейный размер, чем исходный),  $\Sigma(F)$  - некоторая функция сжатия в 2 раза.

Для получения ряда кадров с разным масштабом последовательно применяется функцию сжатия сначала к исходному кадру, а потом и к сжатым.



Рис. 1.3. Последовательность (пирамида) сжатых в 2 раза кадров.

В качестве функции сжатия в данном случае выбрана следующая:

$$\Sigma(F)_{ij} = \sum_{l,k=0}^1 F_{i+l,j+k}, \quad (1.13)$$

По такой пирамиде сжатых кадров выполняется процедура определения сдвига. Работает она по следующей итеративной схеме:

1. Для первого шага выбирается область  $D_1$  как прямоугольник заданных размеров (практика показывает, что сдвиги в пол кадра определяются достаточно надёжно). Внутри этой области считается функция  $X_{(F_1^P F_2^P)}(dx, dy)$  - т.е. корреляционная функция двух сжатых кадров. Величина  $P$  выбирается такой, чтобы сжатые в  $2^P$  раз кадры имели размер сторон 15–30 точек (см. Рис. 1.1, пункт 1).
2. Выбирается область  $D_k$  по формулам (1.7), (1.8) и (1.9) только вместо функции  $X_{(F_1 F_2)}(dx, dy)$  используется  $X_{(F_1^{P-k} F_2^{P-k})}(dx, dy)$  - пункт 2 Рис. 1.1.
3. На  $k$ -ом шаге берутся кадры  $F_1^{P-k}$  и  $F_2^{P-k}$  - сжатые в  $P-k$  раз исходные кадры. Для них вычисляется функция  $X_{(F_1^{P-k} F_2^{P-k})}(dx, dy)$  внутри области  $D_{k+1}$  (см. Рис. 1.1, пункт 3).
4. Если шаг  $k$  – не последний – переход на следующий шаг  $k+1$  на пункт 2. Иначе – переход на пункт 5. Здесь следует отметить, что данную процедуру целесообразно прерывать задолго до достижения шага, на котором используются кадры исходного масштаба, т.к. сравнение целого кадра требует большого количества операций. Удобнее сделать 2–3 шага, а затем перейти к описанной ниже процедуре определения параметров преобразования с использованием нескольких окон (см. Рис. 1.1, пункт 4).

5. Для последнего шага определяется дробный сдвиг, и все сдвиги приводятся к масштабу исходного кадра – пункт 5 Рис. 1.1.

Метод с использованием сжатия кадра обладает существенными преимуществами по сравнению с другими вышеописанными методами:

- Он позволяет вычислять сдвиги до полукадра, что трудно осуществимо другими способами ввиду большого количества требуемых операций.
- Вероятность ошибок существенно меньше, т.к. можно полностью проверить все возможные варианты сдвигов, а значит нельзя попасть в локальный минимум. Важно, что здесь сравнивается весь кадр, а не какая-то его часть. Это опять-таки приводит к уменьшению количества ошибок.
- Метод хорошо работает со смазанными кадрами или кадрами, в которых преобладают низкие пространственные частоты.

Следует заметить, что, несмотря на то, что при сжатии не определяются угол поворота и коэффициент масштабирования, они вычисляются в процедуре уточнения, а их наличие в кадре слабо сказывается на работоспособности алгоритма определения сдвига с помощью масштабирования.

#### **1.2.4 Выделение информативных областей**

Для уменьшения количества вычислений используется не весь кадр, а только некоторая его часть (пункт 6 Рис. 1.1). Произвольно ее выбирать нельзя, т.к. зачастую в кадре попадаются неинформативные участки, ошибка определения сдвига по которым, даже полным перебором, может достигать размеров самого сдвига.

Удобно выбирать некоторое количество прямоугольных областей с наибольшей информативностью, и только по ним определять невязку. Т.е. под областью  $M$  следует понимать объединение нескольких прямоугольников (корреляционных окон). Это позволяет существенно снизить количество выполняемых операций и ускорить алгоритм.



**Рис. 1.4. Пример выделения наиболее информативных областей кадра.**

В целом, метод выбора окон с наибольшей информативностью обладает следующими преимуществами:

- Позволяет снизить количество вычислений, а, следовательно, повысить скорость алгоритма.
- Позволяет уменьшить влияние шумов, за счет исключения из рассмотрения участков с низким отношением сигнал-шум, которые не только не прибавляют точности определения сдвига, но и уменьшают ее.
- Позволяет избежать краевых эффектов, т.к. наиболее информативные окна можно выбирать вдали от краев изображения.

Корреляционные окна выбираются из нескольких соображений. Во-первых, они должны обладать максимальной информативностью. Во-вторых - отстоять как можно дальше друг от друга. Т.е. критерий выбора  $k$ -ой точки:

$$\vec{r}_k = \arg \max_{\vec{r}_k} (I(\vec{r}_k) \cdot (\min_{i=1..k-1} |\vec{r}_k - \vec{r}_i| + C)), \quad (1.14)$$

где  $I(\vec{r}_k)$  - информативность точки  $\vec{r}_k$ ,  $C$  – некоторая константа, подбираемая экспериментально (можно взять порядка половины размера кадра).

Информативность окна вычисляется как минимум автокорреляционной функции вдоль нескольких направлений, т.е. по формуле:

$$I(\vec{r}_k) = \min_{l=1..L} \frac{1}{\sqrt{dx^2(l) + dy^2(l)}} \sum_{(i,j) \in W(\vec{r}_k)} (F_{ij} - F_{i+dx(l), j+dy(l)})^2, \quad (1.15)$$

где  $(dx(l), dy(l))$  – набор направлений,  $W(\vec{r}_k)$  - корреляционное окно с центром в точке  $\vec{r}_k$ ,  $L$  – количество направлений.

В данной работе использовались следующие параметры:

$$L = 4,$$

$$(dx(1), dy(1)) = (0, 1),$$

$$(dx(2), dy(2)) = (1, 1), \quad (1.16)$$

$$(dx(3), dy(3)) = (1, 0),$$

$$(dx(4), dy(4)) = (1, -1).$$

Выбор автокорреляционной функции в качестве критерия информативности обоснован тем, что в качестве целевой функции используется функция невязки, а, следовательно, ее крутизна пропорциональна точности оценки сдвига.

Здесь необходимо заметить, что после того, как вычислен сдвиг по первому окну или алгоритмом с пирамидой сжатых кадров, за область  $D$  удобно принять окрестность вычисленного сдвига. Это сокращает её площадь по сравнению с априорной, т.е. объём вычислений.

### 1.2.5 Определение сдвига отдельного окна

Определение сдвига каждого окна можно проводить простым перебором всех возможных сдвигов. Такой метод, однако, весьма требователен к ресурсам процессора. Для улучшения производительности предлагается два метода – метод с прореживанием и метод спуска.

Для улучшения качества определения сдвига предложен метод накопления.

#### 1.2.5.1 Метод с прореживанием

Для уменьшения количества операций предложена следующая схема. Сначала перебирается область  $D$  с крупным шагом (см. Рис. 1.5), т.е. перебираются не все точки, а пропускаются с каким-то шагом. В данной работе был выбран шаг 4 (пункт 7 Рис. 1.1).

По вычисленным значениям функции  $X_{(F_1F_2)}(dx, dy)$  в соответствии с формулами (1.7), (1.8) и (1.9) находится доверительная область  $\tilde{D}_n$  вероятного нахождения минимума. После этого надо область  $\tilde{D}_n$  обходится с шагом 1.

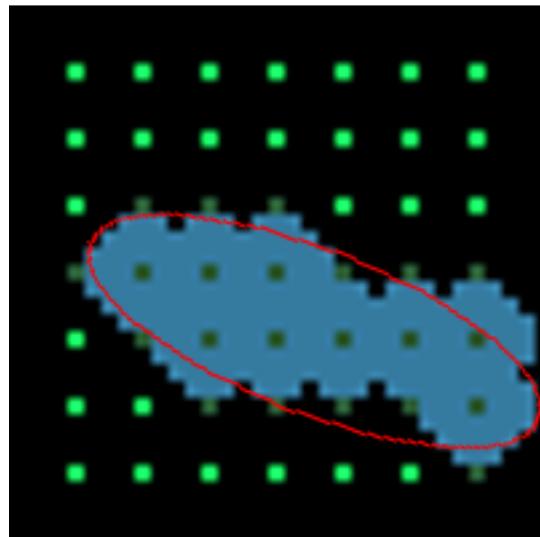


Рис. 1.5. Область, по которой вычислялась функция  $X_{(F_1F_2)}(dx, dy)$ .

Зеленым цветом обозначены точки, в которых функция вычислялась при переборе с крупным шагом. Тёмно-зелёным – область  $\tilde{D}_n$ , светло-зелёным – точки, не попавшие в область  $\tilde{D}_n$ . Синим – область  $D_n$ . Красный эллипс – эллипс ошибок.

Поиск минимума проводится не по всей области вероятного нахождения минимума. Предлагается начать двигаться от центра области  $D$  «по спирали», т.е. по квадратам, постепенно увеличивая их сторону. При нахождении минимума обходится еще несколько крупных витков спирали. Если в них не окажется точки с меньшим значением невязки, то поиск останавливается. Такой метод приводит к уменьшению количества операций в случаях малых сдвигов, оставляя при этом возможность определять и большие сдвиги. Однако он может привести к ошибкам вследствие выбора не глобального, а локального минимума.

Данный подход обладает полезными свойствами:

- Позволяет уменьшить количество вычислений за счет не полного перебора всех вариантов, не увеличивая при этом вероятности попадания в локальный минимум.
- Позволяет учесть не только возможность существования побочных минимумов (т.е. производится обход всей области, пусть и с крупным шагом), но и наличие зашумленных кадров с плохо обусловленным минимумом корреляционной функции.

### 1.2.5.2 Метод спуска

Помимо метода полного перебора возможно также использование метода спуска (пункт 11 Рис. 1.1).

Он работает по следующему итеративному алгоритму:

1. На первом шаге выбирается некоторая начальная точка  $(dx_0, dy_0)$ . Ею может быть точка  $(0,0)$ , либо некоторое предсказание сдвига, полученное или из оценки сдвига, или по предыдущим окнам, или по методу с использованием сжатых кадров.
2. Перебираются все точки в некоторой окрестности  $D_i$  текущей точки  $(dx_i, dy_i)$ . Размеры окрестности выбираются

экспериментально – например, в данной работе в качестве такой окрестности был выбран квадрат размером 3 на 3 пикселя с центром в точке  $(dx_i, dy_i)$ .

3. В выбранной области  $D_i$  считается значение функции  $X_{(F_1F_2)}(dx, dy)$  и выбирается ее минимум. Если точка минимума находится на границе области  $D_i$ , то она выбирается в качестве точки  $(dx_{i+1}, dy_{i+1})$  и переходит на пункт 2. Иначе – точка минимума считается точкой глобального минимума, т.е. искомой.

Результат вычисления функции  $X_{(F_1F_2)}(dx, dy)$  по данному методу представлен на Рис. 1.6. Здесь для наглядности размер области  $D_i$  был установлен 5 на 5 пикселей.



**Рис. 1.6. Вид функции  $X_{(F_1F_2)}(dx, dy)$ , вычисленной методом спуска. Более яркие точки соответствуют большим значениям функции.**

Данный алгоритм работает значительно быстрее алгоритма с перебором, но и не исключает возможности попадания в побочный локальный минимум. Для устранения такой вероятности необходимо, чтобы сдвиг относительно точки  $(dx_0, dy_0)$  был небольшим. Это достигается либо при небольших колебаниях изображения, либо при достаточно точных предсказаниях сдвига. Как показывает практика, предсказания сдвига, сделанные на основе метода с использованием сжатых кадров, достаточно точны и в подавляющем большинстве случаев позволяют точно определить точку глобального минимума функции  $X_{(F_1F_2)}(dx, dy)$  методом спуска.

### 1.2.5.3 Использование накопления

Для улучшения качества работы алгоритмов предлагается использовать эталонные кадры. Т.е. сравнивать текущий кадр не с предыдущим, а с эталонным. Этalonный кадр создается путем накопления кадров с учетом измеренных сдвигов.

При этом целесообразно хранить эталон не всего кадра, а только выбранных окон наибольшей информативности. Это позволяет не только уменьшить количество выполняемых операций, но и улучшить качество эталонов. Это связано с тем, что целиком кадр не всегда точно подчиняется выбранной модели искажений. А в небольшом окне отклонения от модели менее существенны и ими можно пренебречь.

Хорошим примером такого накопления может служить алгоритм, предложенный в [45]. Он разрабатывался для задачи сопровождения объектов (см. главу 3). Для задачи стабилизации удобнее использовать более простой и быстрый алгоритм, описанный ниже.

В качестве начального эталона окна используется окно из первого кадра последовательности. Далее, после определения сдвига очередного окна, необходимо обновить эталон с учетом текущего кадра и измеренного сдвига. Обновление эталона производится по следующей формуле:

$$E_{ij}(t+1) = (1 - W) \cdot E_{ij}(t) + W \cdot T_Q(F(t))_{ij}, \quad (1.17)$$

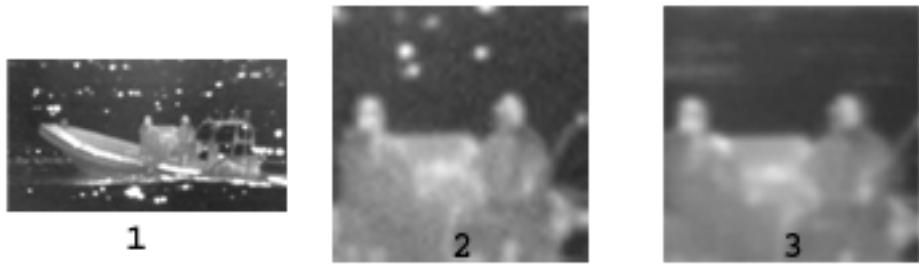
$E_{ij}(t)$  – кадр эталона,  $T_Q(F(t))$  – преобразование текущего кадра в соответствии с измеренными параметрами трансформации  $Q$ ,  $W$  – весовой коэффициент, определяемый формулой

$$W = \frac{1}{\max\{t - t_0, T_w\}}, \quad (1.18)$$

где  $T_w$  – время накопления эталона,  $t_0$  – время начала хранения эталона.

Время накопления  $T_w$  выбирается экспериментально, исходя из точности соответствия исходной видеопоследовательности выбранной модели. Если

соответствие хорошее, то целесообразно выбирать большое время накопления. Если же имеют место некоторые несоответствия модели и реальной последовательности (например, наличие движущихся объектов в кадре, сложные геометрические преобразования от кадра к кадру и т.д.), то при большом времени накопления эталон может размазываться и становиться непригодным к определению сдвига.



**Рис. 1.7. 1 - Исходный кадр – лодка, плывущая на фоне ярких бликов воды, 2 – фрагмент этого кадра – хорошо видны сильные шумы и мешающие объекты фона, 3- эталон этого фрагмента – шумы сильно подавлены, а мешающий фон смазан.**

Следует обратить внимание на то, что в данном случае функция  $T_Q(F(t))$  не является функцией целочисленного сдвига, как всюду в этой работе. Для создания качественного эталона эта функция должна точно учитывать все измеренные параметры преобразования, т.е. дробные сдвиги, угол поворота и т.д.

Для использования эталона для определения сдвига окна необходимо вместо сдвига относительно опорного кадра вычислять сдвиг относительно эталона, т.е. вычислять не функцию  $X_{(F_1F_2)}(dx, dy)$ , а функцию  $X_{(EF_2)}(dx, dy)$ .

Использование эталонов дает целый ряд преимуществ по сравнению с определением сдвига между последовательными кадрами (см. Рис. 1.7):

- Снижается ошибка определения сдвига, т.к. в эталоне уровень шума значительно ниже, чем в исходной последовательности.

- Снижается ошибка определения сдвига между кадрами с моментами времени, далеко отстоящими друг от друга. Она появлялась за счет интегрирования ошибки сдвига между последовательными кадрами. Избежать её трудно, если сравнивать кадр не с предыдущим, а с отстоящим дальше по времени. Т.к. если кадр изменяется (пусть даже медленно), то кадр, отстоящий далеко по времени уже плохо согласуется с текущим. В эталонах используется информация и из последних кадров, в то время как ошибка определения сдвига не интегрируется.
- Если стоит задача стабилизировать не весь кадр, а некоторый объект, который смещается относительного всего остального кадра, то введение эталонов позволяет уменьшить влияние фона на определение сдвига. В этом случае удерживаемый объект накапливается хорошо, в то время как фон размазывается и становится малоинформационным, а, следовательно, слабо влияет на определение сдвига (детали см. в главе 3).

#### 1.2.5.4 Уточнение сдвига с точностью до долей пикселя

Для улучшения работы алгоритма необходимо определять сдвиг с точностью более одного пикселя (пункт 12 Рис. 1.1). Это особенно важно в рассмотренном ниже методе поиска сдвига по сжатым кадрам, т.к. там дробный сдвиг сжатого кадра может соответствовать большому целочисленному сдвигу исходного изображения.

Различные способы отыскания дробного сдвига подробно рассмотрены в статье [12]. Каждый из них можно использовать для уточнения сдвига и в данных алгоритмах.

Здесь, однако, есть одна особенность. Она заключается в том, что в ходе определения целочисленного сдвига уже были вычислены значения функции  $X_{(F_1F_2)}(dx, dy)$  в окрестности минимума. Т.к. сдвиг соответствует минимуму

этой функции, то можно аппроксимировать её значения в дискретных точках гладкой поверхностью и найти минимум на этой поверхности.

Предлагается брать 9 точек вокруг максимума и методом наименьших квадратов находить параметры параболоида  $z = ax^2 + bxy + cy^2 + dx + ey$ . Тогда минимум этого параболоида и соответствует минимуму функции  $X_{(F_1F_2)}(dx, dy)$ , а значит и искомому сдвигу. Этот метод дает несколько меньшую точность, чем рассмотренные в статье [12], но, в силу указанных выше причин, работает существенно быстрее.

### 1.2.6 Вычисление полного преобразования кадра

Для вычисления всех параметров трансформации необходимо учесть оценку сдвига всех окон. Вычисленные оценки локальных сдвигов естественно отнести к центрам соответствующих окон. Вычисление вектора преобразования следует проводить по следующему критерию:

$$\hat{Q} = \arg \min_Q \sum_{k=1}^N (\vec{\rho}_k - \tau_Q(\vec{r}_k)) C_k^{-1} (\vec{\rho}_k - \tau_Q(\vec{r}_k))^T, \quad (1.19)$$

где  $\vec{r}_k$  - координаты центров окон в кадре  $F(t-1)$ ,  $\vec{\rho}_k$  - координаты тех же областей в кадре  $F(t)$  (вычисленные по формулам (1.5) и (1.6)),  $N$  - число окон,  $C^{-1}$  - ковариационная матрица ошибок,  $\tau_Q(\vec{r}_k)$  - функция преобразования координат,  $T$  - операция транспонирования.

Функция  $\tau_Q(\vec{r}_k)$  определяет модель преобразования кадра: аффинные преобразования, проективные, квадратичные и т.д. Это приводит к задаче минимизации в  $n$ -мерном пространстве, где  $n$  равно шести для аффинных преобразований, восьми для проективных и т.д.

В случае аффинных преобразований задача сводится к решению системы линейных уравнений. Введение регуляризации [14] при решении соответствующей системы уравнений позволяет получить разумное приблизительное решение даже для больших ошибок оценивания сдвигов окон, т.е. для случаев низкокачественных входных изображений.

Алгоритм регуляризации использует вычисление решения системы уравнений методом Якоби. Этот метод основан на поиске собственных векторов и собственных значений матрицы системы уравнений. В таком случае следует малые собственные значения заменить на некоторую минимальную величину (в данной работе был выбран порог 0.01). Метод позволяет снизить ошибки в сложных случаях, например в кадрах с линейной структурой (см. Рис. 1.8).



**Рис. 1.8. Пример кадра, для которого существенную роль играет регуляризация решения системы уравнений. Без ее использования ошибка определения поворота может достигать 10°, а ошибка определения сдвига – 20 пикселей. С регуляризацией аналогичные параметры не более 1° и 3 пикселей.**

В случае присутствия на изображении движущихся относительно фона объектов они исключаются из вычислений, используя один из стандартных методов обработки исключений [46] (здесь был выбран метод RANSAC).

Следует отметить одну интересную особенность предложенного подхода. Задачу можно решать и нелинейным методом:

$$\hat{Q} = \arg \min_Q \sum_{k=1}^N X_k(\tau_Q(\vec{r}_k) - \vec{\rho}_k), \quad (1.20)$$

где  $X_k(\tau_Q(\vec{r}_k) - \vec{\rho}_k)$  - подсчитанные заранее функции корреляции для отдельных окон. Этот метод является нелинейным и позволяет учесть различные нелинейные эффекты на изображении. Однако, в связи с относительно низкой надежностью и большой вычислительной сложностью методов нелинейной оптимизации, метод не был использован в конечном варианте предложенного в данной работе алгоритма определения геометрических преобразований.

### 1.2.7 Анализ и результаты

Описанный выше алгоритм исследовался с целью выяснения качества его работы. При этом считались С.К.О. ошибок определения параметров трансформации, (т.е. сдвигов, углов поворота и коэффициентов масштабирования), вероятность сбоя и скорость работы.

Качество работы алгоритмов оценивалось пятью способами:

1. На основе модельных кадров. Т.е. одно и то же изображение преобразовывалось с заданными параметрами и подавалось на вход алгоритма. При этом можно точно оценить ошибки определения параметров преобразования. В то же время, такое моделирование не отражает многих проблем, с которыми приходится сталкиваться в реальности.
2. На реальных видеопоследовательностях. По ним можно определить сдвиг лишь приближенно и субъективно, и здесь основным критерием является вероятность сбоя, т.е. «очевидной» ошибки, превышающей несколько пикселей. Использовалось более ста видеопоследовательностей различного характера, в том числе из популярных общедоступных баз данных ([http://www.elec.qmul.ac.uk/staffinfo/andrea/avss2007\\_d.html](http://www.elec.qmul.ac.uk/staffinfo/andrea/avss2007_d.html)).

3. На реальных последовательностях с известным сдвигом. Был проведен эксперимент по определению сдвига кадров неалгоритмическим методом. Проведено сравнение этих экспериментальных данных с результатами работы алгоритма.
4. Работа в условиях ограничения времени обработки одного кадра. При этом оценивались средние ошибки определения сдвигов и углов поворота на синтетических последовательностях.
5. Сравнение с другими алгоритмами и программными продуктами.

Исследование работы алгоритмов проводилось на ПК со следующими характеристиками:

- Процессор Intel® Pentium® 4 2.4ГГц.
- Память 512Мб.
- Видеокарта NVIDIA GeForce4® MX 440.
- Операционная система Microsoft® Windows™ 2000.

**Табл. 1.1 Результаты работы алгоритмов определения сдвига.**

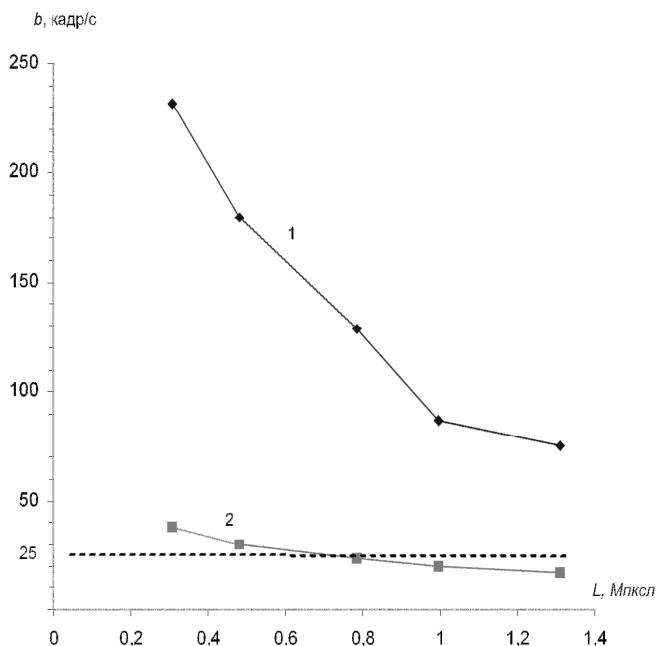
Размер изображения (пиксели)	Сравнение всего кадра		Только метод корреляционных окон		Полнофункциональный алгоритм		
	С.К.О. определения сдвига (пиксели)	Количество кадров в секунду	С.К.О. определения сдвига (пиксели)	Количество кадров в секунду	С.К.О. определения сдвига (пиксели)	Количество кадров в секунду	Максимальный сдвиг (пиксели)
320x240	0.07	0.9	0.04	206	0.08	714	120
640x480	0.07	0.17	0.03	152	0.07	232	260
800x600	0.07	0.1	0.03	116	0.08	180	330
1024x768	0.07	0.055	0.03	83	0.11	129	400
1152x864	0.07	0.044	0.02	42	0.11	87	480
1280x1024	0.07	0.038	0.04	37	0.08	75	540

Сравнивались три различных вариантов алгоритма. Первый – опорный – в котором минимум невязки вычислялся полным перебором для всего кадра.

Второй – без использования пирамиды сжатых кадров, и третий - с ее использованием.

В Табл. 1.1 и на Рис. 1.9 даны результаты работы различных алгоритмов определения сдвига. Сдвиги моделировались случайные по обеим осям, равномерно распределенные в интервале от  $-30$  до  $+30$  пикселей.

В данном случае, под скоростью работы подразумевается только скорость работы алгоритмов определения сдвига, в то время как чтение кадров, генерация выходного кадра и их отображение не учитывается.



**Рис. 1.9. Зависимость скорости обработки от размера кадра: 1 – скорость работы описанных алгоритмов, 2 - скорость полной обработки видеопоследовательности.**

Особенно сильно различаются скорости работы различных алгоритмов. Алгоритм сравнения всего кадра работает на 3 порядка медленнее остальных предложенных. Следует заметить, что трудоемкость первого алгоритма растёт пропорционально количеству точек кадра (т.е. квадратично относительно длины стороны). Трудоёмкость же второго и третьего растёт существенно медленнее.

Также следует заметить, что максимально возможные сдвиги первых двух алгоритмов составляли 60 пикселей. При увеличении этого параметра скорость алгоритмов падает пропорционально его квадрату.

В полнофункциональном алгоритме максимальный сдвиг, который мог быть оценен алгоритмом, был выбран размером в полкадра. Максимальный сдвиг, который реально может быть оценен, определялся на отдельной тестовой последовательности как сдвиг, при котором точность алгоритма падает вдвое.



**Рис. 1.10. Пример кадров из «реальной» тестовой последовательности. Слева – кадр относительно хорошего качества, справа – типичный кадр для данной последовательности.**

Качество же работы приблизительно одинаковое. Более того, метод оценки с использованием только корреляционных окон дает примерно вдвое меньшую ошибку определения сдвига, чем даже метод сравнения всего кадра.

**Табл. 1.2 Вероятность сбоя на реальных последовательностях без поворотов.**

Сравнение всего кадра		Только метод корреляционных окон		Полнофункциональный алгоритм	
Вероятность сбоя	С.К.О. по сдвигу сбоях при (пиксели)	Вероятность сбоя	С.К.О. по сдвигу сбоях при (пиксели)	Вероятность сбоя	С.К.О. по сдвигу сбоях при (пиксели)
0.3%	7	1.3%	20	<0.1%	0

Алгоритмы тестировались и на реальных последовательностях. (Табл. 1.2). Использовалась последовательность, снятая низкокачественной камерой с большой тряской. Изображение в таких условиях получалось смазанное, перекошенное (причем и величина, и направление перекосов и смазов менялась от кадра к кадру), сильно зашумленное, с заметными искажениями за счет неидеальной оптики камеры (Рис. 1.10).

Для каждого сбоя считалась его величина путем сопоставления двух последовательных кадров «вручную».

В результате полнофункциональный алгоритм показал идеальные результаты, в то время как остальные давали небольшое количество заметных сбоев.

В Табл. 1.3 и Табл. 1.4 приведены результаты работы различных алгоритмов определения сдвига, поворота и угла масштабирования.

**Табл. 1.3 Точность определения угла поворота и коэффициента масштабирования.**

Размер изображения (пиксели)	Только метод корреляционных окон			Полнофункциональный алгоритм			Максимальный угол поворота (градусы)
	С.К.О. определения угла (градусы)	С.К.О. коэффициента масштабирования	Количество кадров в секунду	С.К.О. определения угла (градусы)	С.К.О. коэффициента масштабирования	Количество кадров в секунду	
320x240	0.034	0.2%	104	0.036	0.2%	161	10
640x480	0.036	0.18%	84	0.038	0.18%	123	8
800x600	0.041	0.15%	63	0.040	0.15%	75	8

Сдвиги моделировались случайные по обеим осям, равномерно распределенные в интервале от  $-10$  до  $+10$  пикселей. Поворот также выбирался случайным, равномерно распределённым в интервале от  $-2^\circ$  до  $+2^\circ$ .

Коэффициент масштабирования менялся линейно так, что на 100-м кадре последовательности масштаб был в 4 раза больше, чем на первом.

Здесь скорость работы заметно ниже, чем без определения угла поворота, однако по-прежнему остается достаточно высокой даже для кадров 800 на 600 пикселей.

Что касается реальных, а не модельных последовательностей, то здесь имеет место та же ситуация, что и для случая без определения сдвига.

Сбои в работе редки, особенно у полнофункционального алгоритма.

**Табл. 1.4 Вероятность сбоя на реальных последовательностях с поворотом.**

Только метод корреляционных окон			Полнофункциональный алгоритм		
Вероятность сбоя	С.К.О. по сдвигу при сбоях (пиксели)	С.К.О. по углу при сбоях (градусы)	Вероятность сбоя	С.К.О. по сдвигу при сбоях (пиксели)	С.К.О. по углу при сбоях (градусы)
3%	30	10	0.6%	15	10

Особое внимание уделялось исследованию функционирования алгоритмов при ограничении скорости работы. Оно учитывает, что время обработки разных кадров может сильно отличаться друг от друга. И если для определенной последовательности большинство кадров может быть обработано в реальном времени, то для некоторых из них требуется значительно большее время. Чтобы не выходить за пределы реального времени, введена возможность ограничения периода обработки каждого кадра. При этом если кадр не был обработан полностью, в качестве оценки параметров его преобразования берется промежуточная.

В ходе работы алгоритма практически всегда существует некоторая оценка сдвига. В частности:

- В методе определения сдвига по сжатым кадрам можно прекратить обработку на любом масштабе кадров. А вычисление функции  $X_{(F_1^{P-k} F_2^{P-k})}(dx, dy)$  можно провести не по всей области  $D_n$ . При этом увеличивается вероятность попадания в побочный минимум, однако в целом, такая оценка сдвига состоятельна. После этого можно определить дробный сдвиг и привести его к масштабу исходного кадра. Получившаяся оценка сдвига не всегда точна, но в большинстве случаев ошибка не превышает 1-2 пикселей.
- В методе определения параметров трансформации по окнам можно остановить выполнение уже после определения сдвига по одному окну. Далее вычислить общее преобразование по формуле с учетом только просчитанных окон.

Таким образом, при прерывании обработки кадра на каком-то внутреннем шаге алгоритма получается оценка параметров трансформации, которая в большинстве случаев достаточно точна.

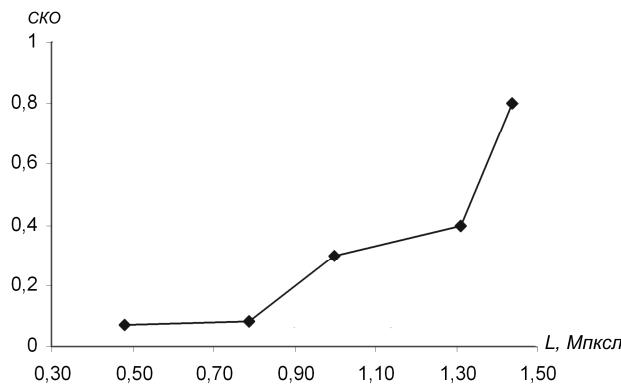
**Табл. 1.5 Точность определения сдвига при ограничении скорости работы.**

Размер изображения (пиксели)	Количество точек на изображении (мегапиксели)	С.К.О. определения сдвига (пиксели)
800x600	0.48	0.07-0.08
1024x768	0.79	0.07-0.09
1152x864	1.00	0.3-0.4
1280x1024	1.31	0.3-0.4
1600x900	1.44	0.8-1.0
1600x1024	1.64	обработка отключается

Тестирование проводилось на синтетических последовательностях. Средний сдвиг составлял 20 пикселей.

При этом исследовалась скорость не алгоритма определения сдвига, а программы в целом, включая чтение из файла, собственно алгоритмы стабилизации и генерации выходного кадра и отображение на экран.

Результаты исследования приведены в Табл. 1.5 и на Рис. 1.11.



**Рис. 1.11. Зависимость ошибки (в пикселях) от размера кадра при ограничении времени обработки каждого кадра.**

Помимо СКО ошибки исследовалась скорость ее накопления. Накопление ошибки связано с тем, что преобразование кадра от момента времени  $t$  к моменту времени  $t + N$  вычисляется как произведение  $N$  преобразований, вычисленных для соседних по времени кадров.

Если не использовать накопление эталона, то ошибка накапливается по известному закону (подтвержденному экспериментально):

$$\Delta_{\Sigma} \approx \Delta_0 \sqrt{\Delta t}, \quad (1.21)$$

где  $\Delta_{\Sigma}$  - ошибка накопления,  $\Delta_0$  - ошибка за один кадр,  $\Delta t$  - разность временных отсчетов между первым и последним кадром.

В случае использования накопления эталона хорошей оценкой накопления ошибки будет следующая:

$$\Delta_{\Sigma} \approx \Delta_0 \sqrt{\Delta t / t_e}, \quad (1.22)$$

где  $t_e$  - время накопления эталона. Это время ограничивается двумя факторами: оно не может быть больше времени полной смены сцены (времени, за которое начальный кадр полностью покидает поле зрения камеры за счет ее движения),

и времени изменения сцены (изменение сцены может быть обусловлено изменением освещения, точки зрения на сцену и т.д.).

Помимо модельных кадров для определения точности предложенного алгоритма использованы кадры с измеренным альтернативным способом сдвигом.

Был проведен отдельный физический эксперимент по оценке сдвига кадров видеопоследовательности. Для этого использовались две одинаковые камеры, одна из которых была выставлена на десятикратное увеличение. По камере с большим увеличением определялись сдвиги кадров с точностью до пикселя путем их совмещения. При этом на камере с меньшим увеличением этот сдвиг составлял доли пикселя. Таким образом, точность экспериментальной оценки составила 0.1 пиксель.

На экспериментальной видеопоследовательности предложенный алгоритм показал результаты в пределах погрешности эксперимента. Большая точность вряд ли имеет смысл для реальных кадров, т.к. сдвиги пикселей на них за счет искажений в атмосфере, aberrаций в оптике и т.д. лишает смысла более точные оценки.

В качестве обоснования высокой эффективности и качества работы алгоритма проведено их сравнение с другими аналогичными методами.

При этом сравнение велось как с теоретически описанными алгоритмами, так и с коммерческими программными продуктами (см. Табл. 1.6). Результаты исследований отнюдь не претендуют на полноту и не могут служить однозначным критерием качества того или иного метода. Это связано с большим количеством факторов – приспособленностью разных алгоритмов к разным ситуациям, труднодоступностью и высокой стоимостью коммерческих продуктов и т.д. Однако по представленным данным можно судить о среднестатистических параметрах существующих алгоритмов<sup>2</sup>.

---

<sup>2</sup> Данные приведены на май 2005 года – когда было проведено данное исследование.

Здесь скорость работы и вероятность сбоя измерялись по алгоритмам, определение угла поворота и коэффициента масштабирования в которых отключено. Под скоростью работы подразумевалась совокупная скорость работы алгоритмов определения параметров трансформации и генерации кадров.

**Табл. 1.6 Сравнение различных алгоритмов оценки геометрических искажений.**

Название алгоритма	С.К.О. определения сдвига (пиксели)	Максимальный сдвиг (пиксели)	С.К.О. определения угла (градусы)	Максимальный угол (градусы)	Вероятность сбоя	Количество кадров в секунду
Описанный выше алгоритм.	0.08	120	0.036	10	$\Delta 0.1\%$	233
DynaPel SteadyHands DV 2,2,0,2 <sup>3</sup> .	$\Delta 2$	$\sim 30$	$\Delta 0.2$	10	100%	14
DeShaker 1.6 <sup>4</sup> .	0.08	90	0.02	10	0.3%	6
Video Stabilizer 2.6.0.0 <sup>5</sup> .	$\sim 3$	10	$\emptyset$	$\emptyset$	100%	40
Спектральный метод компенсации смещений [17].	0.05	7	$\emptyset$	$\emptyset$	100%	21
Projection-Based Image Registration in the Presence of Fixed-Pattern Noise [18].	0.06	80	$\emptyset$	$\emptyset$	1.3%	97
Полупиксельный пирамидальный алгоритм [12].	0.03	45	0.015	30	2%	44
Вейвлет метод [19].	0.04	30	$\emptyset$	$\emptyset$	10%	0.12

<sup>3</sup> <http://steadyhand.dynapel.com>

<sup>4</sup> <http://home.bip.net/gunnart/video/deshaker.htm>

<sup>5</sup> <http://members.aol.com/HuberMM/index.html>

В некоторых случаях не удалось определить точные значения измеряемых величин. Для коммерческих продуктов это трудно сделать, поскольку они не предоставляют достаточно информации. Необходимые параметры удалось оценить лишь приближенно. Некоторые алгоритмы не определяют углы поворота.

Как видно – предложенные алгоритмы работают быстрее других исследованных алгоритмов и имеют преимущество по надежности работы и максимальному определяемому сдвигу.

Точность работы большинства алгоритмов в целом очень близки.

### **1.3 Учет особых явлений на изображении**

В данном разделе описаны некоторые способы учета отклонений параметров входных изображений от принятой модели. Учтено два явления:

1. «Тяжелые хвосты» распределений яркости в пикселе.
2. Малые сдвиги частей изображения друг относительно друга.

#### **1.3.1 Учет тяжелых хвостов распределений**

На изображении часто появляются движущиеся объекты. Они сильно выпадают из общей схемы преобразования кадра.

Приближенно их можно описать через так называемые «тяжелые хвосты» распределений шума в пикселях. Здесь вероятность крупного выброса шума гораздо больше, чем предполагается согласно нормальному распределению – но всё же достаточно мала.

Идея метода учета таких распределений дана в [42]. Однако описанный там метод достаточно трудоемок. Здесь же предлагается более простой и быстрый способ – ввести насыщение в подсчет функции невязки, т.е. вместо (1.5) вычислять:

$$x = \frac{1}{S(M)} \sum_{(i,j) \in M} (F(t-1)_{ij} - F(t)_{i+dx, j+dy})^2, \quad (1.23)$$

$$X(F(t-1), T_Q(F(t))) = \begin{cases} x, & x < C \\ C, & x \geq C \end{cases}$$

здесь  $C$  - некий порог.

Такой простой способ позволяет существенно увеличить надежность работы алгоритма определения геометрических трансформаций для изображений, на которых присутствуют движущиеся объекты (позволяет достичь характеристик, указанных в разделе 1.2.7, если до 20% кадра занято движущимися объектами).

### 1.3.2 Учет соседних пикселей

Для более качественного определения сдвига необходимо учитывать значения яркости в соседних пикселях. Это необходимо, прежде всего, для учета межпиксельной интерполяции. При сдвиге изображения на дробное число пикселей яркость в каждой дискретной точке изменяется. Приближенно ее можно описать как интерполяцию яркости соседних точек. Таким образом, необходимо учитывать яркость соседних пикселей.

Это также важно для учета геометрических искажений в камере (фактически, положение пикселя на изображении определяется не углом относительно оптической оси, но тангенсом этого угла, а, следовательно, нелинейно изменяется при перенацеливании камеры), и в некоторых ситуациях – например, при съемке поверхности воды или листвы, когда части изображения совершают небольшие колебания.

Хорошее описание способа такого учета можно найти в [43]. В данной работе приводится упрощенный, но вычислительно очень эффективный алгоритм.

Зададимся параметром  $d$  - эффективным радиусом учета соседних пикселей (см. Рис. 1.12). Предварительно необходимо вычислить следующие кадры:

сначала целочисленные кадры минимумов и максимумов

$$F'_{\min}(ij) = \min_{|di| < \lceil d \rceil, |dj| < \lceil d \rceil} F_{i+di, j+dj},$$

$$F'_{\max}(ij) = \max_{|di| < \lceil d \rceil, |dj| < \lceil d \rceil} F_{i+di, j+dj}, \quad (1.24)$$

а затем интерполированные

$$F_{\min}(ij) = F'_{\min}(ij) \frac{d}{\lceil d \rceil} + F_{ij} \left(1 - \frac{d}{\lceil d \rceil}\right),$$

$$F_{\max}(ij) = F'_{\max}(ij) \frac{d}{\lceil d \rceil} + F_{ij} \left(1 - \frac{d}{\lceil d \rceil}\right) \quad (1.25)$$

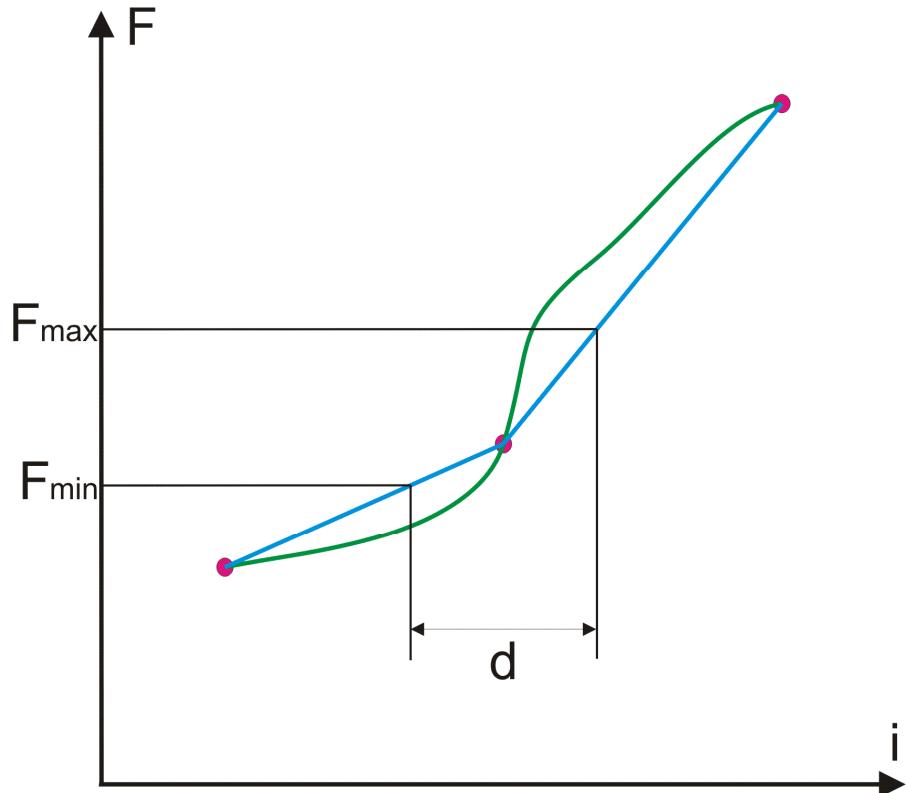


Рис. 1.12. Влияние соседних пикселей за счет межпиксельной интерполяции.

После этого рассчитывается эффективная невязка (вместо (1.5)) с учетом вычисленных значений минимумов и максимумов:

$$X(F(t-1), T_Q(F(t))) = \frac{1}{S(M)} \sum_{(i,j) \in M(dx,dy)} \Delta_e^2$$

$$\Delta_e = \begin{cases} (F(t-1)_{ij} - F_{\min}(t)_{i+dx, j+dy}), F(t-1)_{ij} < F_{\min}(t)_{i+dx, j+dy}, \\ (F(t-1)_{ij} - F_{\max}(t)_{i+dx, j+dy}), F(t-1)_{ij} > F_{\max}(t)_{i+dx, j+dy}, \\ 0, F_{\min}(t)_{i+dx, j+dy} \leq F(t-1)_{ij} \leq F_{\max}(t)_{i+dx, j+dy} \end{cases}, \quad (1.26)$$

Такой способ дает заметную прибавку к точности вычисления сдвига отдельного окна – а, следовательно, и полного вектора параметров трансформации всего кадра для случаев широких полей зрения (когда наблюдается описанная выше нелинейность).

#### 1.4 Теоретическое исследование характеристик алгоритма

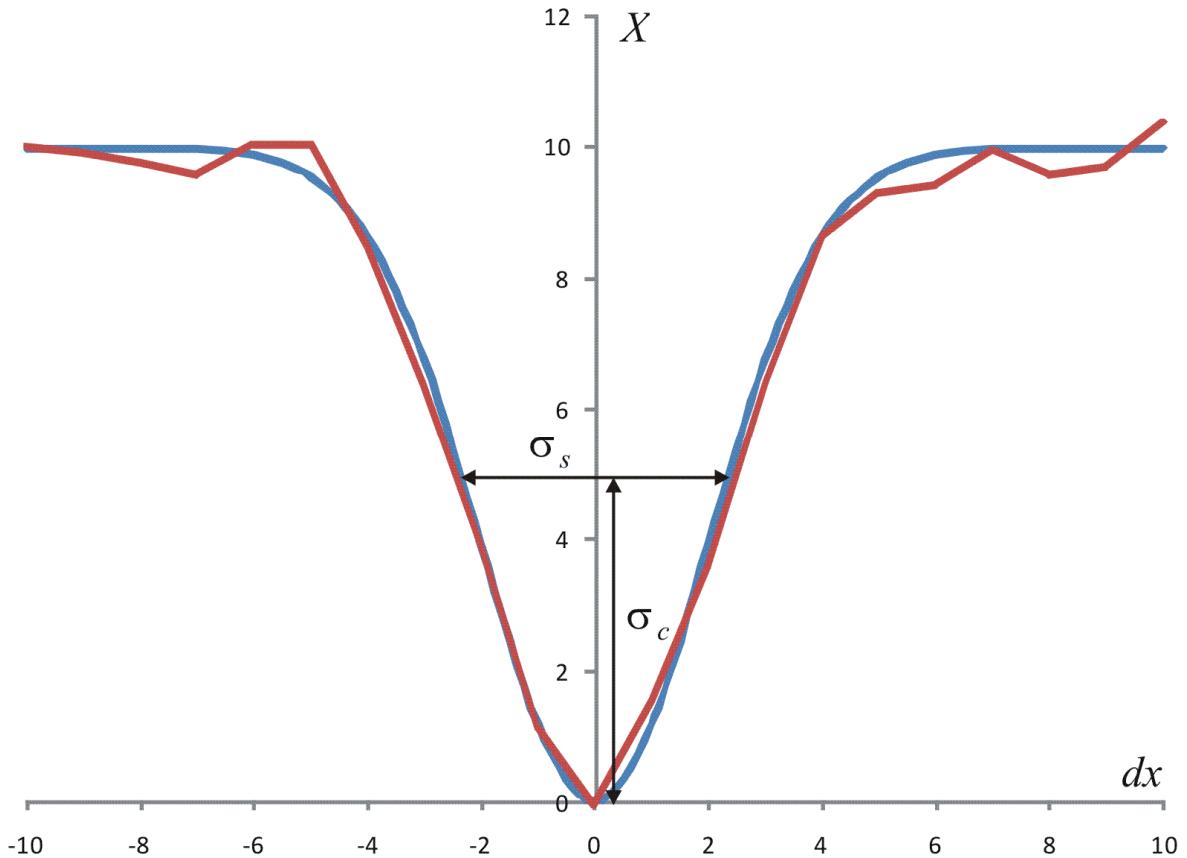
Большинство работ по данной тематике предлагают те или иные способы решения задачи. Они заканчиваются на разборе результатов – т.е. оценки точности, трудоемкости и т.д. представленных алгоритмов на основе тестовых случаев. Однако очень редко удается найти теоретическую оценку точности алгоритмов (некоторые сведения можно найти в [11]). Такая оценка весьма полезна, т.к. позволяет:

1. Оценить точность алгоритмов для конкретной – может быть весьма специфической - ситуации без их реализации.
2. Позволяет синтезировать алгоритмы предварительной обработки изображений, которые минимизируют итоговую ошибку.

Выпишем функцию невязки кадра с самим собой:

$$X = \frac{1}{S(M)} \left( \sum_{(i,j) \in M(dx,dy)} F_{ij}^2 + \sum_{(i,j) \in M(dx,dy)} F_{i+dx, j+dy}^2 - 2 \sum_{(i,j) \in M(dx,dy)} F_{ij} F_{i+dx, j+dy} \right), \quad (1.27)$$

Первые два члена здесь выражают среднюю мощность и изменяются с  $(dx,dy)$  слабо, и случайным образом. Третий член является определяющим (связь между функцией невязки и функцией корреляции показана в [42]).



**Рис. 1.13. Типичный пример функции невязки. Красным обозначена реальная невязка, синим – её гауссовское приближение.**

Хорошим приближением функции автокорреляции является функция Гаусса (постоянное слагаемое опущено для простоты расчетов без ущерба для точности)

$$C(dx, dy) = -Ane^{-\frac{dx^2 + dy^2}{2a^2}}, \quad (1.28)$$

где  $a$  и  $A$  – параметры изображения.

Положим, что шумы в пикселях распределены по нормальному закону и независимы. Тогда величина автокорреляционной функции распределена по закону  $\chi^2$  с дисперсией  $\sigma_c$

$$\sigma'_c = \sqrt{2\sigma_p^2 n}, \quad (1.29)$$

где  $\sigma_p$  – уровень шума в пикселе, а  $n$  – количество точек в окне суммирования.

В действительности сдвиг считается не по автокорреляционной функции, а по функции взаимной корреляции. Т.к. пиксели изображения дискретны, а сдвиг может быть дробным, необходимо учитывать шум интерполяции. Тогда

$$\sigma_c = \sqrt{2\sigma_p^2 n + (0.5\bar{F}')^2}, \quad (1.30)$$

где  $\bar{F}'$  - среднее значение модуля дискретной производной (т.е. разности яркости между соседними пикселями).

Тогда ошибка определения сдвига  $\sigma_s$  будет (см. Рис. 1.13)

$$\sigma_s = \sqrt{2a^2 \ln \frac{An}{An - \sigma_c}} \approx a \sqrt{\frac{2\sigma_c}{An}} \approx 2^{3/4} a \sqrt{\frac{\sigma_p}{A\sqrt{n}}} \quad (1.31)$$

(последнее приближенное равенство получено без учета шума интерполяции).

Таким образом, ошибка определения сдвига приблизительно пропорциональна характерному размеру изображения  $a$ , корню из уровня шума в пикселе и обратно пропорциональна корню четвертой степени из количества пикселей в окне суммирования.

Пусть к каждому кадру применяется предварительная обработка – свертка с функцией Гаусса:

$$G(dx, dy) = \frac{1}{2\pi\sigma_\Gamma^2} e^{-\frac{dx^2 + dy^2}{2\sigma_\Gamma^2}}. \quad (1.32)$$

Найдем преобразование Фурье от функций невязки и функции Гаусса:

$$\begin{aligned} \tilde{C}(\omega_x, \omega_y) &= -An^2 a^2 e^{-\frac{a^2}{2}(\sigma_x^2 + \sigma_y^2)}, \\ \tilde{G}(\omega_x, \omega_y) &= -\frac{\sigma_\Gamma^2}{2\pi} e^{-\frac{\sigma_\Gamma^2}{2}(\sigma_x^2 + \sigma_y^2)}. \end{aligned} \quad (1.33)$$

Т.к. функция автокорреляции является сверткой изображения с самим собой, то, воспользовавшись известным свойством преобразования Фурье, получаем выражение для спектра автокорреляционной функции для изображения, размытого с Гассовским ядром:

$$\tilde{C}_\Gamma(\omega_x, \omega_y) = \tilde{C}(\omega_x, \omega_y) \tilde{G}(\omega_x, \omega_y) \tilde{G}(\omega_x, \omega_y). \quad (1.34)$$

Взяв обратное преобразование Фурье находим:

$$C_\Gamma(dx, dy) = -\frac{A\alpha a^2}{2\pi\sigma_\Gamma^2(a^2 + 2\sigma_\Gamma^2)} e^{-\frac{dx^2 + dy^2}{2(a^2 + 2\sigma_\Gamma^2)}}. \quad (1.35)$$

Найдем теперь дисперсию этой автокорреляционной функции. Её распределение не подчиняется закону  $\chi^2$ , т.к. после свертки с функцией Гаусса значения в разных пикселях стали зависимыми. Точное распределение, выраженное в виде ряда по куммулянтам, можно найти в [50]. Такое представление весьма громоздко и неудобно. Воспользуемся приближением из [51].

Согласно этой работе, распределение суммы зависимых гауссовских величин будет гамма-распределением вида

$$f_{C_\Gamma}(s) = \frac{1}{\Gamma(\nu)} \alpha^\nu s^{\nu-1} e^{-\alpha s}. \quad (1.36)$$

с параметрами

$$\alpha = \frac{nR(0)}{2nR^2(0) + 4\sum_{i=1}^{n-1} (n-i)R^2(i)},$$

$$\nu = \frac{n^2 R^2(0)}{2nR^2(0) + 4\sum_{i=1}^{n-1} (n-i)R^2(i)}, \quad (1.37)$$

где  $R(i)$  - корреляционная функция. Нетрудно показать, что для нашего случая

$$R(dx, dy) = \sigma_p^2 e^{-\frac{dx^2 + dy^2}{4\sigma_\Gamma^2}}. \quad (1.38)$$

Заменив сумму в знаменателе на интеграл и воспользовавшись свойствами гамма-распределения, найдём дисперсию автокорреляционной функции, обусловленную собственным шумом фотоприемника:

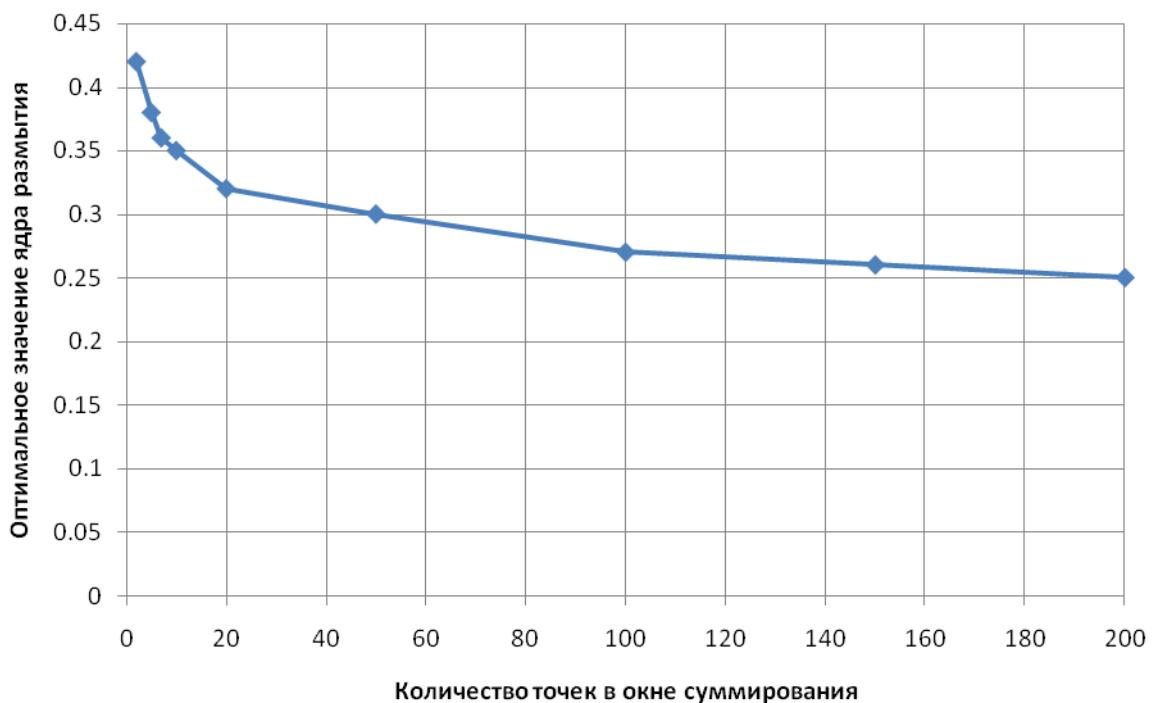
$$\sigma'_{c\Gamma} = \sigma_p \sqrt{2n + 8n\pi\sigma_\Gamma^2 - \sqrt{2}\sigma_\Gamma^3\pi^{3/2}}, \quad (1.39)$$

Дискретная производная при свертке с Гауссом изменяется очевидным образом

$$\bar{F}'_\Gamma = F_\Gamma(x) - F_\Gamma(x-1) = \bar{F}'(G(0) - G(1)) = \bar{F}' \frac{1 - e^{-1/2\sigma_\Gamma^2}}{2\pi\sigma_\Gamma^2} \approx \bar{F}' / 4\pi\sigma_\Gamma^4, \quad (1.40)$$

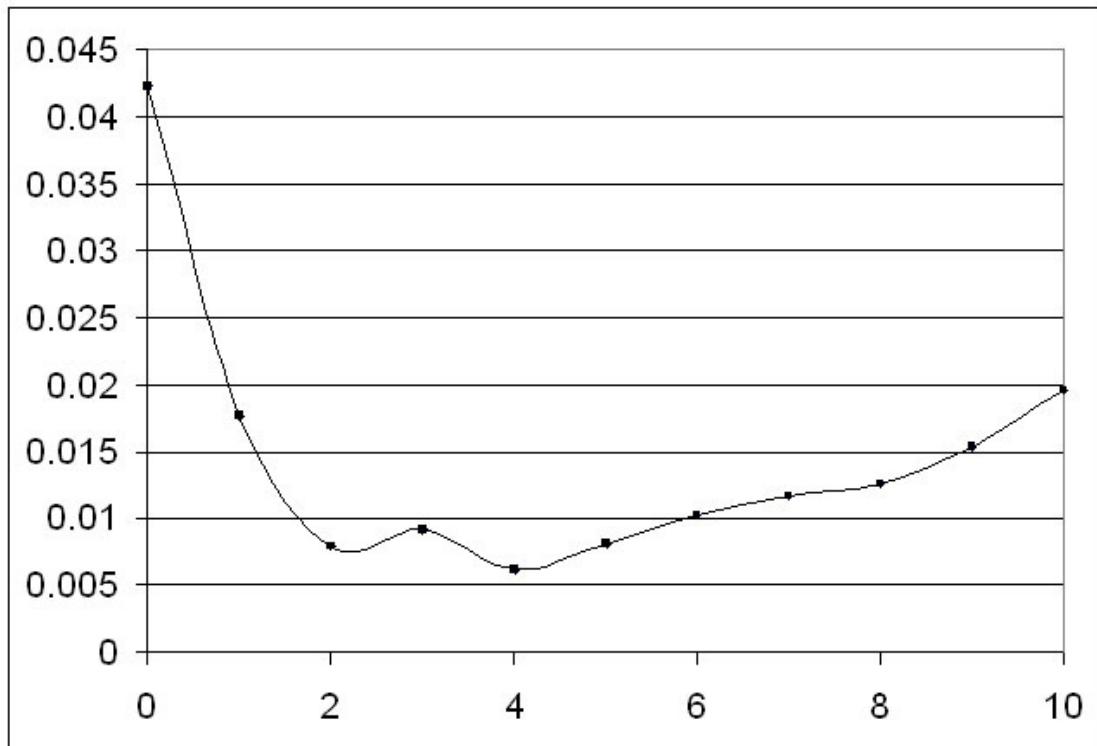
Тогда ошибка определения сдвига будет:

$$\sigma_{s\Gamma} \approx \sqrt{\frac{4\pi\sigma_\Gamma^2(a^2 + 2\sigma_\Gamma^2)^2 \sqrt{\sigma'_{c\Gamma}^2 + (0.5\bar{F}'/4\pi\sigma_\Gamma^4)^2}}{Ana^2}} \quad (1.41)$$



**Рис. 1.14. Зависимость оптимального ядра размытия  $\sigma_\Gamma$  от количества точек  $n$ .**

Данная зависимость при определенных параметрах имеет минимум. Целесообразно производить свертку с функцией Гаусса для улучшения точности определения сдвига. Увеличение точности связано с уменьшением ошибки интерполяции, в то время как ошибка за счет собственных шумов только возрастает. На Рис. 1.14 приведена зависимость оптимального ядра размытия от количества точек для  $a = 2$  и  $A = 10$ .



**Рис. 1.15. Зависимость качества определения сдвига (пиксели) изображения от параметра  $\sigma_G$  (пиксели).**

На Рис. 1.15 представлена экспериментальная зависимость качества определения сдвига изображения от параметра  $\sigma_G$ .

### 1.5 Выводы

1. Создан и реализован алгоритм реального времени оценки геометрических преобразований кадров видеопоследовательности. Он основан на объединении трёх основных существующих подходов. Ключевым моментом в предложенном алгоритме является определение сдвига с оценкой доверительной области.
2. Проведено исследование предложенного алгоритма на основе большего количества синтетических и реальных видеопоследовательностей. Показана высокая точность, надежность и скорость алгоритма. Сравнение с альтернативными методами, как описанными в литературе, так и реализованными в виде

коммерческих продуктов, показало высокие качества предложенного алгоритма.

3. Проведена теоретическая оценка точности алгоритма определения сдвига. Найдены соотношения, позволяющие для произвольной видеопоследовательности найти параметры преобразований, минимизирующие ошибку определения сдвига.

## 2 Стабилизация и построение панорамных кадров

### 2.1 Введение

Алгоритм определения геометрических преобразований используется для решения двух важных практических задач – стабилизации изображения и построения панорамных кадров.

Для их решения, помимо определения геометрических преобразований, необходимо решить еще несколько проблем.

Во-первых, оцененные геометрические трансформации необходимо разделить на движение за счет управления камерой оператором и шумовые колебания. Эта задача не может быть определена формально, но некоторых результатов, значительно улучшающих восприятие выходной видеопоследовательности, добиться можно.

Во-вторых, на вход алгоритма определения геометрических трансформаций должны поступать кадры, которые можно с той или иной степенью точности считать геометрической трансформацией некоторого начального изображения. Если это условие не выполнено, то алгоритм выдаст в качестве параметров геометрических преобразований некоторые случайные величины. Во избежание подобных явлений необходимо перед входом алгоритма поставить фильтр, селектирующий ситуации, которые выходят за рамки математической модели, принятой при создании алгоритма. Такими могут быть смена сцены, шумовые кадры и некоторые другие.

Алгоритм стабилизации описан в разделе 2.2. Необходимые для него алгоритмы разделения геометрических преобразований на управляемые и шумовые и определения качества видеопоследовательности описаны, соответственно, в разделах 2.3 и 2.4. Раздел 2.5 посвящен задаче построения панорамных кадров. В разделе 2.6 содержатся выводы по второй главе.

## 2.2 Стабилизация

Стабилизация изображения производится путем применения к кадрам преобразования, обратного к найденному.

В первой главе описан алгоритм определения геометрических преобразований кадров. Для адекватного отображения видеопоследовательности необходимо компенсировать только шумовую составляющую его геометрических искажений. Регулярную составляющую (например, поступательное смещение камеры) следует, по возможности, оставить с минимальными изменениями.

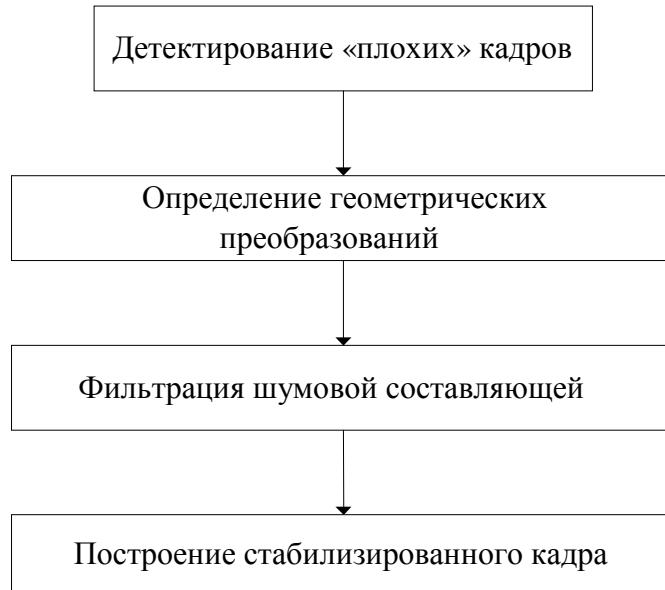
Теоретически, существует возможность определить управляемую составляющую непосредственно из команд оператора. На практике это трудноосуществимо, т.к. нужны устройства, контролируемые с большой точностью по направлению (порядка разрешающей способности камеры) и синхронизованные по времени (с точностью порядка времени кадра). Более того, управление камерой может осуществляться другим приложением и даже не компьютерным электронным устройством, а, значит, передача необходимой информации от модуля управления к модулю стабилизации превращается в нетривиальную инженерную задачу.

Как было сказано, на вход алгоритма определения геометрических трансформаций должны поступать кадры, которые можно с той или иной степенью точности считать геометрической трансформацией некоторого начального изображения. Если это условие не выполнено, то алгоритм выдаст в качестве параметров геометрических преобразований некоторые случайные величины. Стабилизация с такими параметрами даст на выходе изображения, «скачущее» больше, чем входное.

Для решения задачи применяются два метода: введение регуляризации (описанное в первой главе), и пропускание обработки низкокачественных кадров (при этом удобнее всего отображать их с нулевыми преобразованиями).

Ситуациями, в которых алгоритм определения преобразований дает сбой, являются смена сцены, шумовые кадры и некоторые другие. Также необходимо детектировать такие ситуации, как наличие чересстрочной развертки.

Структурная схема алгоритма стабилизации будет выглядеть так, как указано на Рис. 2.1.



**Рис. 2.1 Блок-схема алгоритма стабилизации.**



**Рис. 2.2. Пример работы алгоритма стабилизации. Слева – исходный кадр, в центре – стабилизированный, справа – стабилизированный с построением панорамного кадра.**

Для более качественного отображения целесообразно строить выходной стабилизированный кадр не из входного, а из панорамного кадра. Это позволяет снизить на выходном кадре пространство, на котором нет входной информации (см Рис. 2.2).

### 2.3 Разделение геометрических преобразований на управляемые и шумовые

Геометрические преобразования представляют собой следующую функцию времени (здесь исследуются только аффинные преобразования без учета скоса):

$$\begin{cases} \Delta \vec{r}(t) = \Delta \vec{r}_s(t) + \Delta \vec{r}_n(t) \\ A(t) = A_s(t) + A_n(t) \end{cases}, \quad (2.1)$$

$\Delta \vec{r}_s(t)$  и  $A_s(t)$  - полезные составляющие геометрических трансформаций  $A(t) = (a_0, a_1, a_2, a_3)$  и  $\Delta \vec{r}(t) = (a_4, a_5)$  (см. (1.1)),  $\Delta \vec{r}_n(t)$  и  $A_n(t)$  - шумовые составляющие, которые и надо компенсировать.

Задача не может быть решена без априорных сведений о полезных составляющих искажений. Но они, как правило, скудны и не позволяют полностью и точно решить задачу. Поэтому в данной работе рассмотрены эмпирические способы решения, которые охватывают как можно большее число возможных ситуаций, а также могут быть настроены в соответствии с конкретной задачей.

Таким образом, решается следующую задачу:

$$\begin{cases} \Delta \vec{r}_n(t) = F_r(\Delta \vec{r}(t), \Delta \vec{r}(t-1), \Delta \vec{r}(t-2)...) \\ A_n(t) = F_A(A(t), A(t-1), A(t-2)...) \end{cases}, \quad (2.2)$$

где  $F_r$  и  $F_A$  – некоторые функции, которые должны быть подобраны такими, чтобы наиболее адекватно отделять шумовые составляющие геометрических преобразований от не шумовых.

Для решения задачи удобно сделать замену переменных:

- представлять матрицу  $A$  как комбинацию двух параметров – угла поворота  $\varphi$  и коэффициента масштабирования  $k$ :

$$A = \begin{vmatrix} a & b \\ -b & a \end{vmatrix} \Leftrightarrow \begin{cases} \varphi = \operatorname{arctg} \left( \frac{b}{a} \right) \\ k = \sqrt{a^2 + b^2} \end{cases}, \quad (2.3)$$

- перейти от относительных преобразований двух последовательных кадров к абсолютным преобразованиям  $A_a(t)$  и  $\Delta\vec{r}_a(t)$  последнего кадра относительно первого (опорного):

$$A_a(t) = \prod_{\tau=t}^o A(\tau) \quad (2.4)$$

$$\Delta\vec{r}_a(t) = \Delta\vec{r}(t) + \sum_{k=t-1}^0 \left( \prod_{\tau=t}^{k+1} A(\tau) \right) \cdot \Delta\vec{r}(k)$$

Шумовая и полезная составляющие геометрических искажений вычисляются по аналогичным формулам.

Задача принципиально не формализуема, т.к. данные об управлении камерой в подавляющем большинстве недоступны либо недостаточно точны. Поэтому необходимо искать не точное решение, а приближенное. Здесь возможны четыре подхода:

1. Задаться некоторыми априорными сведениями относительно полезных составляющих  $\Delta\vec{r}_s(t)$  и  $A_s(t)$  искажений. Например, аппроксимировать их какой-либо параметрической функцией.
2. Отфильтровать значения геометрических преобразований низкочастотным фильтром.
3. Задаться некоторой эмпирической формой функций  $F_r$  и  $F_A$  с некоторым числом параметров, и подобрать эти параметры для наиболее адекватного решения задачи.
4. Задаться критерием качества, который может включать в себя разнообразные факторы: например, скорость смещения результирующего изображения, степень несовпадения его с оригиналом и т.д. Тогда искомые параметры находятся путем минимизации этого критерия.

Перечисленные подходы описаны в следующих трех разделах.

### 2.3.1 Аппроксимация движения параметрическими функциями и низкочастотная фильтрация

Задачу выбора параметров отображения (т.е. шумовых составляющих искажений  $\Delta \vec{r}_n(t)$  и  $A_n(t)$ , которые надо компенсировать) можно решать путем аппроксимации параметров  $\Delta \vec{r}_s(t)$  и  $A_s(t)$  какими-либо параметрическими функциями  $f_A(t, r_1, r_2 \dots)$  и  $\vec{f}_r(t, r_1, r_2 \dots)$ . Тогда, (с учетом (2.4)):

$$\begin{aligned} F_r(\Delta \vec{r}_a(t), \Delta \vec{r}_a(t-1), \Delta \vec{r}_a(t-2) \dots) = \\ \Delta \vec{r}_a(t) - \hat{\Delta \vec{r}}_s(\Delta \vec{r}_a(t), \Delta \vec{r}_a(t-1), \Delta \vec{r}_a(t-2) \dots) \\ F_A(A_a(t), A_a(t-1), A_a(t-2) \dots) = \\ A_a(t) - \hat{A}_s(A_a(t), A_a(t-1), A_a(t-2) \dots) \end{aligned} \quad (2.5)$$

$\hat{\Delta \vec{r}}_s$  и  $\hat{A}_s$  - оценки полезных составляющих, вычисляемые по формулам:

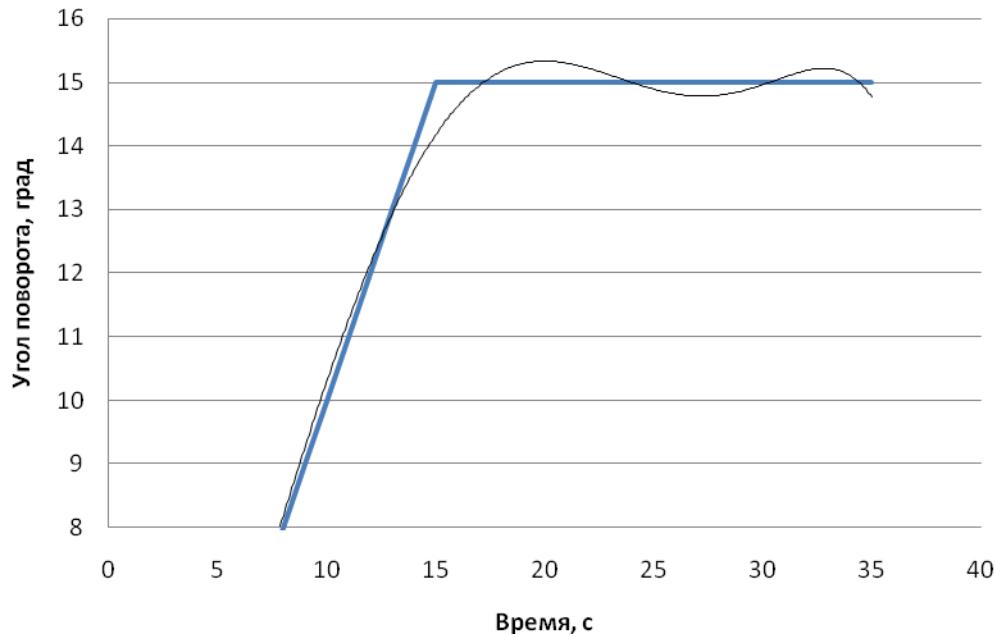
$$\begin{aligned} \hat{\Delta \vec{r}}_s(t) = \vec{f}_r(t, \hat{r}_1, \hat{r}_2 \dots) \\ \hat{A}_s(t) = f_A(t, \hat{a}_1, \hat{a}_2 \dots) \end{aligned} \quad (2.6)$$

$$\begin{aligned} (\hat{r}_1, \hat{r}_2 \dots) = \arg \min_{r_1, r_2 \dots} \sum_{\tau=0}^t (\vec{r}_a(\tau) - \vec{f}_r(\tau, r_1, r_2 \dots))^2 \\ (\hat{a}_1, \hat{a}_2 \dots) = \arg \min_{a_1, a_2 \dots} \sum_{\tau=0}^t (A_a(\tau) - f_A(\tau, a_1, a_2 \dots))^2 \end{aligned} \quad (2.7)$$

$\hat{r}_1, \hat{r}_2, \hat{a}_1, \hat{a}_2 \dots$  - оцениваемые параметры. Оценку этих параметров можно проводить либо методом наименьших квадратов, либо фильтром Калмана.

В качестве функций  $f_A(t, r_1, r_2 \dots)$  и  $\vec{f}_r(t, r_1, r_2 \dots)$  удобно брать полиномиальные функции небольших степеней. Это позволяет сгладить шум и достаточно точно описать поведение полезных составляющих геометрических искажений в простых случаях.

Такой способ решения данной задачи является далеко не лучшим. Это связано с разрывностью первой производной управляемого движения – оператор часто резко изменяет направление движения камеры. В этих случаях фильтрация дает нежелательные колебания вблизи точки разрыва – см. Рис. 2.3.



**Рис. 2.3. Пример полиномиальной фильтрации управляемого движения. Синяя линия – движение камеры, черная – результат фильтрации полиномом 6-го порядка. Видно сильные мешающие колебания.**

Низкочастотная фильтрация (полиномиальное сглаживание также является низкочастотной фильтрацией) имеет тот же недостаток – колебания вблизи резких скачков.

### 2.3.2 Эмпирический метод

Функции  $F_r$  и  $F_A$  можно выбрать эмпирически, а потом найти такие параметры, которые будут описывать ситуацию наиболее адекватно.

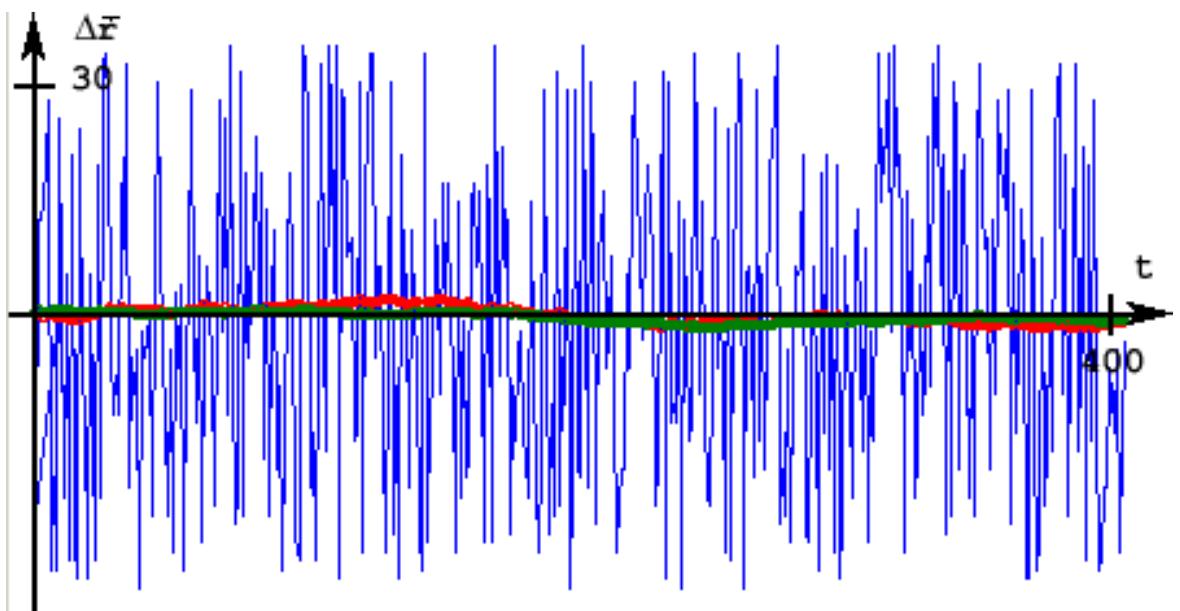
В данной работе рассматривалось несколько таких функций:

*Простейшая линейная функция:*

$$\begin{cases} \Delta \vec{r}_{na} = \Delta \vec{r}_a \cdot \left(1 - \frac{1}{N_r}\right) \\ k_{na} = 1 \\ \varphi_{na} = \varphi_a \cdot \left(1 - \frac{1}{N_\varphi}\right) \end{cases}, \quad (2.8)$$

$k_{na}$  и  $\varphi_{na}$  - представление шумовой матрицы  $A_{na}$  в виде (2.3),  $\varphi_a$  - компонент матрица  $A_a$ ,  $N_r$  и  $N_\varphi$  - параметры, выбираемые экспериментально (порядка 20 – 200).

Шумовая компонента коэффициента масштабирования здесь принята за единицу, так как крайне редко можно встретить случай, когда масштаб изображения колеблется случайным образом.



**Рис. 2.4. Пример работы функций выбора параметров при колебании изображения вокруг некоторого положения. Синим цветом обозначена функция  $\Delta \vec{r}$ , зеленым -  $\Delta \vec{r}_s$ , вычисленная первым способом, красным – вторым. Видна высокая степень подавления шумовой компоненты.**

Данный подход позволяет сгладить шумовые колебания, в то время как медленное поступательное и вращательное движения камеры будет отслеживаться (Рис. 2.4). Однако, если движение будет достаточно быстрым, то в конечном итоге входное изображение может полностью уйти из выходного

кадра, т.е. сместиться на столько, что не будет видно на выходном кадре (Рис. 2.5).

Экспоненциальная функция:

$$\begin{cases} \Delta \vec{r}_{na} = \Delta \vec{r}_a \cdot \exp\left(-\frac{1}{N_{r1}} - \frac{|\Delta \vec{r}|}{N_{r2}}\right) \\ k_{na} = 1 \\ \varphi_{na} = \varphi_a \cdot \left(1 - \frac{1}{N_\varphi}\right) \end{cases}, \quad (2.9)$$

$N_{r1}$  и  $N_{r2}$  – дополнительные настроочные параметры (в данном случае  $N_{r1}$  – 20-200,  $N_{r2} = 0.05N_{r1}$ ).

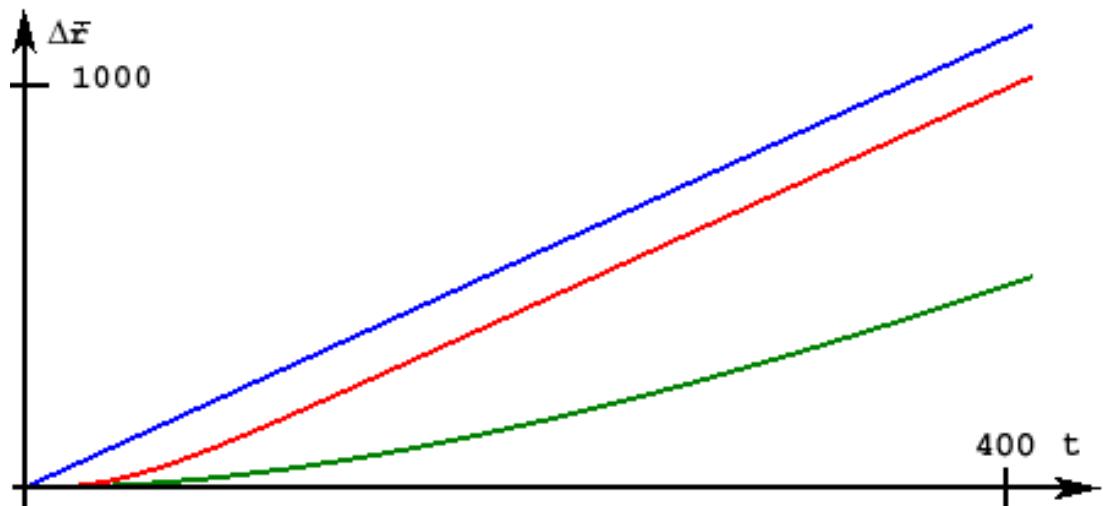


Рис. 2.5. Пример работы функций выбора параметров при поступательном смещении изображения. Обозначения аналогичны Рис. 2.4. Видно, что первая функция не вполне отслеживает поступательное движение изображения.

Данный подход при небольших сдвигах работает аналогично предыдущему (Рис. 2.4), но при больших – обеспечивает незначительное «отставание» выходного изображения от входного (Рис. 2.5).

### 2.3.3 Минимизация критерия качества

Другим методом оценки функций  $F_r$  и  $F_A$  является минимизация критерия качества. Его удобнее записывать не через шумовые составляющие, а через полезные. Тогда:

$$\begin{cases} A_{na}(t) = A_a(t)\hat{A}_{sa}^{-1}(t) \\ \Delta\vec{r}_{na}(t) = \Delta\vec{r}_a(t) - A_n(t) \cdot \Delta\hat{\vec{r}}_{sa}(t) \\ (\hat{A}_{sa}(t), \Delta\hat{\vec{r}}_{sa}(t)) = \\ \arg \min_{A_{sa}, \Delta\vec{r}_{sa}} J(A_a(t), A_a(t-1), \dots, A_{sa}(t), A_{sa}(t-1), \dots, \\ \Delta\vec{r}_a(t), \Delta\vec{r}_a(t-1), \dots, \Delta\vec{r}_{sa}(t), \Delta\vec{r}_{sa}(t-1), \dots) \end{cases}, \quad (2.10)$$

$\hat{A}_{sa}(t)$  и  $\Delta\hat{\vec{r}}_{sa}$  - оценки полезных составляющих геометрических искажений (относительно начального кадра),  $J(\dots)$  - критерий качества.

Данный критерий качества может включать в себя различные характеристики, которые должны быть минимальны для более качественного отображения выходного кадра. В данной работе использовался следующий критерий:

$$\begin{aligned} J(\dots) = & (\Delta\vec{r}_a(t) - \Delta\vec{r}_{sa}(t))^4 + \\ & \alpha(\Delta\vec{r}_{sa}(t) - \Delta\vec{r}_{sa}(t-1))^2 + \\ & \beta|\varphi_a(t) - \varphi_{sa}(t)|_\varphi^2 + \\ & \gamma|\varphi_{sa}(t) - \varphi_{sa}(t-1)|_\varphi^2 + \\ & \delta \cdot D^2(t, \Delta\vec{r}_{sa}(t), \varphi_{sa}(t)) \end{aligned}, \quad (2.11)$$

$|\dots|_\varphi$  - модуль разности углов, учитывающий их цикличность,  $D^2(t, \Delta\vec{r}_{sa}(t), \varphi_{sa}(t))$  - количество точек на выходном кадре, на которые не попала информация с входного кадра,  $\alpha, \beta, \gamma, \delta$  - весовые коэффициенты, которые удобно выбирать в следующих пределах:

$$\begin{cases} \alpha = 5 \cdot 10^4 \div 5 \cdot 10^5 \\ \beta = 0.5 \div 5 \\ \gamma = \beta \cdot (10 \div 100) \\ \delta = 10^5 \div 10^6 \end{cases}, \quad (2.12)$$

В данном критерии первое слагаемое  $(\Delta\vec{r}_a(t) - \Delta\vec{r}_{sa}(t))^4$  не позволяет выходному изображению слишком сильно отдаляться от входного, тем самым обеспечивая отслеживание быстро сдвигающегося изображения. Второе слагаемое  $\alpha(\Delta\vec{r}_{sa}(t) - \Delta\vec{r}_{sa}(t-1))^2$  не позволяет выходному изображению быстро колебаться, тем самым сглаживая шумовую составляющую сдвигов. Вместе эти два слагаемых определяют оптимальное сглаженное слежение за полезным сдвигом и подавление шумового сдвига.

Третье и четвертое слагаемые  $(\beta|\varphi_a(t) - \varphi_{sa}(t)|_\varphi^2)$  и  $(\gamma|\varphi_{sa}(t) - \varphi_{sa}(t-1)|_\varphi^2)$  осуществляют аналогичное поведение выходного изображения относительно угла поворота.

Последнее слагаемое  $\delta \cdot D^2(t, \Delta\vec{r}_{sa}(t), \varphi_{sa}(t))$  характеризует количество точек на выходном кадре, на которые не попала информация с входного кадра. Оно имеет смысл, только если выходной кадр строится не из одного входного кадра, а из всей последовательности. Например, если сначала из всей входной последовательности, с учетом оцененных сдвигов, строится общая мозаика большого размера, а затем из этой мозаики «вырезается» выходной кадр меньшего размера. При этом в выходной кадр могут попасть точки мозаики, в которых не была записана информация ни с одного входного кадра. Количество таких точек необходимо свести к минимуму.

Задача минимизации критерия качества является нелинейной трехмерной задачей минимизации. А из-за присутствия последнего члена она не может быть решена аналитически. Для ее решения необходимо применять методы нелинейной многомерной оптимизации, например, метод Пауэлла [52].

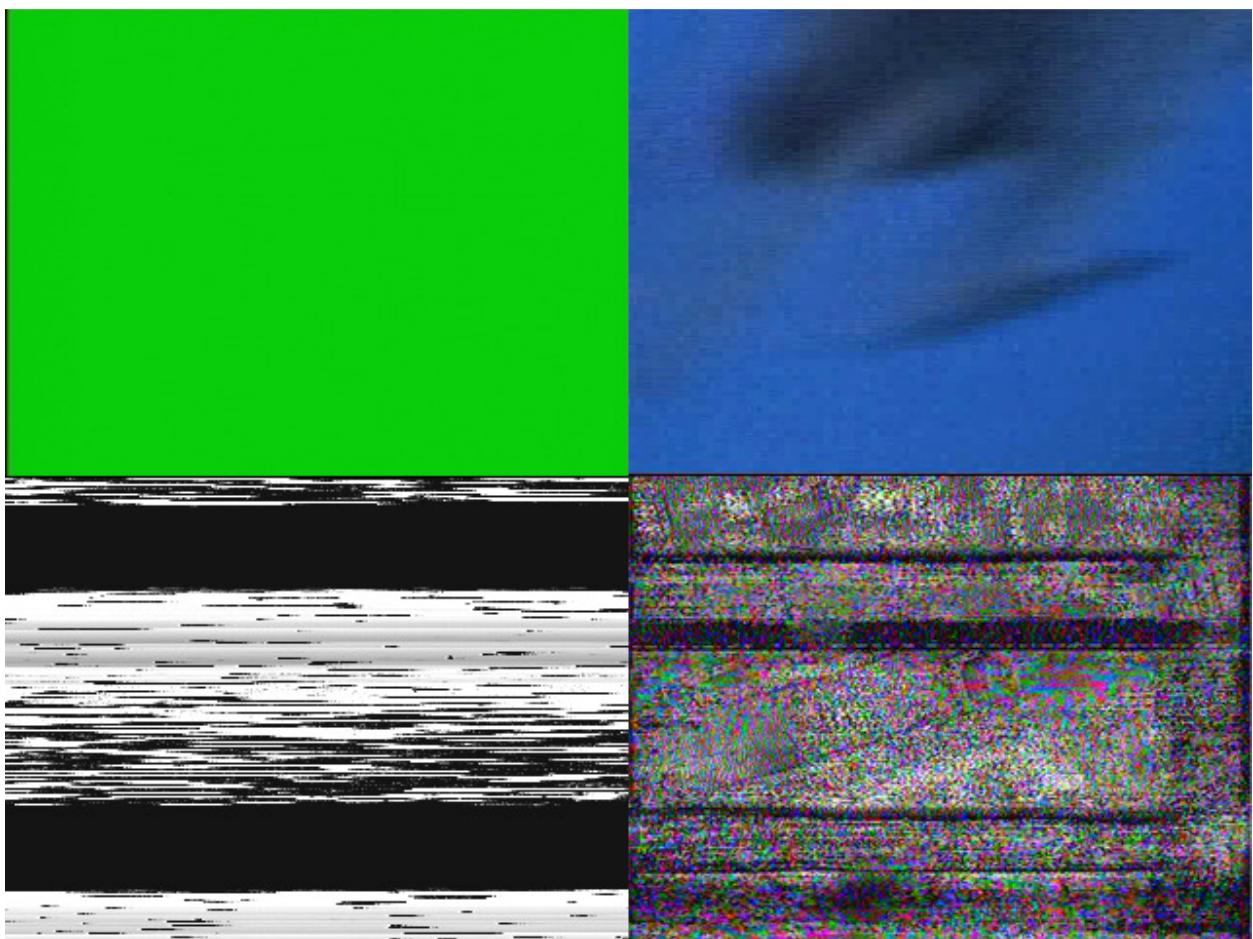
Метод выбора параметров отображения обладает следующими преимуществами:

- В критерий качества можно включать произвольное число параметров, которые позволяют синтезировать критерий произвольного качества и с произвольными свойствами. Причем каждое слагаемое имеет простой и понятный смысл, что позволяет синтезировать критерии с желаемыми свойствами.
- Все остальные описанные выше подходы могут быть представлены через минимизацию критерия качества путем выбора соответствующих слагаемых в (2.11).

## 2.4 Определения качества видеопоследовательности

Как было сказано выше, для качественной работы алгоритма определения геометрических искажений на его вход должны подаваться кадры, в той или иной степени соответствующие выбранной математической модели.

Суть проблемы в том, что алгоритм стабилизации оценивает параметры геометрических преобразований кадров путем нахождения минимума некоторой целевой функции. Таким образом, он не может работать при несоответствии входных изображений модели геометрических преобразований, принятых при разработке алгоритма, а также при слабой обусловленности задачи – когда целевая функция слабо зависит от параметров преобразований. Если алгоритм стабилизации определяет параметры геометрических преобразований неправильно, то выходное изображение, «стабилизированное» в соответствии с такими неправильными параметрами, «скачет» больше, чем входное. Это недопустимо, поэтому следует отсеивать кадры, обработка которых не может быть выполнена достаточно качественно.



**Рис. 2.6. Примеры кадров низкого качества.** Слева вверху – отсутствие полезного сигнала, справа вверху – съемка с движущейся камеры с большим увеличением, слева внизу – помехи в канале связи при отсутствии сигнала, справа внизу – помехи в канале связи на фоне полезного сигнала.

На практике такие ситуации возможны в следующих случаях (см. Рис. 2.6):

1. Смена сцены.
2. Отсутствие полезного сигнала (при наличии или отсутствии шума).
3. Высокий уровень помех.
4. Некоторые другие – например, съемка с большим увеличением при быстром перемещении камеры приводит к сильным «смазам» изображения и слишком большим смещениям.

Несмотря на изначальную нацеленность предложенного алгоритма на работу с алгоритмами стабилизации и сопровождения объектов, он может быть

использован и для более общих задач – определения смены сцены, шумовых кадров и т.д.

В данной области была проделана большая работа разными коллективами. Особенno много исследований коснулось проблемы поиска смены сцены как, пожалуй, самой востребованной задачи из вышенназванных. Большинство алгоритмов легко свести к простой схеме – вычисление некоторых метрик с последующей их классификацией. В простейшем случае – сравнение с порогом. Особую популярность получили алгоритмы, использующие сжатые алгоритмом MPEG-4 последовательности [37], [38], [39] и [40], где в качестве метрик взяты несколько первых коэффициентов разложения в ряд по косинусам. Для получения метрик могут быть также использованы пространственно-временные разложения [34] или разложения по вейвлетам [31]. Также существует ряд алгоритмов [33], которые используют очень сложные алгоритмы анализа сцены, включающие в себя сегментацию и сопровождение объектов, для нахождения смены сцены. В работах [32, 35, 36] представлены методы на основе простых метрик, таких как разность последовательных кадров, скорость изменения этой яркости и анализ гистограмм. Работа [31] особенно интересна тем, что смена сцены в ней не просто детектируется, но выносится решение о «силе» изменения сцены.

Помимо указанных выше случаев также существуют ситуации, в которых необходима оценка параметров входных изображений. Таких ситуаций можно выделить две:

1. Наличие чересстрочной развертки.
2. Наличие шумов особого характера.

Описание детектирование таких ситуаций приведено ниже.

#### **2.4.1 Алгоритм детектирования шумоподобных кадров и смены сцены для общего случая**

Согласно [20] двуальтернативная задача имеет следующее решение:

1-е решение принимается, если

$$\frac{p_1(\vec{x})}{p_2(\vec{x})} > \frac{p_2(g_{22} - g_{12})}{p_1(g_{11} - g_{21})}, \quad (2.13)$$

иначе – 2-е. Здесь  $\vec{x}$  – вектор наблюдения (минимальная достаточная статистика),  $p_i(\vec{x})$  – функция правдоподобия  $i$ -ой ситуации,  $p_i$  – априорные вероятности ситуаций,  $p_{ij}$  – функция потерь.

В реальности функции правдоподобия в различных ситуациях неизвестны, а размерность задачи очень велика (может доходить до миллионов). Поэтому необходимо найти способ снижения размерности, т.е. минимальную достаточную (или приблизительно достаточную) статистику. Такие достаточные статистики принято называть метриками.

На практике найти точную достаточную статистику низкой размерности невозможно, так что используются приближенная достаточная статистика. Функции правдоподобия, как правило, явным образом не выписываются. Для решения задачи используются алгоритмы классификации, которые приближенно выделяют в пространстве метрик область, соответствующую решению 1 или 2.

#### 2.4.1.1 Вычисление метрик

Вычисление метрик основывается на принципах, изложенных в разделе 1.2.

В качестве метрик были выбраны три величины:

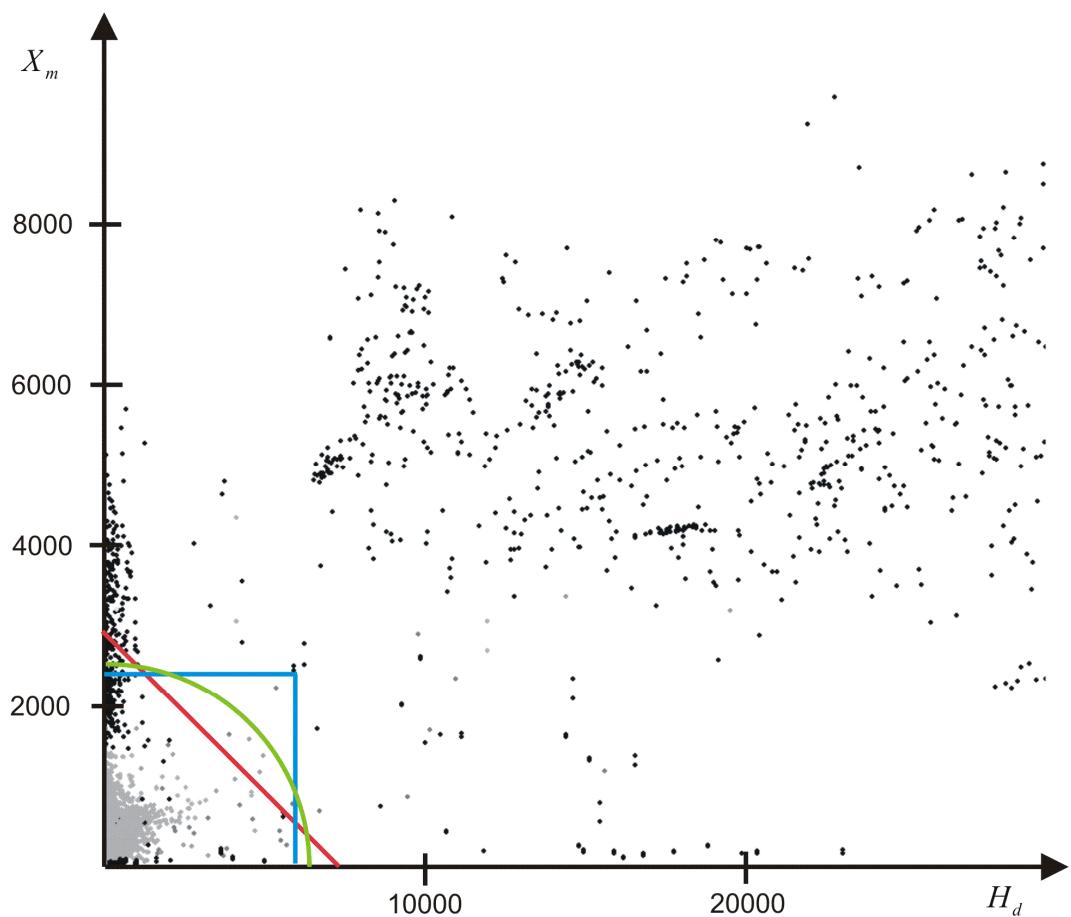
1.  $S_b$  – эффективная площадь минимума функции невязки – площадь области  $\tilde{D}$  из (1.9).
2.  $X_m$  – значение невязки в минимуме.
3.  $H_d$  –  $\chi^2$ -критерий разности гистограмм.

$H_d$  может быть вычислен стандартным образом (см., например, [41]).

#### 2.4.1.2 Классификация на основе метрик

Во многих работах предлагается выбирать пороговый метод классификации – т.е. сравнение метрик с порогом. Такой подход выглядит чрезвычайно малоэффективным.

Гораздо удобнее построить диаграмму (Рис. 2.7), по осям которой стоят значение выбранных метрик. Используя такую диаграмму можно достаточно надежно выбрать пороги для разных характеров областей решений. Более того, можно точно обозначить область значения метрик, которые характеризуют одну из ситуаций.



**Рис. 2.7. Проекция трехмерного пространства метрик на метрики  $H_d$  и  $X_m$ . Яркие точки соответствуют «плохим» кадрам. В левом верхнем углу показаны оптимальные трехпараметрические области.**

Были исследованы следующие трехпараметрические классификаторы: область - параллелепипед

$$\begin{aligned}
 S_b &< \alpha \\
 X_m &< \beta, \\
 H_d &< \gamma
 \end{aligned} \tag{2.14}$$

область – полупространство

$$\alpha S_b + \beta X_m + \gamma H_d < 1, \tag{2.15}$$

область - эллипсоид

$$\frac{S_b^2}{\alpha^2} + \frac{X_m^2}{\beta^2} + \frac{H_d^2}{\gamma^2} < 1, \tag{2.16}$$

Однако более эффективным классификатором является нейронная сеть.

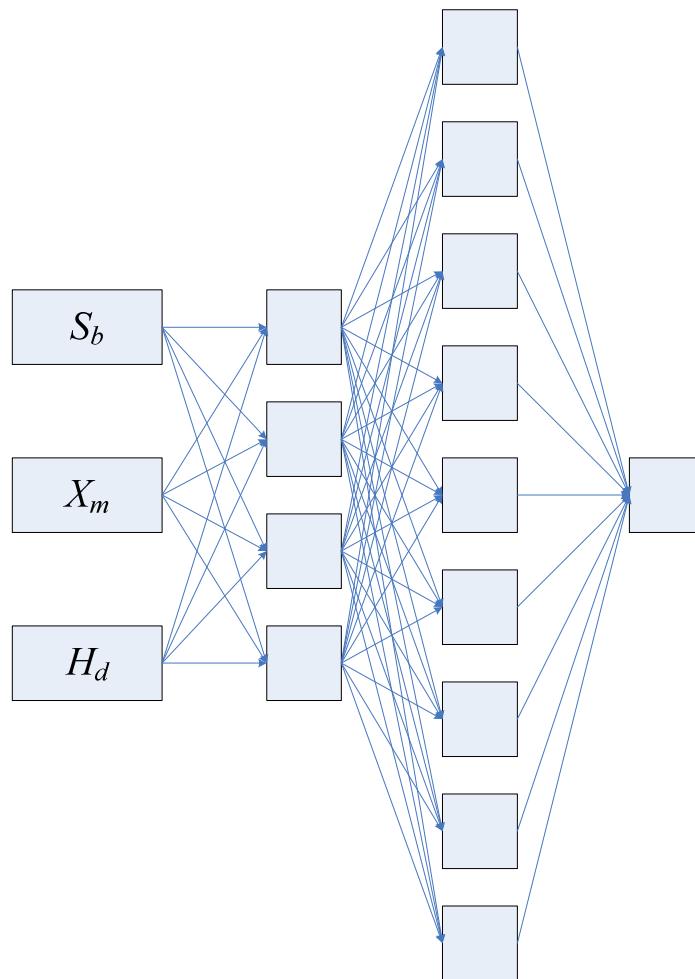


Рис. 2.8. Схема нейронной сети

В данной работе была использована трёхслойная нейронная сеть (Рис. 2.8), обученная на представительной обучающей выборке. Она включала в себя 6 последовательностей достаточно хорошего качества, снятых в разных

условиях (в комнате, на открытой местности, из движущейся машины, с БЛА, при наличии или отсутствии движения камеры и/или объектов в поле зрения) и 6 последовательностей, признанных «плохими» (см. Рис. 2.6).

Важно, что выход нейронной сети является дробной величиной от -1 до 1. При этом величины больше 0 соответствуют «хорошим» кадрам, а меньше 0 – «плохим». Исходную дробную величину можно использовать для характеристики качества кадра в дальнейшей обработке (например, как параметр регуляризации при определении геометрических трансформаций).

Для обучения нейронной сети был использован алгоритм обратного распространения [53].

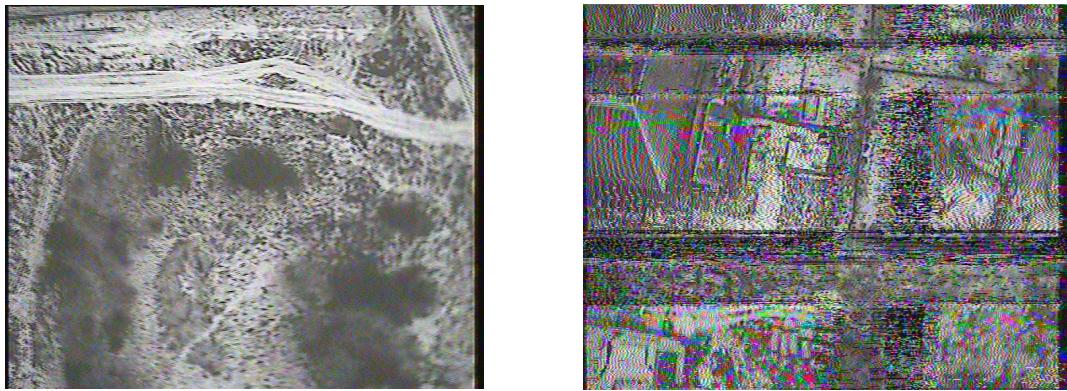
Для достижения лучшего результата использовалась простейшая временная фильтрация по критерию из  $N$  по  $K$ . Т.е. кадр признавался плохим, если из  $N$  предыдущих кадров  $K$  оказывались плохими. Этот метод весьма эффективен во всех случаях, кроме смены сцены. А, как показало исследование, детектирование смены сцены является наиболее простой задачей из решаемых.

#### 2.4.2 Детектирование особых случаев шума

В некоторых случаях можно построить детектор шумовых кадров, основанный на особых свойствах шума.

Если шум связан с низким качеством радиосвязи (что имеет место, в частности, в беспилотных летательных аппаратах - БЛА), то можно построить классификатор на основе оценки степени насыщенности цвета (см. [46]).

В таких кадрах естественное изображение, как правило, имеет блеклые цвета, в то время как шумовые кадры содержат большое количество ярких цветов.



**Рис. 2.9. Пример съемки с БЛА. Слева – качественный кадр, справа – шумовой кадр с наличием большого количества ярко окрашенных пикселей.**

Алгоритм классификации для такого случая сводится к простой схеме:

1. Находятся все точки, насыщенность которых выше порога.
2. Если количество таких точек больше порога, то кадр считается шумовым.

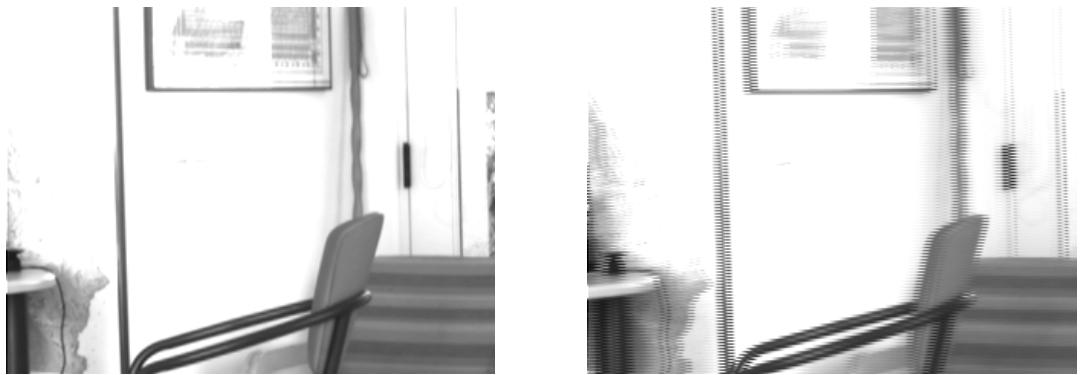
#### 2.4.3 Детектирование чересстрочной развертки

Часто кадры видеопоследовательности снимаются чересстрочной разверткой, т.е. четные и нечетные строчки снимаются в разные моменты времени. При сильной тряске камеры это приводит к искажениям изображения (Рис. 2.10), когда принятая математическая модель вовсе перестаёт быть справедливой. Однако полукадры, собранные только из четных (или нечетных) строк, имеют обычные, указанные в разделе 1.2.1, деформации.

Определение наличия эффекта чересстрочной развертки можно проводить путем сравнения корреляции между соседними строками и строками, идущими через одну, т.е. путем сравнения следующих величин:

$$\begin{cases} I_1 = \sum_{k=0}^N (F_{i(k), j(k)} - F_{i(k), j(k)+1})^2 \\ I_2 = \sum_{k=0}^N (F_{i(k), j(k)} - F_{i(k), j(k)+2})^2 \end{cases}, \quad (2.17)$$

$N$  – количество точек, по которым определяется наличие чересстрочной развертки (может быть выбрано 1-10% от количества точек в кадре),  $i(k)$  и  $j(k)$  – некоторые функции, согласно которым выбирается очередная точка.



**Рис. 2.10. Слева – кадр, на котором эффект чересстрочной развертки отсутствует, справа – присутствует. Видно, что изображение справа сильно искажено и раздвоено.**

Данной работе функции  $i(k)$  и  $j(k)$  были выбраны псевдослучайными для устранения эффекта «резонанса» с возможными регулярными структурами на кадре.

Для вынесения решения о наличии чересстрочной развертки производится сравнение:

$$I_1 / I_2 > T, \quad (2.18)$$

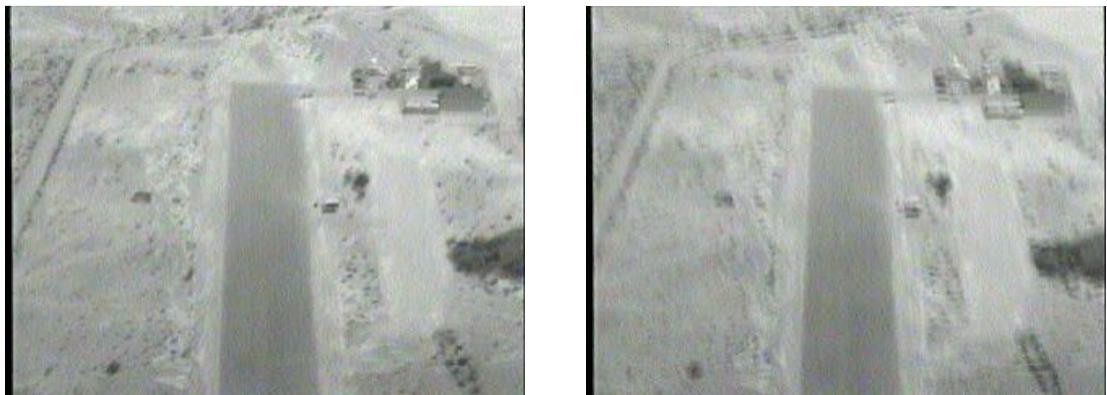
где  $T$  - пороговая константа (в данном случае была выбрана равной 1.2).

Если неравенство верно, значит эффект чересстрочной развертки присутствует.

#### 2.4.4 Отсеивание смазанных кадров

Входные кадры могут быть смазаны за счет сильных колебаний камеры (или испорчены каким-либо другим образом – Рис. 2.11). Но, т.к. эти искажения

носят случайный характер и сильно изменяются по степени от кадра к кадру, то для улучшения качества выходного изображения и качества работы алгоритма определения деформации можно пропускать кадры, испорченные сильнее других.



**Рис. 2.11. Слева – несмазанный кадр, справа – смазанный. На правом кадре видно искаженное раздвоенное изображение.**

Критерием качества здесь может служить выход алгоритма классификации на основе нейронной сети. Если максимальная информативность очередного кадра ниже средней, то такой кадр считается смазанным. Сам же критерий пропуска выбирается из других соображений, например, сколько кадров можно пропустить подряд, насколько сильно может ухудшаться изображение и т.д.

#### 2.4.5 Анализ и результаты

Основным критерием качества работы представленных алгоритмов является их кривая эффективности – т.е. зависимость количества кадров, ложно детектированных как плохие, от количества недетектированных плохих кадров. Для ее построения использовалась минимизация следующего критерия:

$$J = W(\vec{a}) + \alpha M(\vec{a}), \quad (2.19)$$

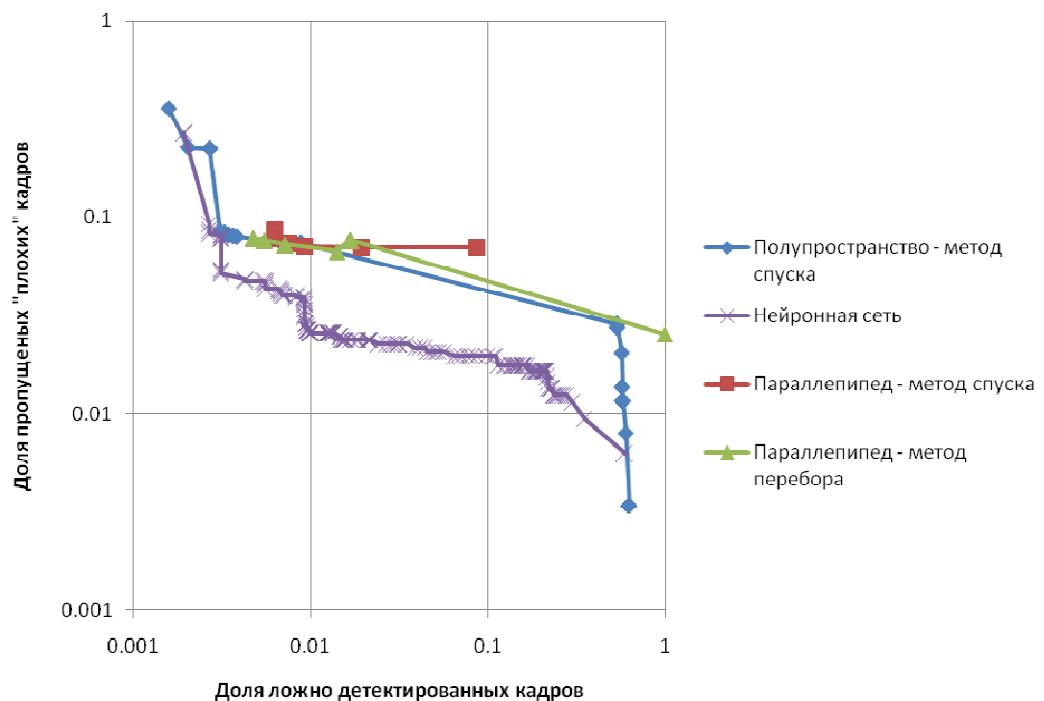
$W$  – количество «хороших» кадров, определенных как «плохие»,  $M$  – количество неопределенных «плохих» кадров,  $\alpha$  - весовой коэффициент,  $\vec{a}$  - вектор настроенных параметров (веса нейронной сети либо пороги).

Минимизируя данный критерий по вектору  $\vec{a}$  можно получить соотношение ложных и пропущенных детектирований как параметрическую функцию  $\alpha$ , и, тем самым, построить кривую эффективности (Рис. 2.12).

Однако минимизация данного критерия задача весьма непростая (за счет дискретности и наличия побочных минимумов) и обычные методы спуска, например, метод Пауэлла, недействительны. Для улучшения сходимости необходимо вводить регуляризацию, т.е. каждой точке в пространстве метрик приписывать вес, равный 1 вдали от границы области, и плавно сходящийся к 0 на самой этой границе.

Для сравнения, кривая эффективности строилась также прямым перебором. Однако, даже задаваясь достаточно мелким шагом, невозможно построить достаточно точную оценку оптимальной кривой эффективности.

На Рис. 2.12 представлены некоторые кривые эффективности для наиболее трудных случаев, когда задача неформализуема и не может быть в точности решена даже оператором (например, сильный шум на фоне полезного сигнала).



**Рис. 2.12. Кривые эффективности.** Для сравнения представлены кривые эффективности, вычисленные для трехпараметрических

## поверхностей.

В Табл. 2.1 приведены сводные результаты работы алгоритмов, касающиеся отдельных подзадач общей задачи определения плохих видеопоследовательностей, а также алгоритма детектирования чересстрочной развертки.

**Табл. 2.1 Основные характеристики алгоритма.**

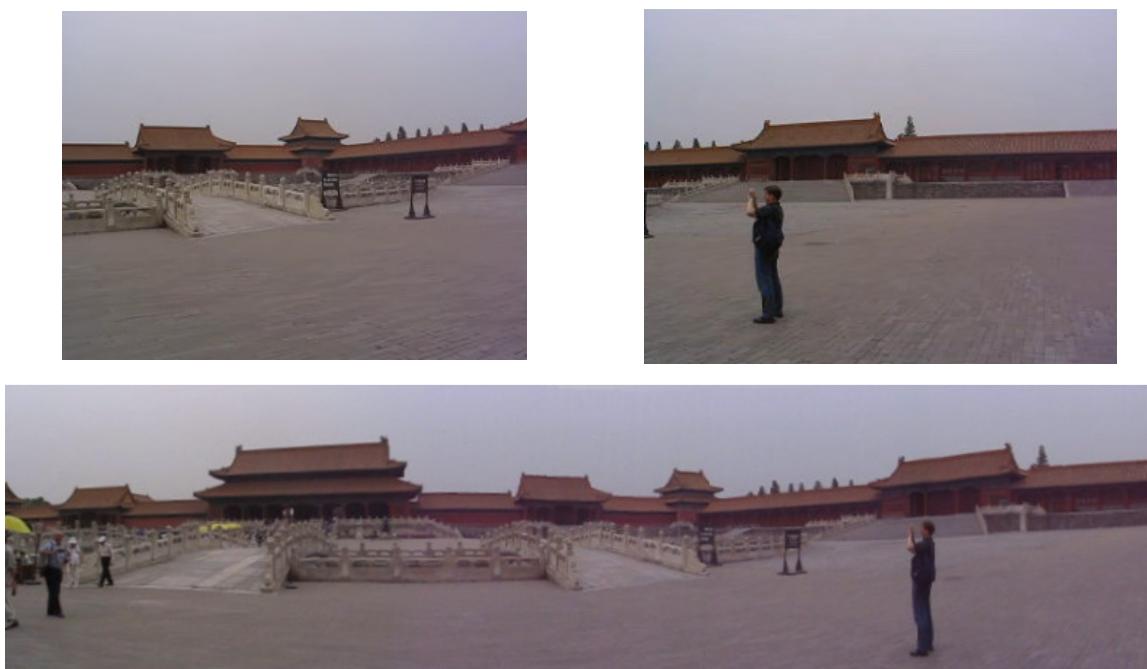
Скорость работы (кадров в сек.) Pentium M 1700МГц	160
Вероятность пропуска / вероятность ложного детектирования для задачи детектирования смены сцены	0.2% / <0.01%
Вероятность пропуска / вероятность ложного детектирования для задачи детектирования шумовых кадров при отсутствии полезного сигнала	<0.01% / <0.01%
Вероятность пропуска / вероятность ложного детектирования для задачи детектирования шумовых кадров в присутствии полезного сигнала	<0.01% / 0.01%
Вероятность пропуска / вероятность ложного детектирования для задачи детектирования чересстрочной развертки	<0.01% / <0.01%

Таким образом, решена задача оценивания качества входной видеопоследовательности и детектирования таких событий, как смена сцены, шумоподобность кадра, высокий уровень шумов на фоне полезного сигнала и др.

## 2.5 Построение панорамного кадра

Оцененные параметры геометрических преобразований можно использовать для создания панорамных кадров (мозаики) – т.е. объединения нескольких кадров в один большего размера.

Как было указано выше, создание мозаики, помимо получения кадра большого размера, позволяет улучшить качество выходных изображений.



**Рис. 2.13. Пример построения мозаики, используя аффинные преобразования. Сверху – кадры исходной последовательности, снизу – кадр мозаики.**

Следует заметить, что построение качественной мозаики возможно далеко не во всех случаях – даже если снимаемая сцена статична. Наблюдаемая картина всегда отличается от принятой в математической модели. В частности, любая пространственная структура вызывает искажение мозаики – ведь при изменении положения камеры одни объекты могут загораживать другие, блики смещаться и т.д. В особых ситуациях оптимальным является использование преобразований, отличных от аффинных. Однако аффинные преобразования позволяют получить равномерно-качественный результат на широком классе ситуаций.

Для случаев съемки при неподвижной камере - у которой изменяется только ориентация, но не положение в пространстве - необходимо учитывать фокусное расстояние камеры. Метод такого учета, вместе с алгоритмом его оценки, можно найти в [44].

Суть отличия от аффинной геометрии заключается в том, что для создания мозаики изображение необходимо строить в угловой системе

координат, в которой по осям отложены углы. У камеры с плоской матрицей по осям расположены тангенсы углов.



**Рис. 2.14. Пример построения мозаики с использованием сферической геометрии.**

Использование сферической геометрии с известными параметрами камеры позволяет создать качественную мозаику на основе алгоритма определения сдвига (см. Рис. 2.14). При этом необходимо, чтобы сдвиг между последовательными кадрами был небольшим (т.е. чтобы выполнялось равенство угла и его тангенса с достаточно хорошей точностью).



**Рис. 2.15. Пример построения мозаики с использованием проективных преобразований.**

Для случаев наблюдения плоской поверхности (что особенно актуально при аэросъемке) оптимальным преобразованием является проективное – оно точно описывает изменение изображения при любых изменениях положения камеры (см. Рис. 2.15).

Здесь, однако, следует заметить, что для получения такой мозаики в реальном времени необходимы кадры с хорошим отношением сигнал-шум и большим количеством информативных объектов.

## 2.6 Выводы

1. Созданы и реализованы различные алгоритмы отделения случайной составляющей геометрических трансформаций от регулярной, связанной с управляемым движением камеры. Предложено несколько методов, и выявлены их достоинства и недостатки.
2. Создан, реализован и исследован алгоритм определения качества видеопоследовательности, позволяющий детектировать такие ситуации, как смена сцены, низкое отношение сигнал-шум, шумоподобность и др. Алгоритм основан на классификации методами нейронных сетей в пространстве выбранных метрик.
3. Создан, реализован и исследован алгоритм детектирования наличия чересстрочной развертки.
4. Реализован алгоритм стабилизации видеоизображений, основанный на описанном в первой главе алгоритме определения геометрических преобразований.
5. Реализован алгоритм создания панорамного кадра – т.е. кадра большого размера, полученного путем объединения нескольких кадров видеопоследовательности в один с учетом оцененных сдвигов.

### 3 Сопровождение движущихся объектов

#### 3.1 Введение

Сопровождение выделенных объектов в видеопоследовательности – актуальная задача для систем наблюдения, охранных систем, интеллектуальных систем управления автомобилем и робототехники. Кроме того, результаты решения этой задачи могут быть использованы как начальный этап для создания более сложной системы анализа видеопоследовательности.

В данной главе рассмотрены только вопросы сопровождения выделенных оператором объектов. Т.е. оператор (либо программа) задает положение объекта на кадре в начальный момент времени (и, возможно, некоторые параметры объекта), а алгоритм должен определить положение этого объекта на последующих кадрах.

На данный момент задача сопровождения объектов в общем виде не имеет единственного удовлетворительного решения. Существует два различных способа ее решения. Первый – на основе отделения движущихся объектов от фона с последующей классификацией (чтобы выбрать на кадре из всех объектов только интересующий). Второй – на основе поиска на следующем кадре выделенного объекта с предыдущего кадра.

В первом случае алгоритм работает с неподвижными камерами и имеет ряд недостатков, например, невозможность сопровождать неподвижные объекты или объекты на сильно изменяющемся фоне. Такие задачи характерны для систем видеонаблюдения и мониторинга (большинство коммерческих продуктов относится именно к этой категории). Здесь сопровождение объекта основано на использовании стационарности хорошо известного «подстилающего» фона. Обзор таких методов дан в [23].

Для сопровождения объектов на более сложном фоне, как правило, используются алгоритмы с накоплением информации (фильтрацией). Базовыми являются алгоритмы фильтрации на основе фильтра Калмана [24], [25]. Среди

наиболее изощренных следует выделить фильтрацию методом частиц [26], [27] и с использованием ядра объекта [28], [29].

В данной главе рассмотрены несколько алгоритмов сопровождения объектов. В разделе 3.2 описан базовый алгоритм сопровождения, полностью основанный на алгоритмах главы 1. В разделе 3.3 представлены различные методы определения контура движущегося объекта, а также методы сопровождения с использованием таких контуров. В разделе 3.4 рассмотрен метод одновременного сопровождения и оконтуривания в условиях существенной априорной неопределенности. Раздел 3.5 посвящен методам оценки уровня собственного шума на изображении. Знание этого шума является важным элементом алгоритмов сопровождения. Раздел 3.6 посвящен вопросам управления нацеливанием камеры на объект. Исследование характеристик алгоритмов приведено в раздел 3.7. Последний раздел данной главы посвящен выводам.

### **3.2 Базовый алгоритм сопровождения**

Рассмотрим базовый алгоритм сопровождения.

Для сопровождения объекта достаточно знать его сдвиг от кадра к кадру. Воспользовавшись методами предыдущей главы определяется сдвиг выбранной части кадра. Таким образом, изложенные выше методы могут без существенных изменений быть использованы для решения задачи сопровождения объектов.

Для корректной работы алгоритма оператору, помимо положения объекта, необходимо также задать его размеры. Иначе в окно сопровождения может попасть слишком большое количество фона, что снизит качество работы и может привести к потере сопровождения в случае, если алгоритм «зацепится» за фон.

В данной задаче алгоритм накопления играет существенную роль. Он позволяет значительно уменьшить влияние фона, который, за счет сглаживания, потеряет информативность (см. часть 1.2.5.3).

Если объект сильно поворачивается и масштабируется, необходимо определить его угол поворота. Этого нельзя сделать методами, изложенными в предыдущей главе, т.к. для объекта, как правило, нельзя выбрать несколько информативных окон. Для определения параметров используется метод нелинейной минимизации невязки по всем параметрам преобразования. Т.е. вместо формул (1.5) и (1.6) используются

$$X(F(t-1), T_Q(F(t))) = \frac{1}{S(M)} \sum_{(i,j) \in M(Q)} (F(t-1)_{ij} - \tilde{F}(t)_{i+ki \cos \varphi + kj \sin \varphi + dx, j+ki \sin \varphi + kj \cos \varphi + dy})^2, \quad (3.1)$$

$$\hat{Q} = (d\hat{x}, d\hat{y}, \hat{k}, \hat{\varphi}) = \arg \min_{(dx, dy, k, \varphi) \in D} X(F(t-1), T_Q(F(t))), \quad (3.2)$$

где  $\tilde{F}(t)_{i+ki \cos \varphi + kj \sin \varphi + dx, j+ki \sin \varphi + kj \cos \varphi + dy}$  - интерполированное к дробным координатам значение яркости в пикселе.

Задачу поиска минимума можно решать и методом Паузлла. Такой способ, однако, весьма трудоемок и вряд ли может быть выполнен в реальном времени.

Для улучшения качества сопровождения и снятия некоторых ограничений (в частности – необходимости задания размеров объекта) необходимо использовать более сложные методы.

Разделим сопровождение объекта на две условные фазы:

1. Завязка сопровождения и первоначальное накопление информации.
2. Устойчивое сопровождение на основе накопленной информации совместно с ее обновлением.

Большинство существующих методов относится ко второй категории – т.е. решают задачу обновления накопленных статистических данных. Все они основаны на рассмотрении способов обновления некоторой модели (фильтрации).

В данной работе основное вниманиеделено методу (см раздел 3.4) завязки сопровождения, который позволяет накопить первоначальную информацию (прежде всего о форме и размере объекта) в условиях сильной априорной неопределенности.

### 3.3 Алгоритмы оконтуривания

Помимо сопровождения объектов, часто требуется определить их форму и размер - оконтурить. Это улучшает визуальное восприятие сопровождения.

Здесь под контуром понимается не только сама линия контура, но и его внутренняя часть, т.к. многие алгоритмы оконтуривания ищут не границу, а область объекта.

Как было указано выше, иногда оконтуривание является строго необходимым для качественного сопровождения объектов. В более простых случаях целесообразно использовать алгоритм оконтуривания после алгоритма сопровождения. Т.е. после нахождения положения объекта на очередном кадре, найти его контур. Информация о контуре используется для последующего сопровождения путем исключения точек, не принадлежащих объекту, из рассмотрения при вычислении функции невязки.

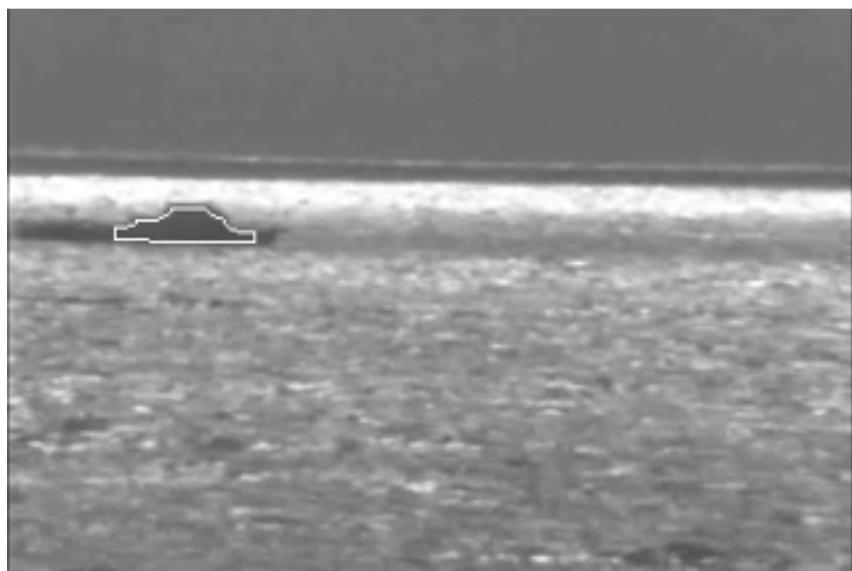
Ниже приведены несколько алгоритмов оконтуривания. Они подразделяются на два класса:

1. Методы обнаружения объекта с заданными свойствами. Они позволяют не только определить контур объекта, но и провести его обнаружение и сопровождение. Первые три описанных ниже метода относятся к этой категории. Основным их недостатком является требовательность к параметрам объекта – они могут адекватно работать только с объектами определённого класса.
2. Методы оконтуривания, явным образом использующие вычисленный сдвиг объекта. Четвертый из описанных ниже – метод временной фильтрации – относится к этой категории. Он достаточно универсален и работает с широким классом объектов, но не позволяет производить их обнаружение и независимое сопровождение.

### 3.3.1 Оконтуривание по цвету

Самым простым способом оконтуривания является выбор объекта как области на кадре с цветом, близким к цвету выбранной точки. Естественно выбирать не только точки, точно соответствующие по цвету, но и близкие к ним.

Этот метод хорошо работает для равномерно окрашенных объектов, а для случаев пестрых объектов фактически бессмысленен.



**Рис. 3.1. Определение контура объекта по цвету.**

Особо хорошо этот метод работает при съемках инфракрасной камерой. В таких условиях яркость на кадре фактически соответствует температуре тела. Как правило, идет сопровождение теплых объектов – людей, автомобилей и т.д. Возможна и другая ситуация – сопровождение объектов на фоне солнечного света (например, отраженного от водной поверхности). В таких условиях объект значительно более темнее фона.

### 3.3.2 Оконтуривание по контрастности

Другим очевидным методом оконтуривания является поиск областей с большой дисперсией цветов. Т.е. выбираются те области, средний квадрат разности яркости пикселей в соседних точках больше порогового значения.



**Рис. 3.2. Определение контура объекта по контрастности.**

Этот метод весьма специфичен и хорошо работает только для контрастных объектов на равномерном фоне – например, при съемке самолета на фоне неба.

### 3.3.3 Оконтуривание методом выделения границ

Для поиска контура объекта можно использовать стандартными методами – например, хорошо известным методом Канни [47]. Следует заметить, что данный метод находит не границы объекта, а фактически все области на кадре, которые могут быть трактованы как линии. Это означает что: во-первых, если фон имеет большое количество линий, то данный метод не позволит отделить объект от фона; во-вторых, если у объекта присутствует внутренние линии, то определенный этим методом контур не будет контуром объекта, а будет содержать большее количество деталей.



**Рис. 3.3. Определение контура объекта методом Канни.**

По своему характеру и характеристикам этот алгоритм похож на предыдущий – т.е. определяет достаточно контрастный объект на малоинформационном фоне (см. например, Рис. 3.3).

### 3.3.4 Временная фильтрация

Для поиска контура объекта можно использовать тот факт, что движение его пикселей связано простым законом (в простейшем случае – у всех пикселей одинаковый линейный сдвиг), в то время как закон движения пикселей фона существенно отличается.

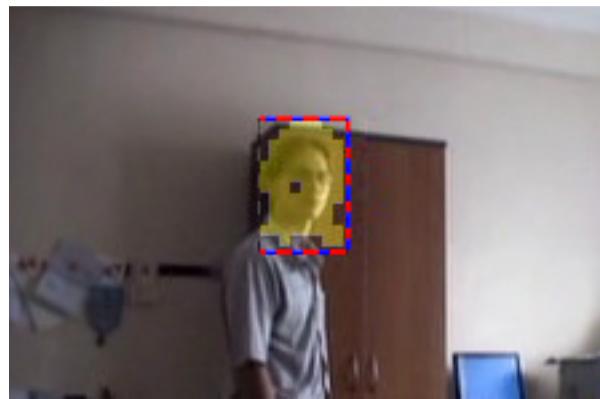
Последовательно такая задача решена в следующем разделе методом максимального правдоподобия.

Здесь приведем упрощенный, но более вычислительно-эффективный метод.

Задавшись определенным сдвигом объекта  $(d\hat{x}, d\hat{y})$  вычислим кадр разности значений яркости

$$\|D_{ij}\| = \|F_{ij}(t) - F_{i+d\hat{x}, j+d\hat{y}}(t-1)\|. \quad (3.3)$$

Выбирая из этого кадра значения меньшие порога, получаем кадр контура объекта.



**Рис. 3.4. Определение контура объекта методом временной фильтрации.**

Данный метод работает в широком классе условий – когда сдвиг объекта определяется достаточно надежно и когда он больше пикселя.

### 3.3.5 Постобработка контура

Для улучшения качества определения контура необходимо использовать постобработку.

Первым этапом идёт морфологическая обработка [48]. Заполняются пустые области внутри объекта (это особенно актуально для метода Канни), отфильтровываются побочные несвязанные области, т.е. выбирается из всех выделенных областей та, которая находится ближе всего к центру объекта. Применяются методы замыкания незамкнутых контуров.

Другим способом улучшения качества определения контура является усреднение по времени.

Самым простым – и, тем не менее, оптимальным – методом фильтрации контура объекта является метод из  $N$  по  $K$  (см. [20]). Он сводится к тому, что

часть считается принадлежащей объекту, если из  $N$  последних кадров она была детектирована на  $K$  кадрах.

### 3.3.6 Алгоритмы обнаружения и сопровождения на основе контуров объектов

Алгоритмы оконтуривания могут быть использованы для обнаружения и сопровождения объектов. Для этих целей пригодны первые три описанных алгоритма.

Их применение для всего кадра позволяет обнаружить все объекты заданного класса на кадре. Дальше необходимо решить задачу разделения всех выбранных областей на отдельные объекты и соотнесение этих объектов с объектами на последующих кадрах. Такая задача имеет богатую историю изучения, но выходит за рамки данной работы.

В данной работе рассматривается только задача сопровождения на основе выделенных контуров. За положение объекта принимается центр контура.

Для сопровождения обнаруживается контур объекта в районе вокруг его центра на предыдущем кадре, а затем определить центр этого контура. Центр контура определяется по следующей очевидной формуле:

$$\vec{c} = \sum_{i \in I} \vec{r}_i / M(I), \quad (3.4)$$

где  $I$  - множество точек контура объекта,  $M(I)$  - количество таких точек, а  $\vec{r}_i$  - их координата.

Этот метод сопровождения чрезвычайно устойчив для объектов подходящего класса, т.к. у него нет проблем интегрального нарастания ошибки, потери объекта при его сильном изменении и т.д. Но следует повторить, что он работает только на объектах достаточно узких классов.

Алгоритмы оконтуривания могут также служить для улучшения характеристик базового алгоритма сопровождения. Для этого при вычислении функции невязки берутся только те точки, которые принадлежат объекту. При этом используется контур объекта, вычисленный на предыдущем кадре.

Предполагается, что он, в отличие от положения объекта, изменяется от кадра к кадру несущественно.

### **3.4 Алгоритм сопровождения методом максимального правдоподобия.**

Рассмотрим усовершенствованный алгоритм сопровождения. Он разрабатывался для управляемых камер с возможностью большого увеличения, что наложило достаточно серьезные требования на его функциональность.

Алгоритм не имеет в своей основе представления о статичности и известности фона. Это обусловлено двумя причинами: во-первых, камера может быстро перемещаться (особенно в условиях сильного увеличения), что не дает возможности накопить информацию о фоне; во-вторых, фон может быть сильно нестационарным, например, поверхность моря.

Еще одним условием является отсутствие априорных знаний о размерах и форме объекта. Это требование связано с тем, что алгоритм разрабатывался для систем, в которых оператор, помимо выделения объектов, вынужден управлять камерой. В этом случае он не имеет возможности точно задать размеры объекта, которые могут варьироваться из-за изменения увеличения.

Основная проблема в данных условиях заключается в завязке сопровождения, т.е. в обработке нескольких первых кадров, пока не накоплено достаточное количество информации. Это особенно актуально, т.к. алгоритм должен управлять камерой, следящей за объектом. Если он не справится с задачей, объект исчезнет из поля зрения камеры за несколько кадров.

Для завязки сопровождения в данном случае необходимо одновременно с задачей определения сдвига объекта решить задачу определения его контура. Иначе точность определения сдвига будет снижена за счет попадания фона в окно, по которому вычисляется невязка. Весьма вероятен срыв сопровождения за счет того, что алгоритм «зацепится» за фон. Один из примеров таких алгоритмов представлен в [30]. Здесь заложено жесткое ограничение на форму объекта – она должна быть близка к прямоугольной.

Предложенный в данном разделе алгоритм позволяет решить задачу сопровождения и оконтурирования движущихся объектов при отсутствии априорной информации об их форме, размерах и характере изменения фона.

### 3.4.1 Оценка функции максимального правдоподобия

Будем решать задачу одновременного определения контура объекта и его сдвига методом максимального правдоподобия. В случае отсутствия априорной информации данный подход выглядит наиболее предпочтительным (см. [20]).

Кадр  $F_t$  момента времени  $t$  состоит из конечного количества частей – пикселей (для ускорения алгоритма можно использовать части, состоящие из нескольких пикселей; далее будет использоваться термин «часть»). Предположим, что каждая часть полностью принадлежит либо объекту, либо фону. Первоначальный выбор объекта осуществляется оператором и заключается в выборе части  $i_0$ , которая заведомо принадлежит объекту.

Запишем функцию правдоподобия для данной задачи, т.е. принадлежности всех частей кадра к объекту или фону:

$$J(I) = \prod_{i \in I} p(i \in \text{object}) \prod_{i \notin I} p(i \notin \text{object}), \quad (3.5)$$

где  $i$  – номер части в кадре,  $I$  – множество частей, принадлежащих объекту,  $p(i \in \text{object})$  - вероятность того, что часть  $i$  принадлежит объекту.

Таким образом, для нахождения сдвига необходимо найти максимум правдоподобия, т.е.

$$I = \arg \max_I J(I), \quad (3.6)$$

Основная проблема заключается в нахождении вероятности того, что часть принадлежит объекту. Т.к., по условиям задачи, объект отличается от фона только вектором сдвига, такая вероятность должна зависеть от сдвига объекта и фона.

Оценка функции правдоподобия проводится с использованием функции невязки, аналогичной использованной в предыдущей главе. Вычислим

функцию невязки (как функцию сдвига  $(dx, dy)$ ) для каждой части и всего объекта:

$$\begin{aligned}\tilde{C}_i(dx, dy) &= \sum_{l, k \in P_i} (F_t(l, k) - F_{t-1}(l - dx, k - dy))^2, \\ \tilde{S}(dx, dy) &= \sum_{i \in I} \tilde{C}_i(dx, dy),\end{aligned}\tag{3.7}$$

где  $P_i$  - пиксели, принадлежащие части  $i$ .

Будем рассматривать вычисленные функции невязки как оценки истинных  $C_i$  и  $S$ . Минимум этих функций будет оценкой сдвига каждой части или всего объекта. Можно было бы выбрать части объекта, сдвиг которых близок к сдвигу части  $i_0$ . Но, как показали эксперименты, этот алгоритм чрезвычайно неустойчив из-за больших ошибок определения сдвига отдельной части.

Более точным будет вычисление вероятности  $p(i \in \text{object})$  как вероятности того, что минимум  $C_i$  и  $S$  совпадают. Тогда

$$p(i \in \text{object}) = \sum_{\delta x, \delta y} p_i(dx, dy) p_S(dx, dy)\tag{3.8}$$

или приближенно

$$p(i \in \text{object}) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} p_i(dx, dy) p_S(dx, dy) d(dx) d(dy)\tag{3.9}$$

где  $p_i(dx, dy)$  и  $p_S(dx, dy)$  - вероятности того, что значение сдвига части или объекта равны  $(dx, dy)$ .

Эти вероятности могут быть оценены из оценок функций невязок. Для этого заменим распределение оценок сдвига каждой части нормальным распределением (как показали исследования, такая замена является вполне адекватной):

$$p_i(dx, dy) = \frac{1}{2\pi\sqrt{\det(\Gamma_i)}} e^{-\frac{1}{2}(\vec{x} - \vec{x}_i)^T \Gamma_i^{-1}(\vec{x} - \vec{x}_i)}, \text{ где} \quad (3.10)$$

$$\vec{x} = (dx, dy), \quad \Gamma_i = \begin{pmatrix} \sigma_{ix}^2 & \rho_i \sigma_{ix} \sigma_{iy} \\ \rho_i \sigma_{ix} \sigma_{iy} & \sigma_{iy}^2 \end{pmatrix}$$

и аналогично для  $S$ .

Параметры этого распределения  $\vec{x}_i, \sigma_{ix}, \sigma_{iy}, \rho_i$  выберем исходя из оценки функции невязки.

Для этого вычисляется доверительная область нахождения минимума функции невязки по формулам (1.7), (1.8) и (1.9). По этой области рассчитываются параметры распределения

$$\vec{x}_i = \begin{pmatrix} x_i \\ y_i \end{pmatrix} = \frac{1}{M(\tilde{D})} \sum_{\delta x, \delta y \in \tilde{D}} \begin{pmatrix} dx \\ dy \end{pmatrix},$$

$$\begin{pmatrix} \sigma_{ix} \\ \sigma_{iy} \end{pmatrix} = \frac{1}{M(\tilde{D})} \sum_{\delta x, \delta y \in \tilde{D}} \begin{pmatrix} (dx - x_i)^2 \\ (dy - y_i)^2 \end{pmatrix}, \quad (3.11)$$

$$\rho_i = \frac{1}{M(\tilde{D})} \frac{1}{\sigma_{ix} \sigma_{iy}} \sum_{\delta x, \delta y \in \tilde{D}} (dx - x_i)(dy - y_i)$$

Тогда, вычисляя интеграл (3.9), находим:

$$p(i \in object) = \frac{\exp(\frac{B}{2A})}{2\pi\sqrt{-A}}, \text{ где} \quad (3.12)$$

$$A = 2\rho_i \rho_F \sigma_{Fx} \sigma_{Fy} \sigma_{ix} \sigma_{iy} + \sigma_{Fx}^2 ((\rho_F^2 - 1) \sigma_{Fy}^2 - \sigma_{iy}^2) + \sigma_{Fy}^2 ((\rho_i^2 - 1) \sigma_{iy}^2 - \sigma_{Fy}^2),$$

$$B = (\sigma_{Fy}^2 + \sigma_{iy}^2)(x_F - x_i)^2 + (\sigma_{Fx}^2 + \sigma_{ix}^2)(y_F - y_i)^2 - 2(\rho_F \sigma_{Fx} \sigma_{Fy} + \rho_i \sigma_{ix} \sigma_{iy})(x_F - x_i)(y_F - y_i)$$

### 3.4.2 Поиск максимума правдоподобия

Мы можем оценить функцию правдоподобия как функцию от  $I$ . Задача нахождения ее максимума является не менее трудной, чем оценка самой функции.

Основная проблема заключается в том, что функция невязки  $S$  для объекта зависит от контура объекта  $I$ . Это затрудняет поиск максимума. Задачу можно решить методом перебора, что чрезвычайно трудоемко и исключает практическое применение. Для сокращения объема вычисления и повышения устойчивости части выбираются как квадраты размером  $8 \times 8$  пикселей.

Поэтому предлагается следующая схема: контур объекта вычисляется одним из стандартных алгоритмов кластеризации, а затем рекуррентно уточняется, используя функцию правдоподобия.

Задачей алгоритма кластеризации является нахождение первого приближения контура объекта. Кластеризация ведется по сдвигам отдельных частей – т.е. вычисляется множество сдвигов всех частей и делится на две группы, одна из которых соответствует объекту, а вторая – фону.

В качестве алгоритма кластеризации были использованы два хорошо известных алгоритма: на основе матрицы сходства и метод К-средних. Описание обоих методов дано в [46].

Метод на основе матрицы сходства позволяет из всего множества выделить один наиболее различимый кластер. Это важно в тех случаях, когда фон нестационарный и не может быть удовлетворительно описан одним кластером – а значит, задача становится с неизвестным количеством кластеров.

Кластер, полученный таким методом, может быть либо кластером объекта, либо кластером фона (если фон стационарный). Если часть  $i_0$  не принадлежит найденному кластеру, то был найден кластер фона; кластер объекта может быть найден инверсией всего множества.

Метод К-средних предполагает априорное задание количества кластеров – в данном случае их два. Кластер объекта описывается его положением в пространстве сдвигов (точнее – координатами своего центра). Кластер фона описывается минимальным расстоянием от центра кластера объекта. Т.е. все точки, находящиеся на расстоянии большем этого, считаются принадлежащими

фону. Такой выбор параметров кластеров позволяет работать с нестационарным фоном, сдвиги разных частей которого не группируются вокруг некоторого центра.

Первый метод дает несколько лучшие результаты, но и существенно медленнее работает. Все тестовые результаты приведены для метода К-средних.

Задав начальное разбиение кадра на объект и фон можно найти локальный максимум функции правдоподобия методом последовательных приближений.

Для этого вычисляется функция невязки  $\tilde{S}(dx, dy)$  для объекта по контуру  $I$ . Затем для каждой части находятся вероятности  $p(i \in object)$  и  $p(i \notin object)$ . Если  $p(i \in object) > p(i \notin object)$ , то часть принадлежит объекту, иначе – фону (это следует из критерия максимизации функции правдоподобия). Таким образом, строится новый контур объекта. Итерация повторяется (т.е. для нового контура вычисляется  $\tilde{S}(dx, dy)$  и т.д.). Когда контур  $I$  перестанет изменяться – это будет означать, что достигнут локальный максимум, и итерации следует прекратить.

На практике такой подход позволяет достаточно быстро (за 3-4 итерации) находить локальный максимум, который достаточно близок или вовсе совпадает с глобальным.

### 3.4.3 Фильтрация

Предложенный алгоритм нацелен на работу в условиях априорной неопределенности и малого количества информации. Т.е. он оптимально работает на нескольких первых кадрах. Для обработки более длинных последовательностей целесообразно использовать накопление информации – фильтрацию.

Здесь уместны все те методы постобработки, что и для алгоритмов оконтурирования (см. часть 3.3.5).

Для более точного определения сдвига используется накопление эталона – см. часть 1.2.5.3.

#### 3.4.4 Слабообусловленная задача

Данный метод имеет одну особенность – он предполагает, что объект всегда движется по отношению к фону. Это накладывает серьезные ограничения. В частности, маневрирующий объект может останавливаться на какое-то время. Более того, объект может занимать весь кадр – т.е. фактически являться фоном (сопровождать его в таком случае бывает необходимо по двум причинам – для управления камерой и для стабилизации объекта в случае дрожания камеры).

В подобных ситуациях предложенный выше алгоритм будет более-менее случайным образом разделять кадр на объект и фон. Это снижает точность определения сдвига (однако не катастрофически, т.к. все части кадра имеют одинаковый сдвиг), но полностью лишает смысла найденный контур объекта.

Для качественного решения задачи необходимо определить, содержится ли в кадре движущийся объект или нет, т.е. определить количество кластеров в задаче кластеризации. К сожалению, на сегодняшний день данная задача не имеет точного решения. Это связано с тем, что гипотеза с большим количеством степеней свободы (т.е. когда объект существует) всегда будет точнее походить под наблюдаемые данные (будет иметь большее правдоподобие), чем гипотеза с меньшим количеством степеней свободы.

Классическим методом (см., например, [46]) разрешения этого противоречия является введение эмпирических «штрафов», т.е. из функции правдоподобия вычесть некий «штраф», растущий вместе с ростом количества степеней свободы.

Для данной задачи этот подход не дает удовлетворительного результата. Воспользуемся другим методом.

Помимо сдвига объекта, определенного описанным выше алгоритмом, определяется сдвиг всего кадра. Если разность между сдвигами меньше порога (0.1 – 0.5 пикселей), то считается, что фон движется так же, как и объект.

Метод позволяет достаточно хорошо детектировать подобные ситуации. Но относительно велика вероятность ложного детектирования. Т.е. в некоторых случаях алгоритм позволяет определить довольно точно контур объекта, но сравнение с порогом отсеивает этот случай как слабообусловленный.

### 3.4.5 Возможные модификации

Предложенный метод максимального правдоподобия может быть модифицирован для учета более тонких эффектов. В частности, может быть учтен контур объекта на предыдущих кадрах путем добавления в произведение (3.5) соответствующих условных вероятностей. Такая модификация, однако, может быть сведена к известному методу из  $N$  по  $K$  (см. раздел 3.3.5).

Можно ввести «штраф» на периметр контура, чтобы объект получался более компактным. Такой учет приводит к невозможности максимизации функции правдоподобия изложенными выше методами, т.к. она перестает быть произведением вероятностей, независимых для каждой части. Это приводит к необходимости решать задачу полным перебором, что недопустимо.

Другой интересной модификацией может служить введение коэффициента масштабирования. Это приводит к необходимости максимизации функции правдоподобия в трехмерном пространстве. Несмотря на то, что данный метод весьма требователен к вычислительным ресурсам, в перспективе он вполне может быть использован в системах реального времени.

## 3.5 Алгоритм оценки шума

Для реализации вышеизложенного алгоритма необходимо иметь оценку уровня собственного шума фотоприемной матрицы. Самый простой вариант – это задать шум как константу исходя из технических характеристик камеры.

Такой вариант не всегда осуществим. Рассмотрим здесь два метода оценки шума на основе самого изображения.

Первый метод заключается в вычислении среднеквадратичного отклонения яркости пикселей в пределах небольшого окна. Перебираются все его возможные положения на кадре. Верхней оценкой уровня шума будет минимальное значение среднеквадратичного отклонения, т.е.

$$n_{\max}^2 = \min_{x,y} \left( \sum_{|i| \leq d, |j| \leq d} F_{x+i, y+j}^2 / (2d+1)^2 - \left( \sum_{|i| \leq d, |j| \leq d} F_{x+i, y+j} / (2d+1) \right)^2 \right)^2, \quad (3.13)$$

где  $n_{\max}$  - оценка уровня шума сверху,  $d$  - размер окна.

Метод дает точную оценку в случае, если на кадре есть монотонные области. В них разброс значений яркости обусловлен только собственным шумом фотоприемника. Следует исключить из рассмотрения области с яркостью, близкой к максимальной, т.к. в них за счет эффекта насыщения, среднеквадратичное отклонение может быть заметно меньше уровня шума.

Другой способ оценки шума - использование минимума функции невязки (1.7). Метод даёт неправильную оценку в случае, если в пределах окна вычисления невязки присутствуют движущиеся объекты.

Для устранения этого недостатка необходимо провести калибровку камеры – определять уровень шума в произвольно выбранных окнах на протяжении нескольких (лучше большого количества) кадров. Тогда минимум вычисленных значений будет соответствовать уровню шума, т.к. на таком окне было меньше всего движущихся объектов (как правило, всегда попадается такое окно, на котором их не было вовсе).

Уровень шума изменяется при изменении диафрагмы и выдержки камеры, так что подобная калибровка применима далеко не всегда.

### 3.6 Алгоритм управления приводом камеры

Определив положение объекта на очередном кадре и зная геометрические параметры камеры (разрешающую способность в градусах на пиксель), можно решить задачу управления камерой для слежения за объектом.

Существенным моментом здесь является задержка между моментом снятия информации с матрицы фотоприемника и ее обработки. Она может достигать нескольких десятых секунды (подробности см. в главе 5).

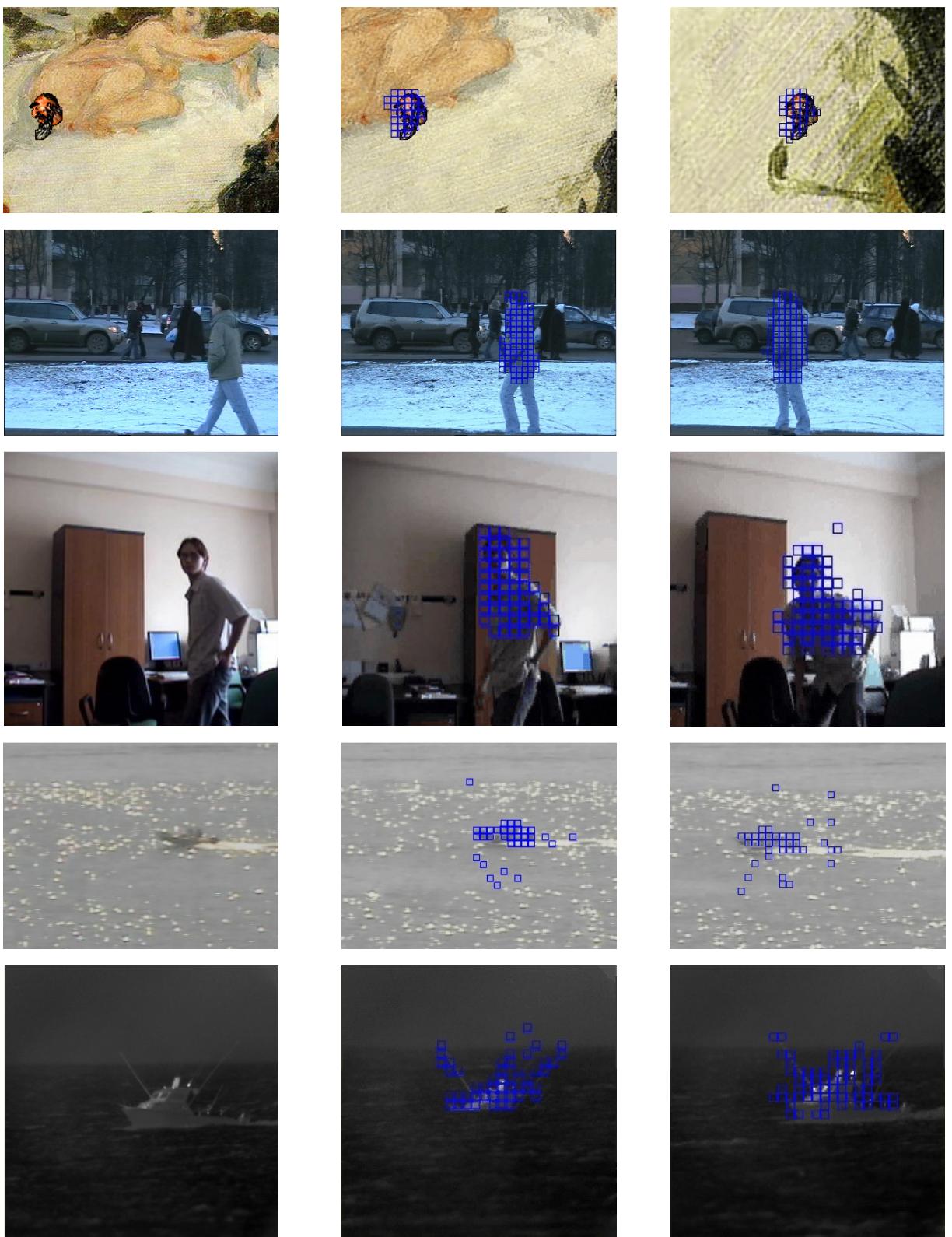
Точное решение задачи управления для такого случая дает очевидный алгоритм нацеливания камеры – необходимо нацелить камеру в точку экстраполированного положения объекта (экстраполяция проводится как раз на величину задержки). Для выполнения экстраполяции оптимальным является использование полиномов 1-3 степеней.

### ***3.7 Анализ и результаты***

Работа алгоритмов была проверена на большом количестве видеопоследовательностей разного характера, в том числе и из популярной открытой базы данных ([http://www.elec.qmul.ac.uk/staffinfo/andrea/avss2007\\_d.html](http://www.elec.qmul.ac.uk/staffinfo/andrea/avss2007_d.html)). Особенно внимательно исследовались случаи существенно нестационарного фона и быстрых перемещений камеры (а, значит, и фона).

На Рис. 3.5 приведены результаты работы алгоритма с различными видеопоследовательностями. В верхнем ряду представлены результаты работы алгоритма на синтетических кадрах. Два средних ряда – сопровождение человека на нестационарном фоне. В двух нижних рядах – сопровождение лодки на водной поверхности с яркими информативными бликами при перемещающейся камере. Видно, что некоторые части фона ошибочно определяются как принадлежащие объекту. Они в дальнейшем отсеиваются методами морфологической обработки. На изображениях нижнего ряда видно, что определились как части объекта даже удочки и антенны.

Помимо реальных видеопоследовательностей, качество работы алгоритма проверялось и на модельных, где на нестационарный фон накладывалось изображение движущегося объекта. Метод позволяет получить оценки точности алгоритма определения сдвига. На исследованных видеопоследовательностях СКО определения сдвига составила 0.08 пикселей.



**Рис. 3.5. Результаты работы алгоритма. Синими квадратами обозначены части, определенные как принадлежащие объекту.**

Характеристики всех описанных в данном разделе алгоритмов приведены в Табл. 3.1.

**Табл. 3.1 Сводная таблица характеристик предложенных алгоритмов.**

Метод	Базовый алгоритм	Сопровождение на основе оконтурирования	Метод максимального правдоподобия
Точность, пиксели	0.07	0.2	0.07
Скорость, кадры в секунду	$10^3$	$10^4$	30
Среднее время до срыва сопровождения, кадры	$10^3$	$>10^5$	$\sim 10^4$

### 3.8 Выводы

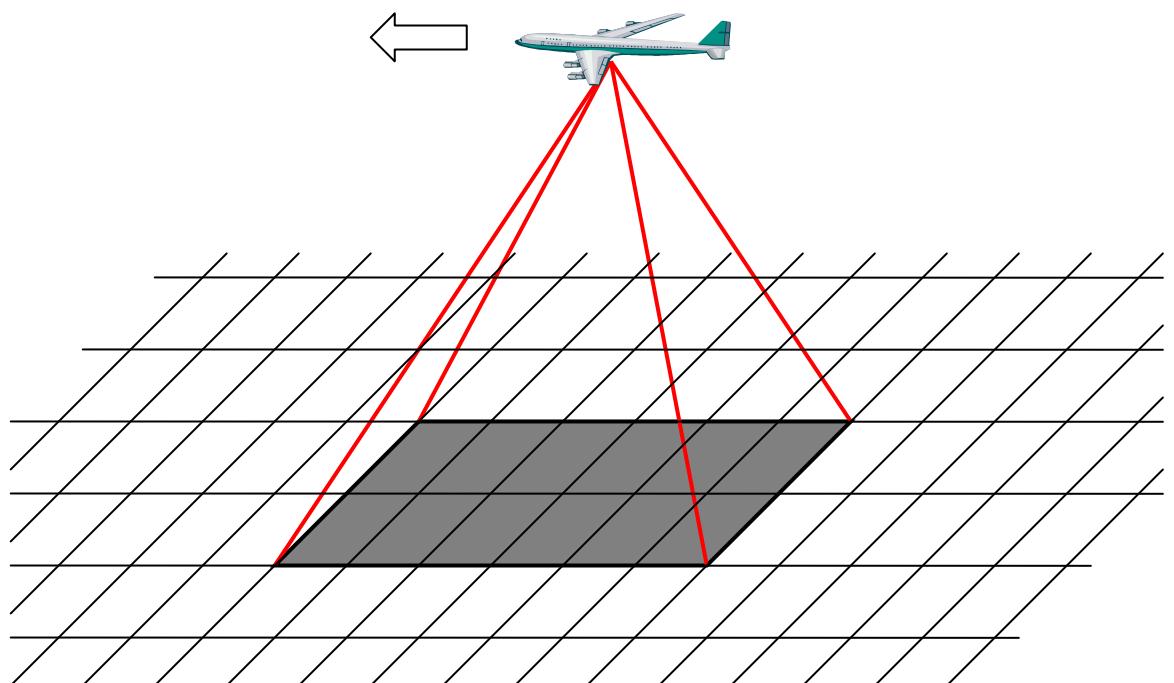
1. Разработан, реализован и исследован алгоритм одновременного сопровождения и оконтурирования на основе метода максимального правдоподобия. Исследования на основе модельных и реальных видеопоследовательностей показали хорошие результаты, в том числе на нестационарных фонах в условиях существенной априорной неопределенности.
2. Для некоторых типичных ситуаций разработан и реализован ряд алгоритмов сопровождения и оконтурирования, которые показали хороший результаты в плане точности, надежности и скорости работы.
3. Разработан и реализован алгоритм оценки собственных шумов матрицы фотоприемника.
4. Реализован метод управления приводом камеры для слежения за объектами по результатам алгоритма сопровождения.

## 4 Селекция движущихся целей

### 4.1 Введение

В обработке видеоизображений распространена задача выделения полезных объектов в видеопоследовательности.

Если мы имеем дело с движущимся точечным или квазиточечным объектом, то такая задача называется задачей селекции движущихся целей. На сегодня в этом направлении проделана большая работа. В основном исследования коснулись радиолокационную технику [21]. Есть также некоторые работы по селекции движущихся целей при наблюдении оптическими средствами [12]. Хорошо исследована и решена задача для случая нормального белого шума.



**Рис. 4.1 Схема наблюдения**

В данной главе рассматривается задача селекции движущихся целей в случае нестационарного фона. Нестационарность проявляется в плоскопараллельном движении подстилающей поверхности. Она характерена

для наблюдения с авиационных или космических носителей в направлении надира (Рис. 4.1).

Также существенным моментом является возможность случайных колебаний оптической оси сенсора, которые необходимо учитывать при построении алгоритма.

Здесь используется следующая структура алгоритма селекции движущихся целей:

1. Построение оценки фона для текущего кадра на основе нескольких предыдущих.
2. Обеление кадра с использованием оценки фона.
3. Выделение на обеленном кадре отметок.
4. Обнаружение связанных траекторий.

В данной работе использовались готовые алгоритмы [21] выделения отметок и объединение их в траектории. Однако для оценки эффективности (т.е. подсчета вероятности ложных тревог и пропуска цели) использовались именно результаты последнего модуля – обнаружения траекторий.

Для построения оценки фона выбран метод авторегрессии [12], модифицированный для работы с нестационарным фоном заданного характера и описанный в разделе 4.2.

В качестве алгоритмов обеления использованы как линейные, так и нелинейные методы, описанные в разделе 4.3.

В разделе 4.4 приведены результаты тестирования алгоритма, а в разделе 4.5 указаны границы применимости предложенных методов.

Раздел 4.6 посвящен выводам.

## **4.2 Алгоритм авторегрессии**

Алгоритм фильтрации для данного случая был создан на основе алгоритма авторегрессии. Согласно ему обеленный кадр вычисляется по формуле:

$$\|W_{ij}\| = \left\| F_{ij}(t) - \sum_k a_k f^k{}_{ij} \right\|, \quad (4.1)$$

где  $f^k{}_{ij}$  - выбираемые специальным образом предыдущие кадры,  $a_k$  - коэффициенты, вычисляемые из критерия

$$[a] = \arg \min_{[a]} \left\{ \sum_{i,j} (W_{ij})^2 \right\}, \quad (4.2)$$

Тогда коэффициенты  $a_k$  находятся из системы

$$[R] \cdot [a] = [\rho], \quad (4.3)$$

где  $[R]$  – корреляционная матрица  $k$ -ых кадров,  $[a]$  – коэффициенты  $a_k$ , а  $[\rho]$  - коэффициенты корреляции  $k$ -го кадра и текущего.

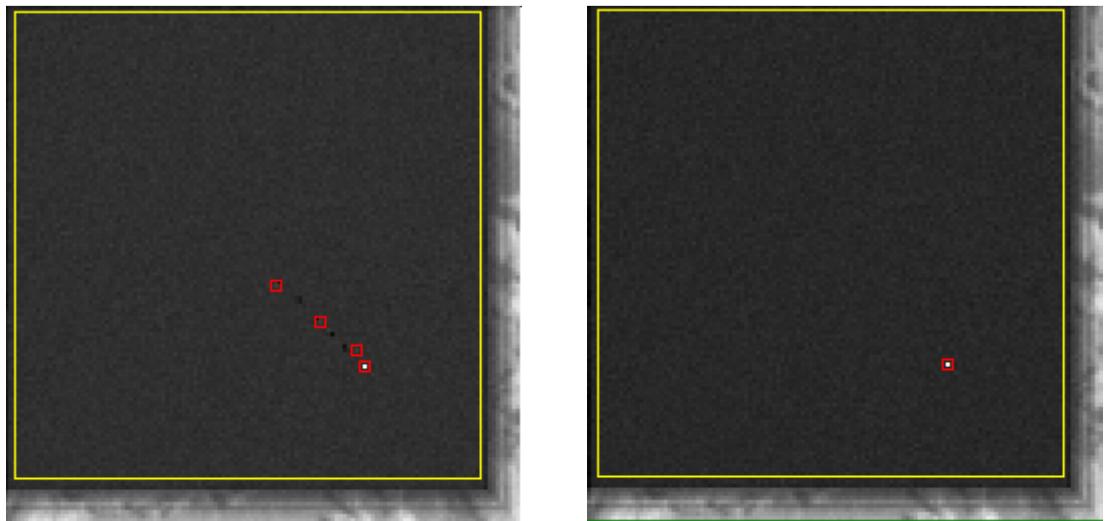
Для правильного выбора кадров в линейной комбинации определяется сдвиг  $T$  предыдущих кадров относительно текущего (где  $T$  – длина временного окна) методом, описанным в первой главе, и в  $f^k{}_{ij}$  подставляются «сдвинутые» кадры. Т.к. сдвиг – нецелая величина, то для достижения хорошего результата в линейной комбинации вместо одного кадра подставляются четыре, сдвинутые относительно текущего на целое число пикселей, ближайшее к вычисленному.

### 4.3 Сигнальная компенсация

Недостатком предложенного метода авторегрессии является возможность возникновения большого количества ложных отметок, связанных с одной яркой действительной отметкой. Из формул (4.1), (4.2) и (4.3) следует, что некоторые из весовых коэффициентов  $a_k$  могут быть отрицательными, а, значит, при обелении соответствующий кадр будет прибавлен к текущему, что приведет к появлению ложных отметок (см. Рис. 4.2).

Для устранения этого недостатка был предложен нелинейный метод сигнальной компенсации. Согласно ему при обелении в кадрах, относящихся к предыдущим моментам времени «вырезаются» яркие отметки, которые были селектированы как цели. Яркость в точках, из которых данные отметки

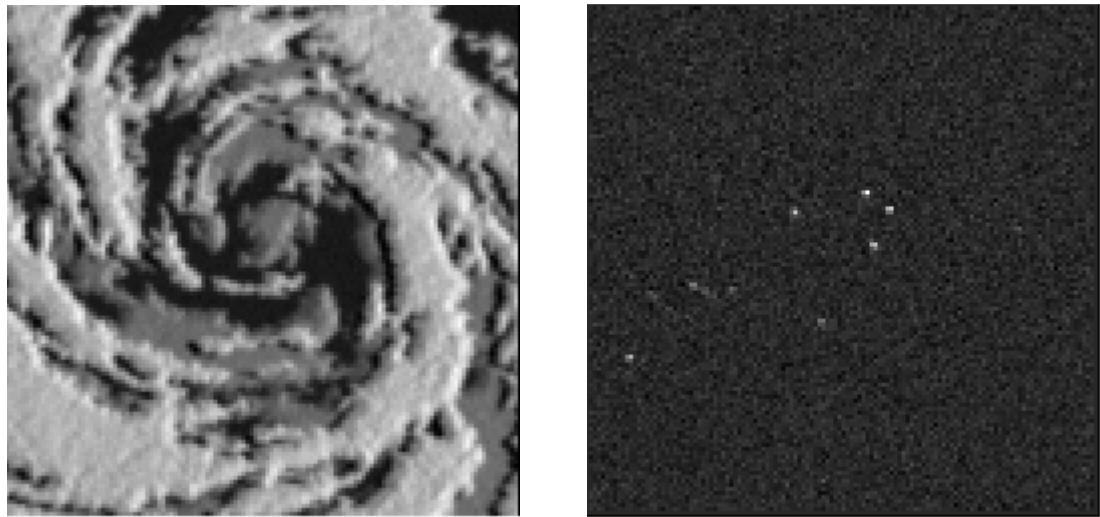
вырезаются, заменяется либо на среднюю яркость в окрестности этих точек (пространственная сигнальная компенсация), либо на яркость этих точек в предыдущие моменты времени, когда цель находилась за их пределами (временная сигнальная компенсация).



**Рис. 4.2. Слева – результат обеления без сигнальной компенсации, справа – с временной сигнальной компенсацией. Желтый квадрат - область без краевых эффектов, красные квадраты - обнаруженные отметки.**

Метод сигнальной компенсации имеет свои тонкости. В частности, очевидная пространственная сигнальная компенсация может привести к серьезному сбою в работе алгоритма. На Рис. 4.3 представлен пример такого сбоя. За счет ошибок интерполяции алгоритм пространственной сигнальной компенсации привёл к появлению на каждом кадре ложных отметок, которые приводят к завязке ложных траекторий.

Алгоритм пространственной сигнальной компенсации обладает серьезными недостатками и не может быть использован в реальной системе. Все приведенные ниже результаты и выводы относятся только к алгоритму временной пространственной компенсации.



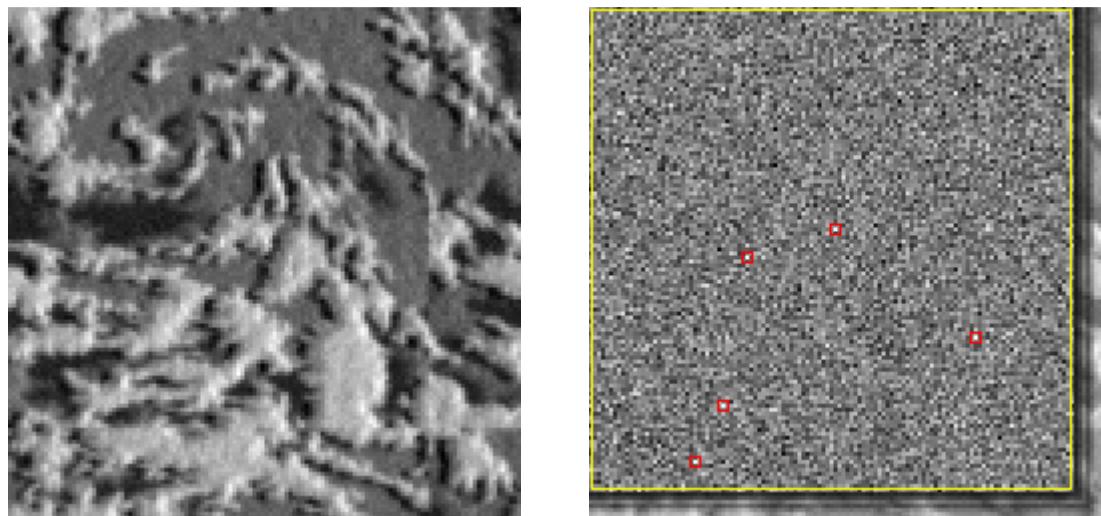
**Рис. 4.3. Пример сбоя алгоритма пространственной сигнальной компенсации. Слева – исходный кадр, справа – обелённый. Видны яркие точки, классифицируемые как отметки, которые выделяются на всех кадрах последовательности.**

#### 4.4 Анализ и результаты

Предложенный алгоритм селекции движущихся целей протестирован на модельных последовательностях.

В качестве генератора тестовых последовательностей использована модель [22], созданная сотрудниками ОАО «МАК «Вымпел», при непосредственном участии автора данной работы. Эта модель позволяет имитировать изображение, получаемое сенсором, установленным на комическом или авиационном носителе, при наблюдении подстилающей поверхности. В модели учтены такие явления, как отражение солнечного света от поверхности Земли и облаков (для генерации рельефа и структуры облачного слоя, а также функции рассеяния использованы соответствующие имитаторы), пропускание атмосферы и эффекты в оптике камеры. Для моделирования целей был использован как имитатор на основе физической модели светимости [55], так и простой имитатор, добавляющий к изображению движущиеся точки с заданными характеристиками.

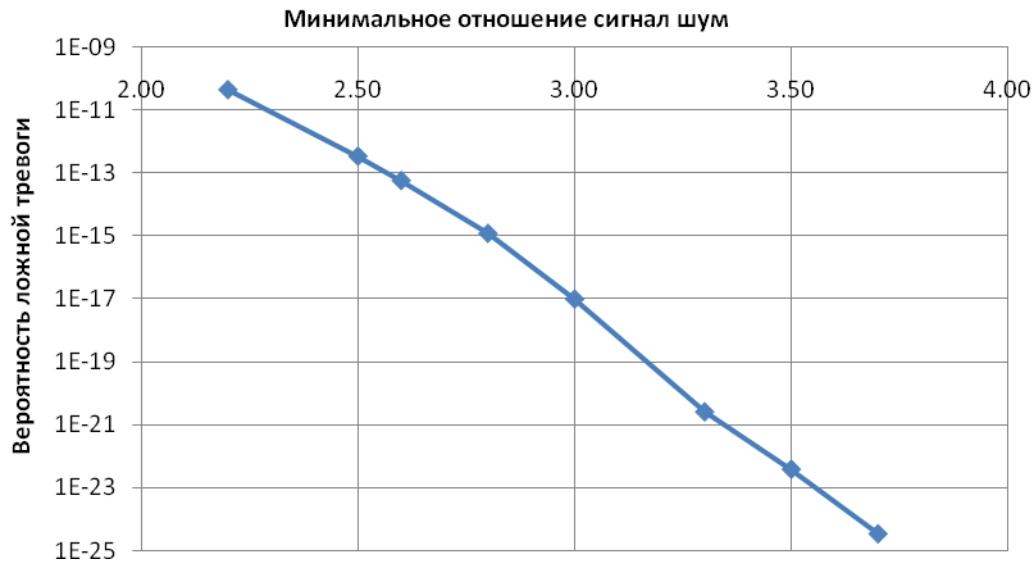
Типичный результат фильтрации показан на Рис. 4.4. Сводные результаты тестирования приведены в Табл. 4.1, а кривая эффективности на Рис. 4.5.



**Рис. 4.4. Кадр тестовой последовательности и результат его обеления. Желтый квадрат - область без краевых эффектов, красные квадраты - обнаруженные отметки.**

**Табл. 4.1 Характеристики алгоритма СДЦ.**

Уровень остаточных шумов обеленного кадра в (шумах матрицы фотоприемника)	1.1
Минимальный уровень обнаружимого сигнала (в шумах матрицы фотоприемника)	3.0
Точность определения сдвига (СКО в пикселях)	0.08



**Рис. 4.5. Кривая эффективности. Вероятность ложной тревоги вычислена на один кадр, а минимальное отношение сигнал шум из расчета вероятности пропуска  $10^{-6}$ .**

В целом алгоритм показал результаты, близкие к оптимальным.

#### 4.5 Границы применимости

Исследуем границы применимости предложенных алгоритмов селекции движущихся целей. Основной величиной, характеризующей качество работы алгоритмов, является предельный уровень сигнала, необходимый для детектирования цели.

Найдём зависимость минимального уровня сигнала от ошибки определения сдвига.

Найдем шум интерполяции исходя из значений производных яркости кадра:

$$n_D = \sigma \left( \frac{\overline{\partial F}}{\partial x} + \frac{\overline{\partial F}}{\partial y} \right) / 2, \quad (4.4)$$

где  $\sigma$  - ошибка определения сдвига.

Исходя из Табл. 4.1, видим, что предельное отношение сигнал-шум равно 2.727 для уровня шума 1.1. Таким образом, зная ошибку определения сдвига, можно вычислить шум интерполяции, а по нему и предельный уровень сигнала.

Эта ошибка зависит от следующих параметров: высоты  $H$  и скорости  $v$  носителя, темпа кадров  $n$ , углового размера пикселя изображения  $\varphi_0$  и собственного шума фотоприемной матрицы  $\sigma_n$ .

В качестве примера такого анализа, рассмотрим подробно две типичные ситуации:

1. Наблюдение с авиационного носителя с высоты 11 км при темпе 10 кадров в секунду и скорости носителя 800 км/ч с разрешающей способностью  $10^{-3}$  рад.
2. Наблюдение с низковысотного спутника с высоты 400 км при темпе 1 кадр в секунду и скорости носителя 8 км/с с разрешающей способностью  $10^{-4}$  радиан.

Собственный шум фотоприемной матрицы в обоих случаях принят равным 1, а средняя производная яркости кадра - 10.

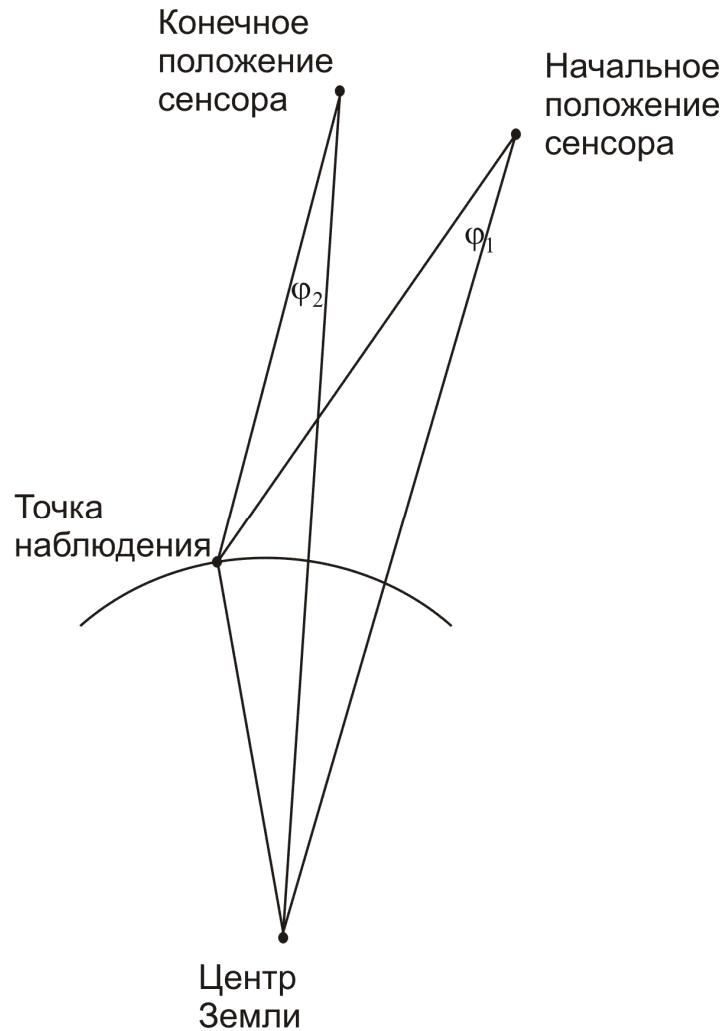
Будем учитывать следующие факторы:

1. Искажения за счет сферичности поверхности Земли.
2. Скорость ветра.

Основным фактором, ограничивающим применение предложенных алгоритмов, является сферичность поверхности Земли. При создании алгоритмов Земля считалась плоской.

Следует заметить, что при достаточно узком поле зрения Землю всегда можно считать плоской. Влияние сферичности Земли увеличивается при увеличении поля зрения.

Найдем зависимость предельного уровня сигнала в зависимости от величины поля зрения.



**Рис. 4.6. К выводу зависимости предельного отношения сигнала-шум от величины поля зрения.**

Для этого вычислим величину изменения угла наблюдения на одну и ту же точку за один такт времени (см. Рис. 4.6):

$$\Delta = |\varphi_2 - \varphi_1|,$$

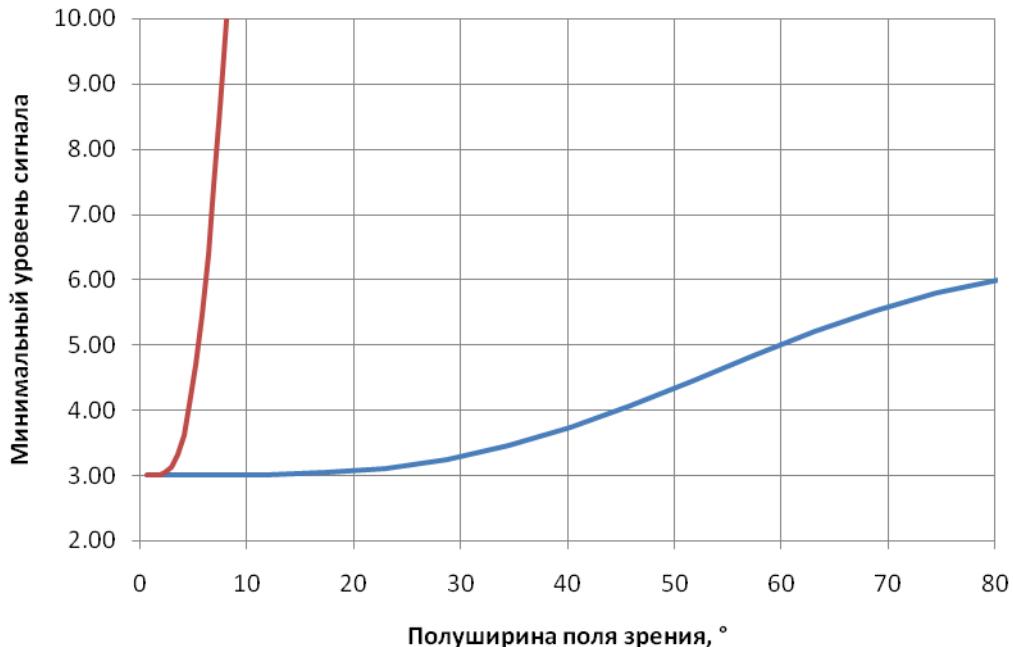
$$\varphi_2 = \arcsin \sqrt{\frac{R \sin \beta}{R^2 + (R + H)^2 - 2R(R + H)\cos \beta}}, \quad (4.5)$$

$$\beta = \pi - \varphi_1 - \arcsin \left( \frac{R + H}{R} \sin \varphi_1 \right) - \arcsin \frac{v}{n(R + H)},$$

где  $R$  - радиус Земли. Тогда разность в изменении угла для центра и края поля зрения будет

$$\Delta_\varphi = \Delta|_{\varphi_1=\theta/2} - \Delta|_{\varphi_1=0}, \quad (4.6)$$

где  $\theta$  - величина поля зрения. График зависимости минимального уровня сигнала для указанных выше параметров приведен на Рис. 4.7.

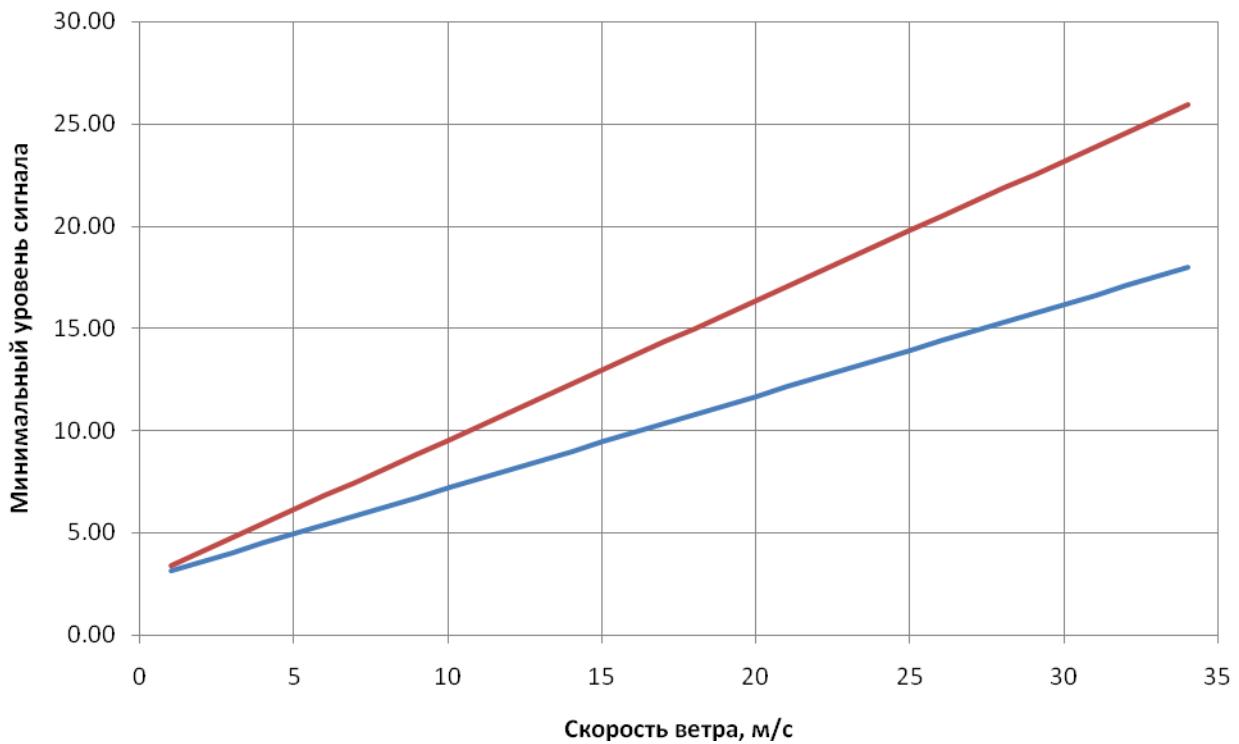


**Рис. 4.7. Зависимость минимального уровня сигнала от угла поля зрения. Красный график – наблюдение с космического носителя, синий – с авиационного.**

Скорость ветра, определяющая скорость смещения облаков, влияет на исследуемую величину. Его влияние особенно сильно в ситуациях, когда облака занимают значительную часть кадра, но не весь кадр. Это приводит к тому, что оцененный сдвиг корректен только для части кадра (поверхности Земли либо облаков). Смещение другой части кадра определяется силой ветра и может быть вычислено по формуле

$$\sigma = \arcsin \frac{v_c}{n(H-h)} / \varphi_0, \quad (4.7)$$

где  $h$  - высота облачного слоя, а  $v_c$  - скорость облаков. График зависимости минимального уровня сигнала для указанных выше параметров приведен на Рис. 4.8.



**Рис. 4.8. Зависимость минимального уровня сигнала от скорости ветра. Красный график – наблюдение с космического носителя, синий – с авиационного.**

Многочисленные эксперименты на модельных кадрах подтверждают полученные здесь зависимости.

#### 4.6 Выводы

1. Разработан, реализован и исследован алгоритм селекции движущихся целей при авиационном и космическом наблюдении. Алгоритм основан на методе авторегрессии с использованием описанного в первой главе алгоритма определения геометрических преобразований кадров. Предложенный алгоритм на модельных последовательностях показал характеристики, близкие к предельным.
2. Разработан метод анализа характеристик в зависимости от конкретных условий решаемой задачи. Показано, что

предложенный алгоритм решает задачу для типичных условий наблюдения.

## 5 Экспериментальная установка и программная реализация

### 5.1 Введение

Для исследования, отладки и тестирования предложенных алгоритмов автором, совместно с коллективом Кафедры информационных систем МФТИ (ГУ) при ОАО «МАК «Вымпел», разработан и создан экспериментальный прототип комплекса реального времени обработки видеинформации.

### 5.2 Описание экспериментальной установки

На Рис. 5.2 представлена схема экспериментальной установки, состоящей из блока видеонаблюдения и рабочего места оператора.

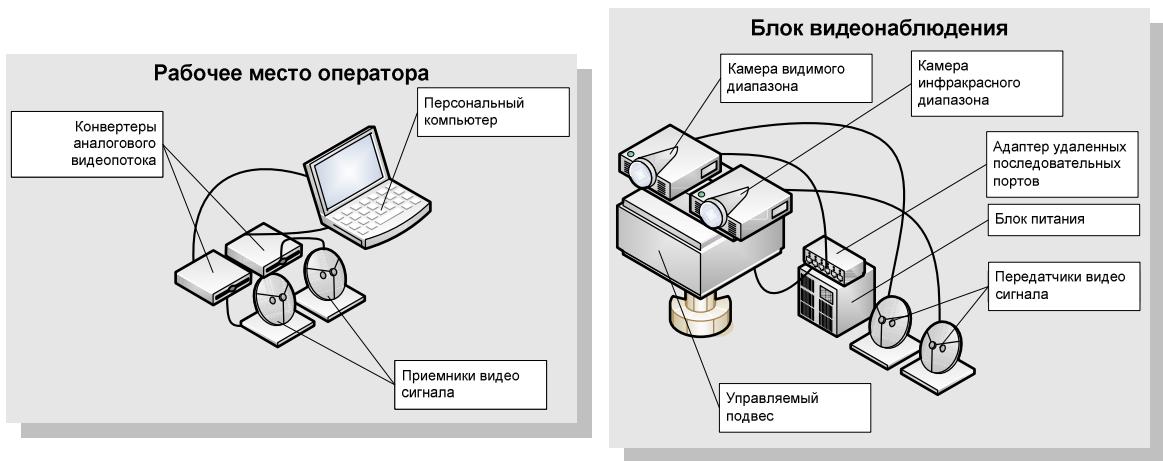
Блок видеонаблюдения осуществляет съемку видео и его передачу по беспроводному радиоканалу, передает текущие параметры камер и подвеса, а также выполняет команды по перенацеливанию камер и изменению их параметров (фокусного расстояния, выдержки и др.).



Рис. 5.1. Внешний вид установки.

Рабочее место оператора обрабатывает команды оператора, принимает видеопоток и поток данных от блока видеонаблюдения, осуществляет интеллектуальную обработку видео, отображает результаты и посыпает управляющие команды на блок видеонаблюдения.

Использование беспроводной связи позволяет разнести рабочее место оператора и блок видеонаблюдения на расстояние до 50 метров без использования непосредственного физического соединения.



**Рис. 5.2. Структура экспериментального прототипа.**

Экспериментальная установка включает следующие элементы:

1. Управляемый подвес QuickSet QTP-90. Осуществляет нацеливание оптических осей камер по командам от персонального компьютера, передаваемым посредством последовательного порта RS-232.
2. Камера видимого диапазона SONY FCB-IX10A. Осуществляет видеосъемку в видимом диапазоне. Фокусное расстояние, выдержка, относительное отверстие объектива и другие параметры могут быть изменены по командам, передаваемым посредством последовательного порта RS-232.
3. Камера инфракрасного диапазона ThermalEye 3600AS. Осуществляет видеосъемку в инфракрасном свете в диапазоне 7 –

14 мкм, соответствующем температурам от -60 до 140 градусов Цельсия.

4. Адаптер удаленных портов последовательного доступа Quatech DSE-100, позволяющий транслировать команды двух последовательных портов RS-232 по беспроводной связи по технологии WiFi (IEEE 802.11).
5. Два устройства передачи видеопотока по радиоканалу RF-Link Audio/Video Sender, позволяющие передавать видеинформацию на расстояние до 20 метров, используя радиоканал на частоте около 5.8 ГГц.
6. Блок питания на 5 и 12В.
7. Два устройства приема видеопотока по радиоканалу RF-Link Audio/Video Reciever, позволяющие принимать видеинформацию на расстояние до 50 метров, используя радиоканал на частоте около 5.8 ГГц.
8. Два конвертера аналогового видеопотока в цифровой Plextor ConvertX AV100U. Производит преобразование аналогового видеосигнала в цифровую форму и передает оцифрованный сигнал персональному компьютеру через USB 2.0 порт.
9. Персональный компьютер. Осуществляет отображение видеопотоков и другой информации, обработку команд оператора, выполнение алгоритмов обработки видеопоследовательности и контроль над установкой.

Структура установки позволяет осуществлять видеосъемку с последующей передачей информации по радиоканалу, контроль за направлением оптической оси и параметрами камер, такими как фокусное расстояние объектива, выдержка и относительное отверстие объектива и т.д.

Установка имеет особенности, которые необходимо учитывать при проектировке и создании программного обеспечения и алгоритмов обработки видеоизображений:

1. Задержка между моментом съемки кадра и его обработкой на персональном компьютере, вызванная задержками в канале связи и при оцифровке аналогового видеопотока. Данная особенность нашла отражение в алгоритме управления приводом камеры (раздел 3.6).
2. Большое количество выходной информации и решаемых задач, а также большое количество свободных параметров, которые необходимо контролировать (например, направление оптической оси и фокусное расстояние), не позволяют оператору задавать подробные начальные данные для алгоритмов сопровождения. Оператор имеет возможность только указать точку, принадлежащую интересующему его объекту, но не его размер или форму. Это было учтено в алгоритме одновременного сопровождения и оконтуривания методом максимального правдоподобия (раздел 3.4).
3. Возможность управления камерой оператором приводит к необходимости разделения трансформаций кадров на управляемые и шумовые. Решение такой задачи чисто техническими средствами, т.е. выдача реального положения камеры, в рамках предложенной аппаратной организации невозможно. Это связано с недостаточной точностью определения положения оптической оси и недостаточной точностью синхронизации движения подвеса и съемки кадров. Для преодоления этих трудностей разработан алгоритм разделения полной трансформации на шумовую и управляемую (раздел 2.2).

### 5.3 Программная реализация

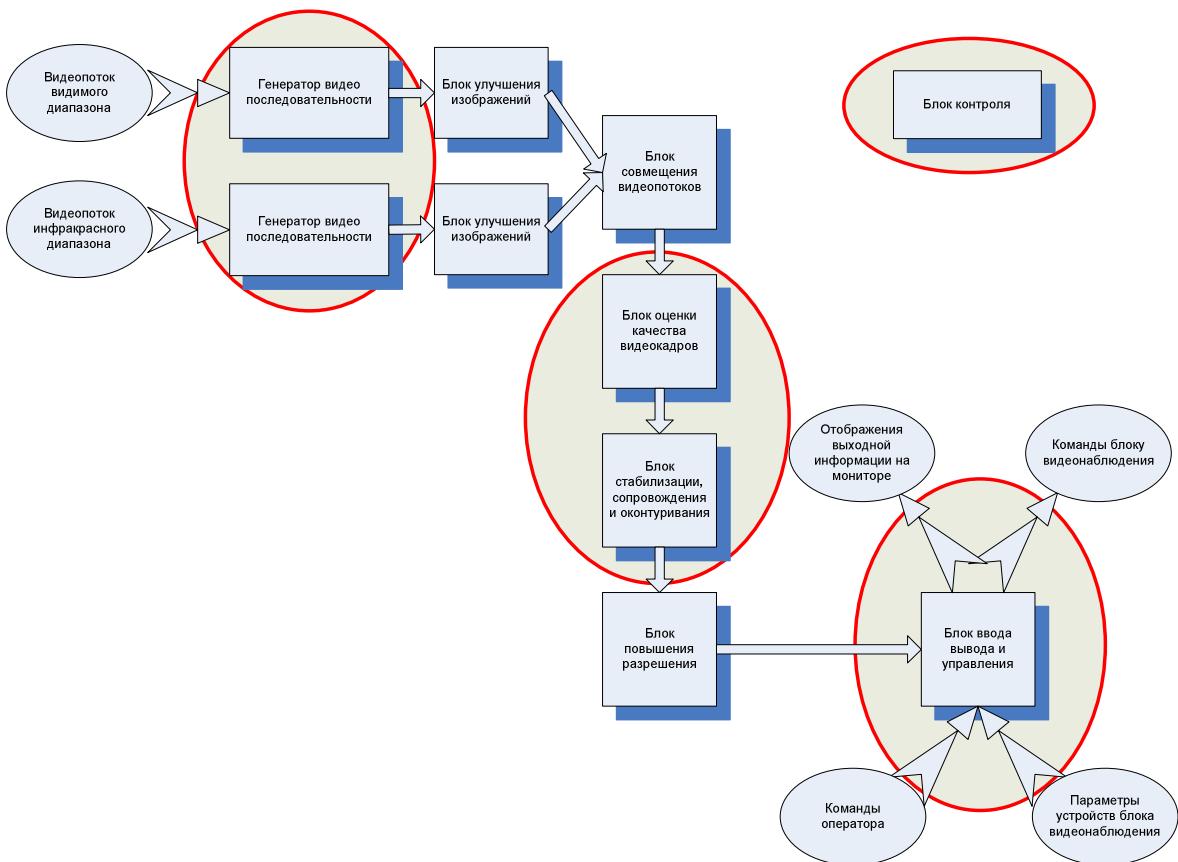


Рис. 5.3. Блок-схема программного обеспечения экспериментальной установки. Прямоугольниками обозначены отдельные процессы. Красными овалами обозначены блоки, выполненные полностью автором представленной работы на основе описанных здесь алгоритмов.

Программное обеспечение, созданное автором в сотрудничестве с коллективом Кафедры информационных систем МФТИ (ГУ) при ОАО «МАК «Вымпел» позволяет осуществлять контроль над блоком видеонаблюдения и выводить на экран выходную информацию. Автором данной работы непосредственно разработана и выполнена объектно-ориентированная многопотоковая архитектура ПО, а также отдельные его блоки, а именно:

1. Блок контроля.
2. Генератор видеопоследовательности.
3. Блок оценки качества видеокадров.
4. Блок стабилизации, сопровождения и оконтуривания.

## 5. Блок ввода-вывода и управления.

Программное обеспечение экспериментальной установки выполнено по многопроцессовой технологии, т.е. каждый модуль программы выполнен в виде отдельного счетного процесса. Каждый процесс имеет независимое адресное пространство и конкурирует за вычислительный ресурс центрального процессора.

Такой подход позволяет максимально использовать существующие многопроцессорные и многоядерные компьютеры, т.к. операционная система автоматически распределяет вычисления разных потоков на разные процессоры, оптимально используя вычислительные ресурсы.

Взаимодействие между процессами осуществляется стандартными средствами операционной системы и заключается в обмене видеокадрами, а также управляющими и информационными командами.

В связи с тем, что каждый процесс должен в реальном времени обрабатывать синхронизированную последовательность кадров и асинхронные команды, использована многопоточная технология. Т.е. в рамках адресного пространства одного процесса выполняются несколько параллельных потоков. Среди них: поток обработки видеинформации с заданным темпом, поток обработки асинхронных команд, отдельные потоки для записи и для чтения в последовательные порты и др.

Разработанное программное обеспечение включает следующие блоки (процессы):

1. Блок контроля следит за корректностью выполнения программы, а также производит запуск и закрытие остальных процессов.
2. Генератор видеопоследовательности. Обеспечивает взаимодействие с устройством видеовхода и осуществляет запись видеопотока в память для последующей обработки.

3. Блок улучшения изображения решает задачи по улучшению качества изображения, такие как фильтрация, эквализация гистограмм [49] и др.
4. Блок совмещения видеопотоков инфракрасного и видимого диапазонов, выполненный на основе методов из [54].
5. Блок оценки качества видеокадров, использующий методы раздела 2.4.
6. Блок стабилизации, сопровождения и оконтуривания, решающий эти задачи методами, изложенными в первых трех главах.
7. Блок повышения разрешения. Производит улучшение качества и повышение разрешения выделенных частей кадра. Выполнен на основе алгоритмов [17].
8. Блок ввода-вывода и управления. Осуществляет обработку команд оператора, введенных с клавиатуры, джойстика и других устройств, отображение выходной информации, а также чтение и контроль параметров устройств блока видеонаблюдения.

Программное обеспечение написано, в основном, на языке C++ в среде со следующими параметрами:

1. PC совместимый компьютер.
2. Операционная система: Microsoft® Windows™ XP.
3. Среда разработки: Microsoft® Development Environment 2005 v8.0.
4. Компилятор: Intel® C++ Compiler v9.0.
5. Дополнительные библиотеки функций:
  - a. Intel® Integrated Performance Primitives 3.0.
  - b. Intel® Open source computer vision library.
6. Средство оптимизации: Intel® Vtune™ Performance Analyzer v7.0.

Для увеличения скорости работы, помимо алгоритмических, были предприняты определенные программные ходы. Это, прежде всего, использование SSE2 инструкций процессора. Данные инструкции позволяют за

один такт выполнять до 16 однородных операций. Код, использующий такие инструкции, написан на языке Ассемблера.

Следует указать, что в описанных выше алгоритмах используется некоторое количество настроек параметров: размеры окон наибольшей информативности, степень сжатия, пороги и т.д. Оптимальные их значения могут зависеть от различных входных параметров, в том числе: размер изображения, наличие или отсутствие чересстрочной развертки, включения режима определения сдвига по сжатым кадрам и т.д.

Для настройки внутренних параметров алгоритмов, исходя из внешних параметров программы, используется блок автонастройки. Все внутренние параметры и их зависимость от внешних были подобраны экспериментально на основе многочисленных экспериментов с широким спектром входных видеопоследовательностей.

Создана система автоматического тестирования алгоритмов и выявления их оптимальных параметров. Она в автоматическом режиме перебирает все комбинации внутренних параметров алгоритмов и на основе тестовых последовательностей оценивает их характеристики, такие как скорость, точность оценок и др.

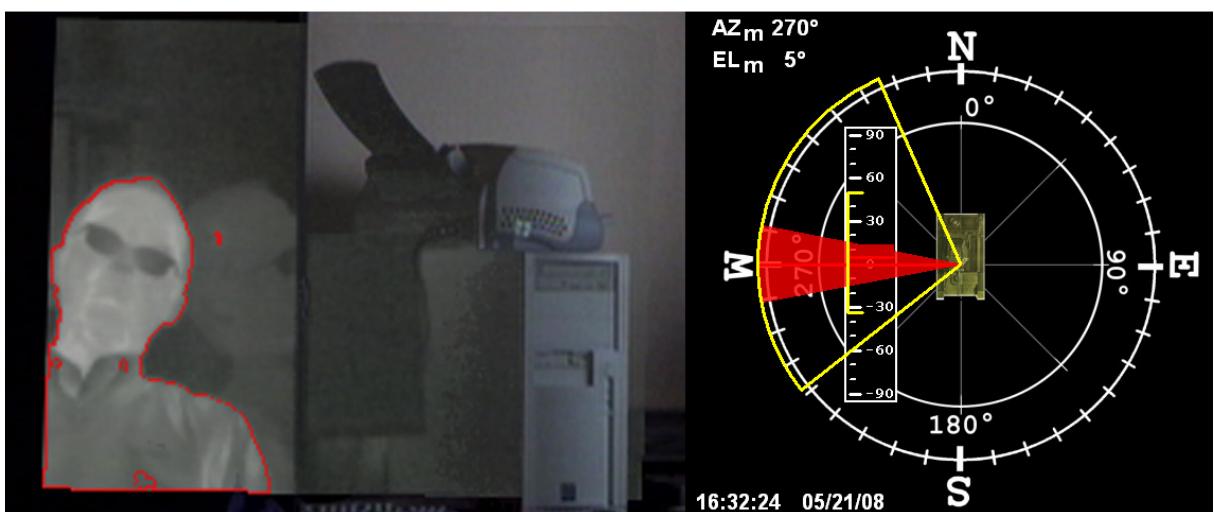


Рис. 5.4. Внешний вид окна приложения.

Другой особенностью реализации алгоритмов является возможность ограничения времени выполнения для обеспечения обработки в реальном времени.

В части 1.2.7 приведены данные о точности при использовании таких ограничений.

#### **5.4 Функциональные возможности**

Предложенный экспериментальный прототип способен решать следующие задачи:

1. Осуществлять видеонаблюдение в двух диапазонах длин волн с помощью автономного блока видеонаблюдения.
2. Осуществлять улучшение качества видеоизображения, в частности, за счет стабилизации и увеличения разрешения.
3. Объединять видеопотоки видимого и инфракрасного диапазона с учетом различий в геометрических параметрах камер. Объединенный видеопоток содержит все наиболее информативные детали обоих диапазонов и позволяет оператору получить максимум информации с одного кадра.
4. Осуществлять слежение за объектом и перенацеливать камеры для обеспечения длительного слежения при выходе объекта за границы первоначального поля зрения.
5. Осуществлять поиск контура объекта для облегчения восприятия информации.

Данная экспериментальная установка может рассматриваться как прототип милицейской системы видеонаблюдения, позволяющей осуществляют наблюдение и слежение за объектами. Такая система может быть установлена как стационарно для контроля периметра, так и на подвижных носителях, таких как вертолет или автомобиль.

**Табл. 5.1 Характеристики экспериментального прототипа.**

Диапазон длин волн видеокамер	0.38-0.76 мкм и 7-14 мкм
Поля зрения камер	63° на 43° для видимого диапазона и 50° на 37° для инфракрасного диапазона
Предельное оптическое увеличение	10
Дальность беспроводного соединения между рабочим местом и блоком видеонаблюдения	50м
Размер кадров видеопотока	640 на 480 пикселей
Темп съемки кадров	30 кадров в сек
Задержка между съемкой и отображением кадров	не более 0.2 сек
Предельная скорость перенацеливания (слежения) камер	3° в сек
Вес блока видеонаблюдения	9 кг

Среди возможных гражданских применений - охранные системы, системы помощи водителю в автомобилях и другие системы наблюдения и мониторинга.

### **5.5 Выводы**

1. Спроектирована и создана экспериментальная установка комплекса реального времени обработки изображений с возможностью удаленного контроля и передачи видеинформации.
2. Разработанное программное обеспечение позволяет в реальном времени на современных неспециализированных вычислительных средствах под управлением многозадачной операционной системы

выводить на экран все результаты обработки входных видеоизображений, контролировать параметры системы оператором, а также проводить автоматическое управление устройствами, в частности, осуществлять слежение за объектом.

3. Данная система может быть рассмотрена как прототип системы видеонаблюдения для милицейских и военных целей, а также для гражданских охранных и других систем видеонаблюдения.

## Выводы

1. Разработан алгоритм определения геометрических трансформаций кадров видеопоследовательности. Под геометрическими трансформациями подразумеваются различные модели преобразований, включая аффинные и проективные. В основе предложенного алгоритма лежит объединение трех основных существующих на сегодняшний день подходов с постоянным контролем качества принятия решений и точности оценок на каждом шаге алгоритма. Это позволило создать алгоритмы, обеспечивающие высокую рабочесть на широком классе видеопоследовательностей, работающие в реальном времени на современных неспециализированных вычислительных средствах.
2. Алгоритм определения геометрических преобразований успешно использован для решения задач стабилизации изображений и построения мозаики.
3. Как необходимое дополнение к задаче определения геометрических трансформаций, получены решения следующих задач: разделение движения камеры на управляемое и шумовое и определение качества видеопоследовательности.
4. На основе алгоритма определения геометрических трансформаций создан комплекс методов и алгоритмов сопровождения объектов. Выведено решающее правило на основе критерия максимального правдоподобия для оценки смещения объекта одновременно с оценкой его контура и размера. Предложенный метод обеспечивает высокую точность и надежность в условиях высокой априорной неопределенности.
5. Создан целый спектр алгоритмов сопровождения и оконтуривания для различных типичных ситуаций. Они показали высокое быстродействие и надежность.

6. Алгоритм сопровождения объектов на видеопоследовательности использован для управления направлением оптической оси фотоприемника для слежения за объектом.
7. Разработанная модификация алгоритма определения геометрических преобразований использована для решения задачи селекции движущихся целей для задач авиационного или космического наблюдения. Исследованы границы его применимости. Разработанный алгоритм показал характеристики, близкие к предельным.
8. Спроектирована и создана экспериментальная установка комплекса реального времени обработки изображений с возможностью удаленного контроля и передачи видеинформации.  
Разработанное программное обеспечение позволяет в реальном времени на современных неспециализированных вычислительных средствах под управлением многозадачной операционной системы выводить на экран все результаты обработки входных видеоизображений, контролировать параметры системы оператором, а также проводить автоматическое управление устройствами, в частности, осуществлять слежение за объектом.  
Данная система может быть рассмотрена как прототип системы видеонаблюдения для милицейских и военных целей, а также для гражданских охранных и других систем видеонаблюдения.

## Литература

1. *S. Negahdaripour, S. Lee.* Motion Recovery from Image Sequences Using First-Order Optical Flow Information. // IEEE Workshop Visual Motion. – Oct. 1991. - P. 132–139.
2. *Berthold K.P. Horn, Brian G. Schunck.* Determining Optical Flow. // Artificial Intelligence, 17, pp. 185-203, 1981.
3. *B. Lucas, T. Kanade* An Iterative Image Registration Technique with an Application to Stereo Vision. // Proc. of 7th International Joint Conference on Artificial Intelligence (IJCAI), pp. 674-679.
4. *Stephen L. Keeling, Wolfgang Ring.* Medical Image Registration and Interpolation by Optical Flow with Maximal Rigidity. // Journal of Mathematical Imaging and Vision, Volume 23, Issue 1 (July 2005).
5. *P. Stumpf.* Über die Abhangigkeit der visuellen Bewegungsrichtung und negativen Nachbildes von den Reizvorgangen auf der Netzhaut. // Zeitschrift fur Psychologie, 1911, V59, P. 321-330.
6. *R. Chipolla, Y. Okamoto, Y. Kuno.* Robust Structure from Motion Using Motion Parallax. // Int'l Conf. Computer Vision. – May 1993. - P. 374–382.
7. *F. Lustman, O.D. Faugeras, G. Toscani.* Motion and Structure from Motion from Point and Line Matching. // Proc. First Int'l Conf. Computer Vision, P. 25–34, London, 1987.
8. *A. Censi, A. Fusiello, V. Roberto.* Image stabilization by features tracking // Image Analysis and Processing, 1999. Proceedings. International Conference on Volume , Issue , 1999 Page(s):665 - 667
9. *B. K. P. Horn, E. J. Weldon, Jr.* Direct methods for recovering motion. // International Journal of Computer Vision, 2:51–76, 1988.
10. *David G. Lowe* Distinctive Image Features from Scale-Invariant Keypoints. // International Journal of Computer Vision, Volume 60, Issue 2 (November 2004).

- 11.И. С. Грузман и др. Цифровая обработка изображений в информационных системах. Новосибирск, НГТУ, 2002.
- 12.А.К. Ким, А.Е. Колесса, В.Н. Лагуткин, А.В. Лотоцкий, В.Г. Репин. Алгоритмы идентификации и подавления нестационарного мешающего фона и повышение разрешающей способности в оптическом сенсоре с хаотически колеблющейся оптической осью и динамическими аберрациями. // Радиотехника, 1998, №12
- 13.Дж. Дэннис, Р. Шнабель. Численные методы безусловной оптимизации и решения нелинейных уравнений. / Пер. с англ. М: Мир, 1988
- 14.Й. Абаффи, Э. Спендикато. Математические методы для линейных и нелинейных уравнений. Проекционные ABS-алгоритмы. М: Мир, 1996.
- 15.Alex Rav-Acha, Shmuel Peleg. Restoration of Multiple Images with Motion Blur in Different Directions. // IEEE Computer Society Conference on Computer Vision and Pattern Recognition, proceedings of the conference, 2000.
- 16.Richard Szeliski, Heung-Yeung Shum. Creating Full View Panoramic Image Mosaics and Environment Maps. // Microsoft Research, 1997.
- 17.В.Г. Репин, В.Н. Лагуткин. Спектральный метод компенсации смещений и повышения разрешения изображений при обработке последовательности смещающихся цифровых кадров. // Радиотехника и электроника, том 45, №11, с 1359-1364.
- 18.Stephen C. Cain, Majeed M. Hayat and Ernest E. Armstrong. Projection-Based Image Registration in the Presence of Fixed-Pattern Noise. // IEEE transaction on image processing, V. 10, N. 12, December 2001.
- 19.В.Н. Лагуткин, Ю.В. Радченко. Применение вейвлет-преобразования в задаче оценки смещения объекта. // Вопросы радиоэлектроники, сер. РЛТ, 2004, вып 1.

20. *Репин В.Г., Тартаковский Г.П.* Статистический синтез. М: Советское радио, 1977.
21. *Г. Ван Трис.* Теория обнаружения, оценок и модуляции. М: Советское радио, 1977.
22. *В.Н. Лагуткин, А.П. Лукьянов, Е.Н. Подкорытов, В.Г. Репин.* Трехмерная динамическая модель полей яркости фона, использующая расчетные индикатрисы многократного рассеяния излучения в облаках, полученные метеорологическими спутниками. // Вопросы радиоэлектроники (РТЛ), вып. 1, 2004.
23. *Fatih Porikli.* Achieving Real-Time Object Detection and Tracking Under Extreme Conditions. // JRTIP-2006.
24. *C. Wren, A. Azarbayejani, T. Darell, and A. Pentland.* Pfnder: Real-time tracking of the human body. // IEEE Trans. Pattern Anal. Machine Intell., vol. 19, pp. 780–785, 1997.
25. *Y. Boykov and D. Huttenlocher.* Adaptive bayesian recognition in tracking rigid objects. // Proc. IEEE Conf. on Computer Vision and Pattern Recognition, Hilton Head, SC, vol. II, 2000, pp. 697–704.
26. *M. Isard and I. Blake.* Condensation – conditional density propagation for visual tracking. // Intl. J. of Computer Vision, vol. 29, 1998, pp. 5–28.
27. *N. Bouaynaya, W. Qu, and D. Schonfeld.* An online motion-based particle filter for head tracking applications. // Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing, Philadelphia, 2005.
28. *D. Comaniciu, V. Ramesh, and P. Meer.* Real-time tracking of non-rigid objects using mean shift. // Proc. IEEE Conf. on Computer Vision and Pattern Recognition, Hilton Head, SC, vol. 1, 2000, pp. 142–149.
29. *F. Porikli and O. Tuzel.* Object tracking in low-frame-rate video. // Proc. of PIE/EI - Image and Video Communication and Processing, San Jose, CA, 2005.
30. *S. Avidan.* Ensemble tracking. // Proc. IEEE Conf. on Computer Vision and Pattern Recognition, San Diego, CA, 2005.

31. *J. Korpi-Anttila.* Automatic Colour Enhancement and Scene Change Detection of Digital Video. // Graphic Arts in Finland 32(2003)1.
32. *Ralph M. Ford, Craig Robson, Daniel Temple, and Michael Gerlach.* Metrics for Scene Change Detection in Digital Video Sequences. // ICMCS '97, IEEE.
33. *Shu-Ching Chen, Mei-Ling Shyu, Cheng-Cui Zhang, R. L. Kashyap.* Video scene change detection method using unsupervised segmentation and object tracking. // IEEE International Conference on Multimedia and Expo (ICME), pp. 57-60, August 22-25, 2001, Waseda University, Tokyo, Japan.
34. *Chong-Wah Ngo, Ting-Chuen Pong, Hong-Jiang Zhangz & Roland T. Chin.* Motion-based Video Representation for Scene Change Detection. // International Journal of Computer Vision. Volume 50, Issue 2 (November 2002).
35. *Jong Wan Jang and Il Kyun Oh.* Performance Evaluation of Scene Change Detection Algorithms. // APCC/OECC '99. Vol.2.
36. *Xiaoquan Yi and Nam Ling.* Fast Pixel-Based Video Scene Change Detection. // ISCAS 2005. Vol. 4.
37. *Georgios Akrivas, Nikolaos. D. Doulamis, Anastasios. D. Doulamis and Stefanos. D. Kollias.* Scene Detection Methods for MPEG – Encoded Video Signals. // MELECON 2000. Vol. 2.
38. *Serkan Kiranyaz, Kerem Caglar, Bogdan Cramariuc and Moncef Gabbouj.* Unsupervised Scene Change Detection Techniques In Feature Domain Via Clustering And Elimination. // IWDC 2002, Capri, Italy.
39. *Edmundo Saez, Jose I. Benavides, Nicolas Guil.* Reliable real time scene change detection in MPEG compressed video. // ICME '04. Vol. 1.
40. *H.B. Lu, Y.J. Zhang, Y.R. Yao.* Robust Gradual Scene Change Detection. // ICIP 99. Vol. 3.
41. *Смирнов Н.Б.* Теория вероятностей и математическая статистика. Избранные труды. М.: Наука 1970г.

42. *Alistair J. Fitch, Alexander Kadyrov, William J. Christmas, and Josef Kittler.* Fast Robust Correlation. // IEEE transaction on image processing, V. 14, N. 8, August 2005.
43. *Liwei Wang, Yan Zhang, and Jufu Feng.* On the Euclidean Distance of Images. // IEEE transaction on pattern analysis and machine intelligence, V. 27, N. 8, August 2005.
44. *Heung-Yeung Shum and Richard Szeliski.* Panoramic Image Mosaics. // Microsoft Research. Technical Report. MSR-TR-97-23.
45. *Hieu T. Nguyen and Arnold W.M. Smeulders.* Fast Occluded Object Tracking by a Robust Appearance Filter. // IEEE Transaction on pattern analysis and machine intelligence, V. 26, N. 8, August 2004.
46. *Д. Форсайт, Ж. Понс.* Компьютерное зрение. Современный подход. Вильямс, 2004.
47. *J. Canny.* A Computational Approach to Edge Detection. // IEEE Trans. on Pattern Analysis and Machine Intelligence, 8(6), pp. 679-698 (1986).
48. *Serra J.* Image Analysis and Mathematical Morphology. // Vol. 2. N. Y.: Academic Press, 1988. 480 p.
49. Методы компьютерной обработки изображений. / Под ред. В.А. Сойфера. М.: Физматлит, 2003.
50. *Левин Б.Р.* Теоретические основы статистической радиотехники. Кн. 1. М.: Сов. радио, 1968.
51. *Алейник С.В.* Приближенная плотность распределения суммы квадратов зависимых гауссовых случайных величин. // Радиотехника, 1999, №1. С. 53-55.
52. *W.H. Press et al.* Numerical Recipes in C. Cambridge university press. 1992.
53. *Ф. Уоссермен.* Нейрокомпьютерная техника: Теория и практика. М.: «Мир», 1992.
54. *Пругло А.В., Равдин С.С.* Алгоритм объединения изображений различных спектральных диапазонов с использованием пирамидального разложения. // Вопросы радиоэлектроники. 2006, вып 1.

55. *А.К.Ким, В.Н.Лагуткин, А.П. Лукьянов, Ф.А.Козин, Ю.В.Радченко, Слынько Ю.В.* Пакет программ для согласованного моделирования движения и инфракрасного излучения ракет на активном участке полета. // Вопросы радиоэлектроники, сер. РТЛ, 2007, вып. 4. С. 30-44. М.: ЦНИИ Электроника.